

HUMAN-COMPUTER INTERACTION

JENNY PREECE

- YVONNE ROGERS
- HELEN SHARP
- DAVID BENYON
- SIMON HOLLAND
- TOM CAREY



ADDISON-WESLEY

Human-Computer Interaction

JENNY PREECE

Yvonne Rogers

Helen Sharp

David Benyon

Simon Holland

Tom Carey



Addison-Wesley

Harlow, England • Reading, Massachusetts • Menlo Park, California • New York
Don Mills, Ontario • Amsterdam • Bonn • Sydney • Singapore • Tokyo • Madrid
San Juan • Milan • Mexico City • Seoul • Taipei

© 1994 The Open University

Addison-Wesley Longman Limited
Edinburgh Gate
Harlow
Essex, CM20 2JE
England

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior written permission of the publisher.

The programs presented in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. The publisher does not offer any warranties or representations, nor does it accept any liabilities with respect to the programs.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Addison-Wesley has made every attempt to supply trademark information about manufacturers and their products mentioned in this book. A list of the trademark designations and their owners appears below.

EDITOR-IN-CHIEF: Simon Plumtree
PRODUCTION MANAGER: Stephen Bishop
PRODUCTION EDITOR: Susan Keany
PRODUCTION CONTROLLER: Jim Allman
TEXT DESIGNER: Valerie O'Donnell
ILLUSTRATIONS: Chartwell Illustrators
TYPESETTERS: CRB Associates, Norwich
COVER DESIGNERS: Designers & Partners, Oxford
PRINTED AND BOUND in Great Britain by Biddles Ltd, Guildford and King's Lynn

First printed 1994. Reprinted 1995 (twice) and 1996. Reprinted 1997 (twice)

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library.

Library of Congress Cataloging in Publication Data

Preece, Jenny.

Human-computer interaction / Jenny Preece [with] Yvonne Rogers . . .
[et al.].

p. cm.

Includes bibliographical references and index.

ISBN 0-201-62769-8

1. Human-computer interaction. I. Title.

QA76.9.H85P74 1994

004'.01'9—dc20

94-16158

CIP

Trademark notice

MacPaint, QuickTime, MacDraw, HyperCard and Macintosh are trademarks of Apple Computer Inc.
Excel, Word, Microsoft and Windows for Workgroups are trademarks of Microsoft Corporation
Post-it is a trademark of 3M

UNIX is a trademark of UNIX System Laboratories Inc.

VODIS is a trademark of British Telecom, Logica (Cambridge) Ltd, Cambridge University

Private Eye is a trademark of Reflection Technology

SuperBook is a trademark of AT & T

Telstar is a trademark of British Telecom

DataView is a trademark of VI Corp.

PARTS Workbench is a trademark of Digital Inc.

XII Windows System is a trademark of Massachusetts Institute of Technology

Prograph is a trademark of Gunakara Sun Systems Ltd

UIM/X is a trademark of Visual Edge Software Ltd

Etchasketch is a trademark of Peter Pan Playthings Ltd

MacroMind Director is a trademark of Macromedia

Data Glove is a trademark of VPL

Contents

Foreword	v
Preface	vii
Readership	viii
How to use this book	viii
About the Authors	xxi
The Story of this Book and Acknowledgements	xxv
Acknowledgements	xxix

PART I Introduction	1
1 What is HCI?	3
1.1 Technological change: Different design needs	5
1.2 The challenge of HCI	8
1.3 The goals of HCI	14
1.4 HCI and its evolution	16
1.5 The importance of HCI: Productivity	19
1.6 When things go wrong	23
2 Components of HCI	29
2.1 HCI as interdisciplinary practice	30
2.2 Disciplines contributing to HCI	37
2.3 A conceptual model for HCI	43
2.4 Designing HCI	45
Interview with Terry Winograd	53

