

**Developing expert systems: a knowledge engineer's handbook for rules and objects** by E. C. Payne and R. C. McArthur, John Wiley & Sons, USA, 1990, pp 401, £27.95, ISBN 0-471-51413-6.

According to the Preface, this book is described as being in keeping with the philosophy of the AXLE (Albathion eXpert system Learning Environment) from the same authors. The two are intended to provide a stepwise description of building a particular expert system and its possible variants. This book has been reviewed without the benefit of the software.

In the Introduction the primary aim of the book is said to be combine theory and practice to demonstrate how a real expert system is built. Accordingly, the book is structured in chapters following the natural pitfalls and progress of the development of an example system. This treatment primarily discusses the solutions to practical problems, and uses these as a means to describe the underlying principles. In this respect, it may offer a shortcut to practical experience for the novice in this field.

The example chosen is a case study in a canning-factory, following the implementation of a frame-based, object-oriented expert system for production line fault diagnosis. The system's evolution is described from prototype to first installation, through multiple lifecycles, validation and the delivery of an installed system, in chapters illustrated with frame descriptions and samples of factory-floor consultant-expert interaction.

There are discussions of the use of a particular expert system shell, the interpretation and translation of expert knowledge and knowledge representation (in this case using frames for objects), and forward-chaining inference engines using rules. End-user considerations such as graphical interfaces, their basic components and popular forms are described for those who have not experienced them. Topics such as backward-chaining rules for problem solving, the representation of connected object and rule priorities form the third section to the book, and the need for graphical model editors, system customization and simulations for model-based reasoning in the real world complete the book. The numerous examples of frame implementations presented throughout use HyperTalk, and commercially available shells for system development are mentioned but without much critical comparison.

The audience for whom this book recommends itself are students of AI, corporate persons responsible for expert system implementation, managers and interested individuals. The book initially assumes too little knowledge for the reader to be a knowledge engineer, despite its claim to being a "knowledge engineer's handbook for rules and objects", and it appears to be most suitable for the non-technical, managerial reader. For such a reader there is description of the rationale behind the use of expert systems, assessment of their appropriateness and the required analysis of problems they are used to solve. Practical matters are set in the context of political and financial considerations of their implementation, management champions, and delivery timescales. Serendipitous, additional applications for expert systems are included to increase their appeal, as discovered in real life situations. In contrast, the use of material in later sections including examples of rules, frames and simulations might appeal more to students and implementors in the area.

The text itself is highly structured, punctuated with chapter summaries and frame examples as well as anecdotes of experts and figures from operational systems. These, plus the quotation of rules, frames and slot names, are all distinguished by different fonts and emphases which do not combine for well-set type. The explanations of frame and rule usage are sometimes obscured by examples which are too domain-specific (inevitable using a case study), when the general case would perhaps be easier to present. In general, however, there is sufficient emphasis and repetition throughout the text and examples to convey the concepts and provide the demonstration as the authors claim.

This book provides a basic guide to the world of issues surrounding expert systems as well as available commercial shells and the construction and operation of expert systems themselves. As an introductory text to a new world of industrial and commercial problem-solving it makes alluring reference to the like of NASA life-support systems without indicating how these differ from the case study in complexity. There may not be sufficient detail in the technical content to satisfy the

reader enquiring at any more than a superficial level. So many topics receive a mention that few, but the obvious, receive a satisfactory explanation. Perhaps precisely because this books sets out to treat so many of the implementation issues for so many readers, it may not maintain the interest of any of the intended audience for all of its duration.

Reviewed by Catherine Hearne, Advanced Computation Lab, Imperial Cancer Research Fund, London, UK

**Computers and conversation** edited by Paul Luff, Nigel Gilbert and David Frohlich, Academic Press, London 1990, pp 284, £14.50, ISBN 0-12-459560-X.

This book is an attempt to bring the findings and methodology of conversation analysis to bear on human-computer interface design. The book, one in Academic Press' "Computers and people series", grew out of a symposium held at the University of Surrey in 1989. It is aimed primarily at those working in human-computer interaction (HCI), and includes both theoretical and applied papers, most of which are well written.

Conversation analysis (CA) is a research discipline with strong sociological roots. It is concerned with the organization of conversation in everyday interaction, including the structure of conversation openings and closings, and descriptive rules governing turn-taking, change of topic and repair. An introductory chapter by Robin Wooffitt makes the book accessible to those with little or no background in CA. Wooffitt both emphasizes the sociological aspects of CA and illustrates its findings with examples from the structure of lists of an analysis of teasing. Further useful introductory information concerning some of the findings of CA arises throughout the text (in, for example, the contribution of Norman Thomas and Button), and notes on transcription are included as an appendix.

The book lacks a clear organizational structure, and the sound introduction is followed by what I found to be a confusing discussion of the relationship between HCI and software engineering. In "Towards a sociology of human-computer interaction: a software engineer's perspective", Hugh Robinson argues for a sociology of HCI based on ethnomethodological ethnography. The chapter does not flow neatly from the previous introduction, with the link to CA being weak and only explored in the latter sections of the paper.

The track returns to firmer ground with a discussion by Norman and Thomas of the roles that CA can play in HCI design. They emphasize that both the methods and the findings of CA have applicability in HCI design: human-computer interactions may be constructively analysed with the methods of CA (i.e. in the context of conversational openings, closings, repair, etc.); and the general findings of CA (such as, for example, on the expectations of people in the structuring of repair conversations) can inform HCI design. The authors admit that the insights from CA as presently formulated are no better than previous HCI guidelines, but hope that this could be restricted by an adequate framework describing HCI.

A note of debate is introduced to the volume via Graham Button's contribution, "Going up a blind alley: conflating conversation analysis and computational modelling". Button argues that it is inappropriate to use the rules developed within CA as rules within dialogue systems to control human-computer conversations. The arguments are reminiscent of those put forward by some connectionists arguing against symbolic accounts of cognitive processes, and might be summarized by claiming that the rules that CA develops correspond to emergent regularities. They play no causal role, and as such computational models which follow such rules are misguided. Button's point seems valid, but is blunted by the earlier remarks of Norman and Thomas, who acknowledge that the role of CA in HCI is not to mimic human conversational behaviour, but "rather to provide for the design of interaction in ways which are 'in accordance with expectations' about interaction" (their emphasis). The cautious line is continued by Paul McIlvenny, who complains that "research in empirical investigations of communicative action" has disembodied conversation analysis, and