strength in the system specification stage, while other HCI design techniques tend to focus on the system evaluation stage of the life cycle. Other approaches to HCI design are not necessarily excluded, but they are considered to be seriously incomplete.

The MUSE methodology is concerned primarily with the development of large systems, typically those involving multidisciplinary design teams. In the context of small projects with a single designer the use of MUSE might be regarded as overkill, but even here the well documented design solution which MUSE facilitates may still be appropriate.

The book is aimed at software designers with some understanding of existing HCI principles and techniques. It is essential reading for any HCI specialist who is interested in the formal integration of human factors into the system development life cycle. It is also highly recommended for software engineers who use structured methods. The well-established methodology which the authors have selected to illustrate MUSE is JSD. The authors are attempting to assimilate human factors considerations into JSD, rather than to extend JSD to encompass human factors. Thus the desirable characteristics of JSD and the well-established JSD techniques are not affected by the authors' enhancements. This work is not the first to attempt this synthesis of methodologies, but is does seem to be the most complete so far.

Does the method work? The reviewer has not implemented any large systems during the review period, so the method has not been put to any practical test. However, there is a great deal of good sense in the MUSE approach. It attempts to find a sensible balance along the general–particular scale, i.e. avoiding both under- and overdesign. It attempts also to give a complete coverage of HCI issues in the design of systems. The authors succeed to quite a large degree in their aims of capitalizing on well-established techniques, and promoting communication between HCI designers and more traditional software designers.

The book includes a useful case study of the application of the MUSE method for network security management at University College London Computer Centre, and this provides a good illustration of the power of the methodology.

Both HCI academics and HCI practitioners will find this book thought provoking and genuinely useful. Most will not agree with everything written here: that is the price paid for evincing a distinctive point of view and attempting something of genuine practical value. Some may be dismayed that their favourite HCI sub-topic is not mentioned here, but the book is not intended as a general HCI text. However, all should enjoy the lucidity of the text, the well structured development and the explicit expression of the key ideas. The attention to detail is thorough. The diagrammatic notations are generally clear enough for use by non-HCI specialists and sophisticated users. Anybody already familiar with JSD will find this book doubly useful, but readers who do not use JSD need not be deterred.

J. Rosbottom
University of Portsmouth
Department of Information Science

Dermot Browne
STUDIO—Structured User-interface Design for Interaction Optimisation. Prentice-Hall. 1994. ISBN 0 13 014721 4 £22.95. 291pp. softbound.

The book addresses a neglected area, that of user interface (UI) design. The content affords the user interface the respect it deserves, given that so much resource is spent on coding and maintaining it. Browne quite rightly points out that the UI is often of as an afterthought or something that is tacked on after the other parts of the system have been designed. The idea of treating UI developments as a separate activity from systems development is a good one, particularly for client–server applications.

The introduction provides a good 'route map' for navigating through both the book and the STUDIO method. Relevant anecdotes and a continuous case study of financial dealing software are used to good effect throughout the book to communicate the importance of a systematic but user-centred approach to design.

The book challenges the complacency of those IT professionals who assert that to achieve a quality user interface requires only that '*style guidelines are followed unerringly*'. Browne uses anecdotal evidence to show that if rules are followed slavishly without understanding, they are likely to be applied inappropriately. His systematic treatment of user requirements analysis as a crucial precursor to the generation of user interface designs reveals the true complexity of user needs and behaviour. The case for thorough documentation and understanding not provided by conventional systems analysis is well made.

By example, rather than exhortation, STUDIO sets about showing readers (it is hoped, software developers), the procedures and benefits of a user-centred approach to UI design. This is a most valuable contribution. The reader, after exposure to the processes involved in thorough exploration of user requirements, is shown how to the generate user interface options, to use prototyping and impact analysis to reject inappropriate options and finally how to firm up and develop the relevant ones. The handover from user-interface designers to other developers also receives due attention, one more filling a significant gap in available guidance.

While STUDIO has a great deal to offer it also has some unfortunate limitations. For instance, the introductory example based on the HOLMES system for the Home Office is perhaps not as convincing as it could be. While it supports the need for user-centred design and shows that after *one* day new users were as productive as trained existing users, much of this benefit seems to accrue simply from the use of a graphical user interface allowing WYSIWYG document display. No evidence is provided to support the view that it is the structure and design of the UI *per se* which had promoted productivity.