PART II Humans and Technology: Humans	55
Interview with Donald Norman	59
3 Cognitive Frameworks for HCI	61
3.1 A cognitive perspective	62
3.2 Broadening the cognitive framework	68
4 Perception and Representation	75
4.1 Visual perception	76
4.2 Graphical representation at the interface	82
5 Attention and Memory Constraints	99
5.1 Focusing attention	100
5.2 Memory constraints	108
6 Knowledge and Mental Models	123
6.1 Knowledge representation and organization	124
6.2 Mental models	130
6.3 The utility of mental models in HCI	137
7 Interface Metaphors and Conceptual Models	141
7.1 Verbal metaphors	142
7.2 Virtual interface metaphors	145
7.3 Classification of interface metaphors for applications	149
7.4 Ubiquitous computing	149
7.5 Conceptual models	151
8 Learning in Context	155
8.1 Learning as an active process	156
8.2 Gaining expertise	163
8.3 Psychology of programming	166
8.4 Collaborative and situated learning	170
9 Social Aspects	173
9.1 Analysing conversation	174
9.2 Group communication	178
9.3 Group working in context	181
10 Organizational Aspects	185
10.1 The nature of organizations	186
10.2 The impact of information technology on organizations	188

10.3	Methods for organizational change	190
10.4	CSCW and organizational considerations	197

Interview with Marilyn Mantei	201
--------------------------------------	------------

PART III Humans and Technology: Technology 203

Interview with Ben Shneiderman	207
---------------------------------------	------------

11 Input 211

11.1	Input devices	212
11.2	A sample of input issues: Keyboards	214
11.3	Pointing devices	217
11.4	Matching devices with work	221
11.5	Matching devices with users: Input for the disabled	224
11.6	Matching devices with environments of use	227
11.7	Developments in input	230

12 Output 237

12.1	Devices and output	238
12.2	Visual output	239
12.3	Dynamic visualizations	241
12.4	Sound output	247
12.5	Speech output	252
12.6	Developments in output	255

13 Interaction Styles 261

13.1	Interaction styles	262
13.2	Command entry	263
13.3	Menus and navigation	265
13.4	Form-fills and spreadsheets	268
13.5	Natural language dialogue	269
13.6	Direct manipulation	270
13.7	Cognitive issues in direct manipulation	273

14 Designing Windowing Systems 285

14.1	General issues	286
14.2	Basic window components	290
14.3	Common tasks in windowing systems	300
14.4	Issues in windowing systems for CSCW	305

15	User Support and On-Line Information	307
15.1	Active learning with minimalist manuals	308
15.2	User assistance and on-line help	311
15.3	Hypertext and hypermedia	319
15.4	Designing hypermedia for training in HCI	321
16	Designing for Collaborative Work and Virtual Environments	325
16.1	Computer supported cooperative work	326
16.2	Virtual environments and virtual reality	332
16.3	Design trade-offs: An environment for learning about motion	335
	Interview with Roy Kalawsky	343
PART IV	Interaction Design: Methods and techniques	345
	Interview with Tom Moran	349
17	Principles of User-Centred Design	351
17.1	Fundamentals	353
17.2	The design of software systems	355
17.3	Two examples of user-centred design	360
17.4	The scope of human-computer system design	366
18	Methods for User-Centred Design	371
18.1	Soft systems methodology	373
18.2	Cooperative design	375
18.3	Multiview: A user-centred approach	379
18.4	An HCI design approach	380
19	Requirements Gathering	383
19.1	Functional requirements	385
19.2	Data requirements	395
19.3	Usability requirements	401
19.4	Relationship between requirements and usability	406
20	Task Analysis	409
20.1	Goals, tasks and actions	411
20.2	Hierarchical task analysis	413
20.3	Cognitive task analysis	417
20.4	Modelling 'how to do it' knowledge	419

