

strational interfaces and proposes these as the next step in the development of direct manipulation graphical interfaces. Such a system would monitor the user's actions and guess generalizations from his actions to increase the apparent power of the interface while not making his task any more complex. In the second invited paper, James Alty of Loughborough University examines multimedia interfaces, noting that previous studies in this developing field have tended to examine the possibilities of multimedia, rather than the users' requirements and the opportunities presented by the increased communication bandwidth. He concludes with a series of research questions for multimedia use.

Two papers from the HCI Group at York University examine multi-user interfaces and how an individual user's expectations can be supported in such an environment. Co-ordination of groupware work is examined by Pendergast and Beranek, with a description of both the problem areas and a prototype system developed to overcome them. Iconographer, an interactive tool for investigating alternative iconic object representations, is described in two papers from GIST in Glasgow, and a third paper describes the novel "Wet and Sticky" system intended to reintroduce the qualities of real painting materials into artwork produced on computers.

The first of two further papers from GIST in the section on "Evaluation" proposes a better method of HCI evaluation using focus groups rather than individual assessments of usability. The second proposes usability assessment parameters of guessability, learnability and experienced user performance, and demonstrates these by way of two simple experiments. The third paper in this section reports a large study to evaluate two multimedia training systems.

The Speech and Language Technology Group from Hatfield Polytechnic demonstrate an improved recognition performance in a speech-driven word processor by using linguistic prediction to focus the recognition domain on the user's current action. The Police HOLMES computer system was the subject of another group, with a user-centred approach to produce an improved demonstration interface for the system, with some success. Conversational turn-taking was investigated by a group from Dundee and Edinburgh, and this knowledge applied to developing a prototype text telephone for the speech and hearing impaired. An interface for the blind is proposed by a group from York University, suggesting that use of stereo audio cues may be a viable method of navigation for such individuals.

The volume concludes with two papers examining spatial concepts; the first describes research seeking to improve the usability of graphical information systems and the final paper describes a hypertext prototype based on the spatial metaphor.

I. R. MURRAY
Dundee

JOHN J. DARRAGH and IAN H. WITTEN

The Reactive Keyboard. Cambridge University Press. 1992. 0-521-40375-8. £25.00 hardback

There are a number of people who have physical conditions which, in addition to a general physical disability, render them permanently unable to communicate using their voice. It would seem that computer technology has a great deal of potential help to offer such people: with a computer-based system the user can control a powerful and versatile system with only a small amount of movement. The reality is that computer-based communication systems thus far developed are a certain amount of help to non-speakers, but they as yet only allow them to achieve communication rates far below that of unimpaired speakers. (Typically 2-10 words per minute, compared with a normal rate of 120-200 words per minute.) Physically impaired people who are able to make use of their voice face this same rate barrier when they wish to access computers.

There is clearly the need for research into methods for improving this communication rate. A general point: devising ways of helping severely physically impaired people use computer systems is an endeavour which is obviously of benefit in itself, but it also has the potential to give us insights and information which can be more generally applied in interface design.

The task here is, given a ceiling on the input rate which is imposed by the user's impairment, to maximize the result of each input action. One technique is abbreviation expansion. Another is a coding system for storing and retrieving words or sentences. A third method is prediction: attempting to predict what the user is trying to say and completing the task for them.

This book traces the development of a predictive typing system, originally designed to help physically impaired users of a computer system save keystrokes (or the equivalent) in entering UNIX commands. Darragh and Witten begin with an introduction to the problem of communication disability and technical systems which have been developed to help. They devote a separate section to an analysis of predictive text generation systems, both commercially available systems and research prototypes. An earlier prototype of their Reactive Keyboard is described in this section.

The main part of the book is a detailed description of the Reactive Keyboard program. The authors describe the system in terms of its user interface, the predictive mechanism, and its implementation in code. A complete "C" code listing is given in an appendix, since the authors have made the program freely available and invite others to use and modify it as they wish.

The predictive method used is a stochastic model at the character level. While this method can produce very good results, particularly with strings of characters which are used often, as with command sequences to an operating system, one problem is that it can produce plausible but nonsensical strings. The authors' solution

has been to let the user select as much of the predicted string as is correct, by using Control-key commands or a mouse.

The evaluations of this system have thus far been informal but encouraging. The only formal trials reported were done with able-bodied students to test an early prototype. Results of this were used in the design of the final version. The usefulness of the Reactive Keyboard program itself is indicated by the fact that it has been used for some time as a command interface to UNIX by disabled university students and some very positive anecdotal evidence. It would have been helpful in judging the success of the Reactive Keyboard to have more formal evaluations of it in use.

This book reads as a case history of the development of a new software system. It has the immediacy and practical interest of a case history. The computational techniques employed are not new, but are here applied to a difficult and worthwhile problem. The attempts to generalize and put the work into context are less convincing. They have the feeling of being just added on, rather than informing the work throughout. For a reader new to the subject, however, these sections provide a useful introduction and overview.

N. ALM
Dundee

DERRICK GROVER

The Protection of Computer Software – its Technology and Applications (second edition) Cambridge University Press, 1992. £14.95. ISBN 0 521 42462 3.

This book has been written to advise authors of computer software how to prevent illegitimate use of their products. At the technical level, it addresses itself predominantly to PC applications and in particular to the IBM PC. Questions arising from the networking of machines (PCs) are also considered. The book is now published in its second edition. The major addition is a new chapter on computer viruses, due to increased occurrences and increased awareness of this threat to users of computer software; elsewhere, the book says little about protection of software from the users' point of view.

The book is a collection of essays covering a spectrum of means of protection that ranges from mechanisms for copy protection to legal deterrents, written by experts in the respective areas. The chapter on disk-based protection methods gives a lucid exposition of the arms race between the copy protection and copy programs. In the opinion of the author of this chapter, copy protection

frequently imposes restrictions that a user should not reasonably accept, which may also explain why this approach has become somewhat unfashionable. Two chapters, dealing with physical protection devices and intelligent modules, discuss hardware that can be employed to control usage of software. This approach avoids the inconveniences of copy-protection mechanisms, but may introduce some of its own. Intelligent modules can provide high security, usually at a price, which means that the value of the software to be protected has to stand in a proper relation to the cost of protection. Software protection may use cryptographic methods for enciphering programs that are then stored publicly and, as in the case of intelligent modules, deciphered and executed only in a special device. In addition, cryptographic methods like message authentication codes or digital signatures allow investigators to detect whether a file has been modified or to identify the originator of a file. Both mechanisms may be found in virus-protection software. These issues, a discussion of key management, and a description of the popular algorithms DES and RSA, are covered in the chapter on cryptography. The chapter on hacking and the new chapter on viruses give an overview on threats, both to authors and users of software, and advice on appropriate countermeasures. Ironically, by being a threat to users, computer viruses have helped the cause of authors by discouraging illegal copying of software. The chapter on licensing and the extensively revised chapter on legal protection discuss contractual and legal issues of software protection in the UK and abroad. This is an ongoing concern with the current attempts at defining a common European position on intellectual property rights for software. The chapter on program identification explores means of providing evidence of copying, like fingerprinting, birthmarks and correlation tests.

In all chapters, the authors examine the scope and efficacy of the respective means of protection. They strike a laudable balance between good advice, which seems sometimes so obvious that it could become boring, and technical detail, which makes some very entertaining reading. The only regret while reading the book was due to the fact that not all chapters had been updated from the first edition. This was felt most acutely with some hardware issues, where it would have been interesting to see how some ideas suggested five years ago had fared since. However, this should not detract from a very useful and comprehensive survey on the protection of software written for PC applications.

D. GOLLMANN
London