Book Reviews

L. M. G. Feijs & H. B. M. Jonkers
Formal Specification and Design. Cambridge University Press, 1992, £29.95, 335pp hardbound, ISBN 0 521 43457 2

Gradually 'Formal Methods' are being accepted as a 'good thing' but as yet they have not made a significant impact on the production of software. In principle the use of formal methods means that 'correct' programs can be shown to be correct (with respect to an agreed specification), or alternatively that programs can be derived in such a way that they are guaranteed to be consistent with a given formal specification. For a program/system to be correct all its necessary parts also have to be correct. Ideally an initial specification should only indicate what is to be done, not how it is to be achieved (i.e. it should not suggest a particular design) and hence is, of necessity, abstract. Consequently, the gap between an initial specification and a final implementation is conceptually BIG. This is often cited as the reason why formal methods have not been adopted as readily as some of us would have hoped.

Various attempts have been made to reduce this gap and thus make it easier to bridge. The use of VHLL's is one approach that has been tried. Another, is to require that the specifications are written in such a way as to make the extraction of a correct design (for an implementation) relatively automatic. This latter approach can be criticised—it injects a considerable 'how' component into what the purists argue should be a 'what' specification. Nevertheless, any piece of research that helps to 'get it right' more often should be welcomed as should any book which makes this work accessible to a wider audience. (Although it is nowhere clearly stated, I presume that the target readership is not confined to the computer science research community who, via journal publications, technical reports and conference papers, have already had sight of much of this material—and, indeed, more up-to-date variants—albeit in a more condensed and less discursive form.)

Essentially the book is about the language COLD. *Common Object-oriented Language for Design.* a language developed, as part of an ESPRIT project called METEOR, by Philips in Eindhoven. The language incorporates both the algebraic and state-based styles of specification as well as facilities to assist in the description of designs. But beware, the algebraic specifications, though superficially similar to those of OBJ etc., do not give rise to *initial* algebras. Instead, they require inductive or algorithmic definitions and axioms which disallow certain interpretations rather than rely on (implicit) minimality. Similarly, the state-based specifications do not look at all like Z or VDM, but include axioms that indicate the required properties of the operations being specified.

The book is appropriately organised in three parts; the first being devoted to algebraic specifications, the second to state-based specifications, and the third to more advanced features of COLD and associated theory. It is based on courses given to graduate students in Holland and assumes some familiarity with the mathematical notations and terminology used in formal computer science. This may 'throw' the ill-informed reader; this is a specialist book, but very readable by anyone familiar with the subject area and who wishes to know about COLD.

Some aspects could be explained more succinctly—BNF could be usefully employed in several places—and elsewhere there is an unfortunate overloading of notation (here *p*! does not mean *p* factorial, or that *p* is unique, or that *p* is an output variable), but these are minor criticims and do not detract from the flow of the presentation. The style of the COLD definitions and axioms provides a nice link with computability theory and, swimming against the tide, reinforces the case for retaining the more theoretical aspects of computing in our courses. A graduate computer scientist, or someone of an equivalent professional standing, who is involved with software production should find this book both readable and interesting.

D. J. Cooke
Loughborough

Steve Scrivener
Computer-Supported Cooperative Work. Ashgate Publishing. 1994. ISBN 0-291-39812 X. £45.00. 286 pp. hardbound.

Recently, a number of computer products and services have been announced that claim to offer support for collaborative and group work. These range from systems for structuring and coordinating formal work processes through to video facilities for enhancing informal communication between personnel. The design and evaluation of such technology has been the focus of much academic research in the last few years, generally collected together under the term Computer-Supported Cooperative Work (CSCW). With the imminent possibility of deploying collaborative technologies within organizations, it appears timely to bring together the relevant experiences of CSCW researchers and report these to a more general audience. This was the intention of a seminar held in London in 1992. Following from this meeting, Steve Scrivener has compiled a varied and lively collection of contributions on CSCW.