20.5	Representing task knowledge	424
20.6	Some conclusions	428
21	Structured HCI Design	431
21.1	A framework for design	434
21.2	Conceptual design example: Eurochange	435
21.3	From logical to physical design: Task allocation	441
21.4	Physical design	444
21.5	Completing the design	446
22	Envisioning Design	451
22.1	Holistic design	452
22.2	Sketching and metaphor	456
22.3	Scenarios, storyboards and snapshots	462
PART V	Interaction Design: Support for designers	465
	Interview with Bill Verplank	467
23	Supporting Design	469
23.1	Supporting the design process	470
23.2	Supporting designers	472
23.3	Supporting design teams	479
23.4	Different kinds of support	483
24	Guidelines: Principles and Rules	487
24.1	Principles and rules	488
24.2	Where do guidelines come from?	492
24.3	Evaluating guidelines	495
24.4	An example of applying conflicting guidelines	496
25	Standards and Metrics	501
25.1	Standards and standardization	502
25.2	Software standards	503
25.3	HCI standards	505
25.4	House style guides	511
25.5	Metrics	517
26	Design Rationale	523
26.1	IBIS	525
26.2	Design space analysis	528
26.3	Claims analysis	533

27 Prototyping	537
27.1 Prototyping techniques	538
27.2 Prototyping to support design	542
27.3 Software prototyping tools	553
28 Software Support	565
28.1 The user interface and software tooling	569
28.2 Stand-alone tools	571
28.3 Integrated environments	579
28.4 Support tools for group working	584
Interview with Deborah Hix	593
PART VI Interaction Design: Evaluation	595
Interview with Brian Shackel	599
29 The Role of Evaluation	601
29.1 What do you want to know and why?	603
29.2 When and how do you do evaluation?	604
30 Usage Data: Observations, Monitoring, Users' Opinions	615
30.1 Observing users	616
30.2 Verbal protocols	621
30.3 Software logging	626
30.4 Users' opinions: Interviews and questionnaires	628
31 Experiments and Benchmarking	641
31.1 Traditional experiments	642
31.2 Usability engineering	650
32 Interpretive Evaluation	657
32.1 Contextual inquiry	659
32.2 Cooperative and participative evaluation	661
32.3 Ethnography	664
33 Predictive Evaluation	671
33.1 Inspection methods	672
33.2 Usage simulations	673
33.3 Structured expert reviewing	676
33.4 Modelling: The keystroke level model	685

34 Comparing methods	691
34.1 Differences between methods	692
34.2 Empirical comparison studies	701
Glossary	709
Solutions to Questions	725
References	745
Index	763

David Benyon

After graduating in 1974 David worked as a systems analyst in a software house before becoming a senior systems analyst/designer at a large Leicester-based knitwear company. He left in 1980 and joined Leicester Polytechnic (now De Montfort University) as a lecturer in systems analysis and design. In 1983 he obtained an MSc in computing and psychology and began to concentrate on the human aspect of computer systems. He joined the Open University in 1987 where he has developed distance learning courses on database systems, software engineering and human-computer interaction. He has published over twenty research articles, authored *Information and Data Modelling* (Blackwell Scientific Publications, 1987) and co-authored *A Guide to Usability* (originally published by the DTI, now available from Addison-Wesley) and *Automating Systems Development* (Plenum Press, 1988). David's research interests centre on human-computer system design and the application of knowledge-based techniques to HCI design. David led the authoring of Part IV on design methods.

Tom Carey

Tom is a professor in the Computing and Information Science Department at the University of Guelph in Canada, where he developed the first HCI course in 1982. Currently he directs the university's human-computer interaction laboratory. He was a member of the ACM SIGCHI Curriculum Development Group and was education editor of the *SIGCHI Bulletin* from 1988 to 1990. He was a speaker at a tutorial on Teachers' HCI at CHI'91 and a workshop on HCI at CHI'92, and was Tutorial Chair for INTERCHI'93. Tom's research interests include design approaches for HCI and an experimental aids for prototyping and evaluation.

devices in system design. Similarly, you also need to consider variation in user characteristics. For example, users can vary in knowledge, skills, abilities or disabilities and experience, as well as in more subtle characteristics like culture and gender. Knowing the scope of different kinds of devices is important too, because ultimately you have to match the technology with the user and work needs, not forgetting the work environment.