A further discomfort with the book is that Browne criticizes, quite rightly, design principles because they offer little in the way of real guidance to designers, but then he proceeds to present STUDIO's design principles which are more of the same sort of thing. This is followed by a section on user interface options which seems strangely at odds with the highly structured layout of the rest of the book (and methodology). In this section he considers haphazardly topics such as colour, windows, tailoring, undo/redo, response times, menus, etc. This is all useful advice but lacks a unifying user-centred framework consistent with the philosophy of the method.

Another concern is the misleadingly superficial treatment of research techniques such as Keystroke Level Modelling and Semantic Differential Scaling. Rather than risk inadvertent misuse of these techniques in unskilled hands it would be better to suggest that expertise in these research skills is sought.

Finally, the most serious limitation relates to the treatment of prototyping. Browne introduces the notion of throwaway versus evolvable prototypes but does not really give any arguments for the use of one over another. User involvement in evaluating 'prototypes' in earlier STUDIO stages appears generally to be restricted to paper screen designs and various diagrammatic notations. To be genuinely user centred, PC based prototypes should be used at all stages, even the initial concept stage, as a way of extracting user requirements and exploring new ways of working. Only through 'hands-on' experience of working prototypes can users contribute their ideas fully to the development process.

Despite its shortcomings this book's real value and significance is that it succeeds in putting User Interface Design firmly on the agenda as an important area of expertise in its own right. Moreover it reveals that systematic user-centred methods for analysis and design are now well developed and have been tried and tested in real development environments. The book is a thus a welcome addition to the literature and includes many good references, unusual for a 'proprietary' methodology. The material it contains will be particularly helpful for dedicated systems where a solid task analysis can be documented as user procedures. Whether it will be as useful for generic applications is less certain.

Leela Damodaran
Loughborough University of Technology

Jan Hardenbergh
Building Applications with PEXlib. Prentice Hall. 1994. ISBN 0 13 012535 0 £36.50 540pp. softbound.

PEX is, as you may or may not know, a set of 3D graphics extensions to the X system. This book sets out to teach the use of PEX and to act as a basic reference to the PEX system.

Along the way is built up the 3D art package named/em PEXDRAW (a 'canonical' PEXlib application, used to test PEX servers); there are therefore plenty of examples of use of the functions presented along with their definitions, which considerably helps those beginning to use PEX.

The documented calls most strongly featured are those which are actually used in /em PEXDRAW: the others are documented but are not presented in as much detail. Since these are mostly more advanced functions, the reduced information is not so much of a problem—when one has reached the point of using them, the PEXlib concepts should be much clearer.

The render pipeline is discussed in general first, followed by the various sections of PEX stages, in (roughly) the order one needs to know them to get PEX to work, in more detail. These sections are covered very well—there are some complex topics involved in PEX and they are described clearly enough that even a newcomer to computer graphics will find them easy to understand while not being 'dumbed down' enough to irritate those with more experience.

The book also forms a reasonably good introduction to three-dimensional graphics, even if this is not its primary purpose. It is not the only book you will need, but a book which concentrates on the implementation of graphics in practice makes a great companion to some of the more academic works.

The end of the book consists of various appendices, including a very handy 'PEX lexicon' which explains the terminology used in the rest of the book, a complete list of the PEX output commands and (possibly most important for a useful reference) a good, detailed, well laid out index.

The book is much more readable than many such works, and does form an ideal introduction—while not losing the ability to form a desktop reference guide after one has really got into the subject.

K. M. Lucas
Warwick University

Jeffrey S. Rosenschein & Gilad Zlotkin
Rules of Encounter. MIT Press. 1994. ISBN 0-262-18159-2 £31.50. 229pp. hardbound.

The information age is upon us. In the minds of many, the future is filled with intelligent agents making deals on our behalf, while we relax, drink in hand, in front of the video wall. However, in *Rules of Encounter*, the authors argue that the future will not necessarily be this rosy. It is