CSCW is a very broad topic that includes research on work practices by sociologists and anthropologists, new computer architectures and network applications by computer scientists, and teamwork by social psychologists. Therefore, drawing together a general collection of papers on the subject presents considerable difficulties for editors. Either they can focus on a particular area or orientation and neglect a substantial amount of relevant work, such as in the collection by Galegher *et al.* (1990) and the later books in the Springer-Verlag series, or they can present the research as a set of unstructured readings (e.g. Baecker, 1993; Greif, 1988). This collection has adopted an approach that is something of a compromise: a lot of short papers largely focused on one topic—multi-media video-conferencing systems. However, even on this subject there is already a large and dispersed literature including work by telecommunications engineers, computer scientists, psychologists and sociologists. Four chapters of largely introductory material tend to drift between distinguishing previous work chronologically and technologically, between when it was done and what it does. As an exception, the chapter by Gale concisely provides technical details, including descriptions of standards for video conferencing, available speeds of transmission and actual prices of equipment.

The majority of the remaining chapters report research undertaken in Europe on various issues associated with multi-media systems. These include descriptions of the various tools that can be provided to support collaborative work, possibilities for interface design and the computational architectures necessary for the system development. Unfortunately, by including a large number of chapters each contribution tends to be brief, resulting in little discussion of the advantages of these systems or the difficulties individuals have using them. The few evaluations are derived from small-scale experiments, sometimes by the developers themselves and are generally reported anecdotally. Moreover, the results tend to be rather ambivalent concerning the success of the technology. It is particularly problematic to assess the effect of collaborative systems on group work and organizations; however, a discussion of the use of such a system in a real-world setting would have been useful, particularly for those considering deploying the new products and services. Although the chapter by Land directly addresses problems related to the introduction of technology, his contribution discusses Information Technology in general. As the other contributors repeatedly emphasize, deployment of collaborative systems raises novel and additional problems.

At present, there is a further difficulty in collecting together research in CSCW. The continuing technological and methodological developments in the field means that papers can become out-of-date very quickly. Unfortunately, little modification appears to have been made to the papers in this collection since the original seminar in 1992. This means that recent related developments in Japan, the USA and Europe are not referred to and this leads to some curiosities. For example, Cullen reports the aims and goals of the European RACE I research programme at a time when the subsequent RACE II programme is coming to an end. Finally, given the time the book has taken to be published and its price, it is a shame that the production quality is so poor, with a large number of typesetting errors, some of which are rather distinctive.

A. PAUL LUFF
University of Surrey

REFERENCES

Baecker, R. M. (eds) (1993) *Readings in Groupware and Computer-Supported Cooperative Work*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Galegher, J., Kraut, R. E. and Egido, C. (eds) (1990) *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Greif, I. (eds) (1988) *Computer-Supported Cooperative Work: A Book of Readings*. Morgan Kaufmann, San Mateo, CA.

PHIL PICTON
Introduction to Neural Networks. Macmillan. 1994. ISBN 0-333-61832-7. £13.99. 168 pp. softbound.

The aim of this book is to introduce neural networks to the technically-minded general reader. It opts for a largely routine organization with chapters on topics such as adalines, perceptrons, Boolean nets, associative memory, probabilistic networks and self-organizing nets.

The title of the first chapter is a promising one: 'What is a neural network?'. Unfortunately, the author fails to provide the general reader with a sure footing. He asserts that neural networks are pattern classifiers but also that pattern classification is just one operation of a neural network. The fog had not cleared by Chapter 5 where we find that a neural net can do other things besides pattern classification but (*errm*) 'as a kind of consequence of its pattern classifying ability'.

For neural networks the excitement really ain't what they do, but the way that they do it. The author sows seeds of confusion for the general reader here too. One thing which sets neural nets apart from other classification systems is their ability to respond intelligently to noisy input and to patterns not previously encountered. The author indicates that this is so, but says that an Adaline would 'behave unpredictably' in the presence of a yet-unseen pattern, surely losing the sympathy of his engineering audience at this point. When the term 'graceful degradation' is introduced there is a weird restraint in the exposition, as if the author had heard of, but didn't really believe, the claims '*it is supposed that* if part of the network malfunctions, the whole system could still continue to operate' and the even weirder '*it is said that* this is preferable to complete failure' (emphasis added).

There is much to bemoan in the rest of the text. There are technical weaknesses which are most apparent in the material on Hopfield nets. For example, the author asserts that asynchronous updating is essential for the correct operation of a Hopfield net, whereas synchronous updating is equally acceptable. The author continually speaks of the existence of local minima in