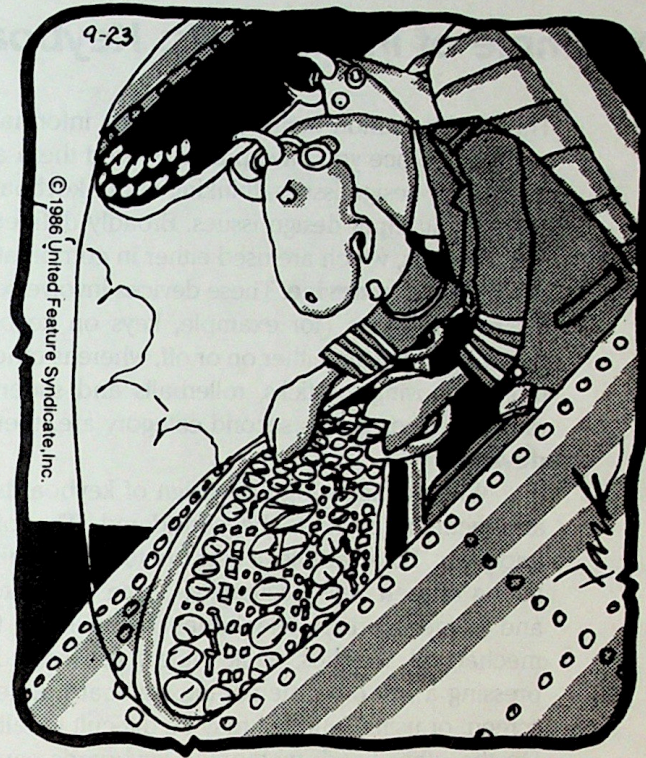
11.1 Input devices

Input is concerned with recording and entering data into the computer system and issuing instructions to the computer. In order to interact with computer systems effectively, users must be able to communicate their intentions in such a way that the machine can interpret them. Therefore, we can define an input device as: a device that, together with appropriate software, transforms information from the user into data that a computer application can process.

One of the key aims in selecting an input device and deciding how it will be used to control events in the system is to help users to carry out their work safely, effectively, efficiently and, if possible, to also make it enjoyable. The choice of input device should contribute as positively as possible to the usability of the system. In general, the most appropriate input device will be the one that:

- matches the physiological and psychological characteristics of users, their training and their expertise. For example, older adults may be hampered by conditions such as arthritis and may be unable to type; inexperienced users may be unfamiliar with keyboard layout.
- is appropriate for the tasks that are to be performed. For example, a drawing task requires an input device that allows continuous movement; selecting an option from a list requires an input device that permits discrete movement.
- is suitable for the intended work and environment. For example, speech input is useful where there is no surface on which to put a keyboard but is unsuitable in noisy conditions; automatic scanning is suitable if there is a large amount of data to be gathered.

Frequently the demands on the input device are conflicting, and no single optimal device can be identified: trade-offs usually have to be made between desirable and undesirable features in any given situation. Furthermore, many systems will use two or more input devices together, such as a keyboard and a mouse, so the devices must be complementary and well coordinated. This means that not only must an



*'Darn these hooves! I hit the wrong switch again!
Who designs these instrument panels, raccoons?'*

(© 1986 United Feature Syndicate, Inc. Reprinted by permission.)

input device be easy to use and the form of input be straightforward, there must also be adequate and appropriate **system feedback** (Norman, 1988) to guide, reassure, inform and, if necessary, correct users' errors. This feedback can take various forms. It can be a visual display on a screen: a piece of text appears, an icon expands into a window, a cursor moves across the screen or a complete change of screen presentation occurs. It can be auditory: an alarm warning, a spoken comment or some other audible clue such as the sound of keys clicking when hit. It can be tactile: the feel of a button being depressed, or a change in pressure, such as occurs when using a joystick. In many cases feedback from input can be a combination of visual, auditory and tactile responses. For example, when selecting an icon on a screen, the tactile feedback from the mouse button or function keys will tell users that they instructed the system to activate the icon. Simultaneously, visual feedback will show the icon changing shape on the screen. This is coordinated with the sound of the button clicking or the feel of the key resisting further pressure. In this chapter the various types of device are discussed in terms of their common characteristics and the factors that need to be considered when selecting an input device. In Chapter 12 we return to the issue of feedback.

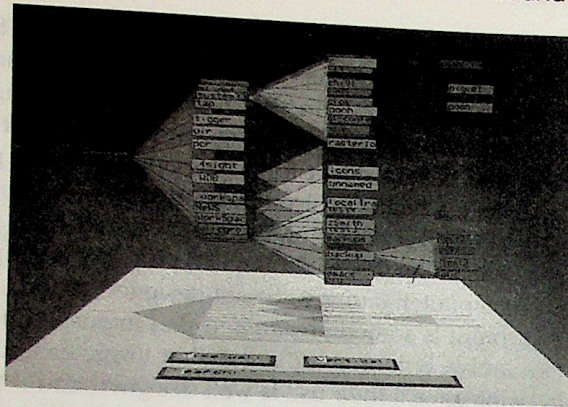


Figure 12.3 A Camtree, one of the three-dimensional animations provided by the Xerox PARC Information Visualizer (Card *et al.*, 1991: © 1991, Association for Computing Machinery, Inc. reprinted by permission).

EXERCISE

If you know a procedural programming language such as Pascal, and if you know any simple general algorithms or data structures, such as sorting algorithms or stacks, see if you can think of a way in which a three-dimensional animation could help a beginner understand how they work. Specify your animation as a sequence of diagrams showing successive stages of the animation. (No comment is provided for this exercise.)

12.4 Sound output

Current uses of sound at the interface are largely for alerting and feedback purposes. For example, various forms of beeps and bells are used to indicate that an incorrect command has been issued or that a process needs attending to. The use of sounds as warning indicators has also been extensive in process control plants. Until recently few computers could generate deliberately designed sounds other than beeps, but this has changed in the last few years. Different kinds of sound, each of which can be either synthesized or sampled (that is, digitally recorded and replayed), include the following:

- speech (see Section 12.5)
- musical sounds
- natural sounds

Some researchers, such as Buxton *et al.* (1989) have suggested that sound can be used in more information-rich ways to show what is happening in a system. In particular, it can be used as a coding method for augmenting graphical representations. They suggest that complex systems might benefit by using sounds that have a direct form of mapping to the processes they represent. Given that we have a highly evolved hearing system capable of gathering very detailed information about our environment, there should be a great potential to improve interfaces by exploiting this capability. To consider a simple illustration, Buxton *et al.* note that in computer games with well-designed sound, scores improve when the sound is turned on. It is sometimes argued that increased use of sound would make computers too noisy. Buxton *et al.* argue that in practical cases, quiet sounds are not perceived as intrusive if their use is well designed and if they provide useful information. In cases where noise pollution is a critical problem, earphones or quiet directional speakers close to the ear could be used.

Sound is of particular value where the eyes are engaged in some other task, or where a complete situation of interest cannot be visually scanned at one time. Sound can be useful for monitoring continuous background processes that are normally not important until something out of the ordinary happens, like hearing an unusual car engine note, or for drawing attention to an unusual one-off event, such as an object hitting a car from the outside. Sound can also be useful for drawing attention when expected background events fail to happen, for example, lack of disk access noise when intending to save a file to disk. These strengths of sound as a means of feedback are reflected in the following list of areas in which sound output has already proved valuable:

- Applications where sound complements a standard visual interface.
- Applications where the eyes and attention are required away from the screen; for example, flight decks, medical applications, industrial machinery, transport and farms.
- Applications involving process control in which alarms must be dealt with, or continuous monitoring is required.
- Applications addressing the needs of blind or partially sighted users.
- Data sonification. That is to say, situations where data can be explored by listening to it (given an appropriate direct mapping).
- Algorithm animation, in which understanding how an algorithm executes can be explored by listening to an appropriate mapping of its operation into sound, sometimes as a complement to visual animation.

Natural sounds

One interesting way of using sound in computer interfaces is to make use of natural sounds, either synthesized or digitally recorded. Gaver (1989) argues that evolution has adapted the human hearing system to be capable of extracting large quantities of information effortlessly from natural sounds such as rustles, thumps, hits, scrapes, tinkles, creaks, patters, reverberations, and so on.

EXERCISE

How might you use sound in an electronic mail system? Think about such things as the length and kinds of message, who they are from and so on.

COMMENT

You may have several suggestions. Bill Gaver (1989), a researcher into the use of sound in interaction design, points out that it might be useful if it were arranged for incoming electronic mail to make a sound something like a letter falling into a real mail box. Such a sound could be interpreted easily by a user concentrating on some other task. So, for example, a lengthy electronic mail message might produce a heavy thud, whereas a short message might produce a light rustle. The degree of reverberation might indicate the extent to which the mailbox was full or empty, with a nearly empty mailbox reverberating most. A metallic clang might indicate the mailing of machine code, whereas a gentler quality of impact might indicate a text file. Different mappings into sound might be preferred by individual users, for example, a deep thud could be set to indicate mail from one's boss, a brief signature tune to herald a mailing from a particular mail group (although this is taking us away from Gaver's ideas, which centre on purely natural sounds), and so forth. Box 12.2 contains another example of the use of sound to provide feedback in the Sonic Finder.

Experiments by Gaver and others (Gaver *et al.*, 1991) have also indicated that sound can be particularly useful in helping a team of users to collaborate on a process control task. For example, in a program that simulated a soft drinks bottling factory (ARKola) implemented in the collaborative SharedARK computer environment, sound was generated from a digital sampler to represent a variety of information. This included the sound of clanking bottles to represent the bottle dispenser and a splashing sound to indicate that liquid was being spilt. The rhythm of the sounds was intended to reflect the rate at which the machine was running, while the rapidity at

Box 12.2 Sonic Finder

A device called the Sonic Finder (Gaver, 1989) has been used experimentally on the Macintosh (see Figure 12.4) to add sounds to actions such as opening windows, moving, copying and deleting files and so forth. In our experience this program was very useful because the sound cues helped to avoid some common errors, such as intending to drop a file in the trash can but missing.

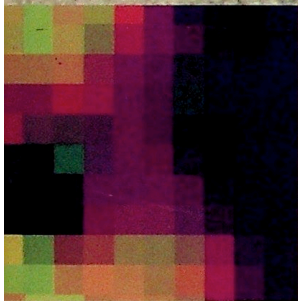
HUMAN-COMPUTER INTERACTION

Offering the most comprehensive account of the multi-disciplinary field of HCI, this book illustrates the powerful benefits of a user-oriented approach to the design of modern computer systems. It balances the technical and cognitive issues required for understanding the subtle interplay between people and computers, particularly in emerging fields like multimedia, virtual environments and computer supported cooperative work (CSCW).

A unique feature is the inclusion of interviews with many leading authorities in HCI, providing personal insight into their work and conveying the excitement of current research activity.

JENNY PREECE

- YVONNE ROGERS
- HELEN SHARP
- DAVID BENYON
- SIMON HOLLAND
- TOM CAREY



DEBORAH HIX
ROY KALAWSKY
MARILYN MANTEI
TOM MORAN
DONALD NORMAN

BRIAN SHATTUCK
BEN SHNEIDERMAN
BILL VERPLANK
TERRY WINOGRAD

Human-Computer Interaction is flexibly structured to allow a variety of learning paths for students in computer science, engineering, psychology and cognitive science. Programmers and system designers will appreciate its emphasis on the design of interactive systems.

The book has been developed from successful modules offered by the UK Open University team chaired by Jenny Preece. Her co-authors are Yvonne Rogers of the University of Sussex, Helen Sharp, David Benyon and Simon Holland (all at the Open University) and Tom Carey of the University of Guelph, Canada.



Addison Wesley Longman Limited



X000L6BE27

Human-Computer Interaction: Concepts
And Design (ICS)
Used - Very Good; ship to IND1

ISBN 0-201-62767-6