Book reviews

Expert systems—strategic implications and applications by Annabel C Beerel, Ellis Horwood

I was once given some good advice on making a speech. The advice was to write the speech out in full and then throw away the first two pages, on the assumption that most of the introductory remarks would contain very little of real value!

I would say that this advice could well apply also to writing books. In the first section of this book (in fact one third of the whole book) there is much of interest on general management concepts, such as the need for vision, commitment, consistency, motivation and values, but not a great deal to take us very much further in the actual use of expert systems in industry.

This is perhaps one of the general problems of the field—articles and books on expert systems are long on theory and description and short on practical examples and advice on "what to do tomorrow". It is almost a self-fulfilling prophesy—the more you talk in rather rhapsodic terms about the abstract principles involved, the more it would be an anticlimax to show a small, sensible and productive system in action. "Is *that* what all the fuss is about?" the reader might say.

There are, however, parts of the book which become a little more detailed and (to my mind) more directly useful. For example, in the second section at page 96, there is a list of potential applications of expert systems. These are grouped under subheadings as follows: strategy and policy, management control, industry, insurance, banking and finance, office automation, general business and training programmes. Each of these gives a list of types of subject matter which might be the core of an expert system.

For example, under the subheading of "insurance", the book lists evaluation risks, assessing claims, underwriting and broking; these match some of the topics covered in various real projects quite closely and could provide the basis for some good examples. The reader who has followed with interest the general descriptions so far is by now surely keen to know more about what exactly has been done in any of these projects, or what *could* be done, but is in fact doomed to disappointment since no further details of any of these potential applications are in fact given.

Indeed, in the whole book, there is not a single example of an expert system in the sense either of showing a few lines of code, or of demonstrating the type of question and answer session which might be generated for the eventual user. At one point, the author directs the reader to Donald Waterman's *A guide to Expert Systems* for examples of expert systems which, whilst an honest acceptance of the lack of examples in her own book, is hardly very helpful for the reader now half way through the current volume!

Here we have the nub of my reservations about this book. The person who buys it presumably does so because he or she knows already that expert systems are receiving a lot of attention at present; that they are a new type of software; that they provide answers to practical questions which otherwise would require a human expert to be available and that they require a considerable investment in financial and staff terms in order to be useful in a commercial sense.

The reader would surely be hoping to get a better understanding of what expert systems actually *are*, in order to plan a corporate strategy which uses them. For this you surely need some practical examples. This really ought not to be too difficult, since one of the claims of expert systems is that they contain knowledge explicitly—in other words, in a form which can, given just a little grasp of whatever syntax is involved, be "read" like a book.

There are plenty of examples around in the literature which make immediate sense to the reader; the British Nationality Act; assessment of a firm's right to claim statutory sick pay from the Government; instructions to an employer on the procedures to be carried out when taking on a new employee; advice to farmers on which fertiliser to use on different types of crop. The authors of these systems will generally allow the use of a small chunk of code to demonstrate the concepts behind

their expert system and the language, or shell, in which they are written, and, in my view, one example of this kind is worth 50 pages of general description.

Let me, however, finish with some rather more positive remarks about this book. If you already understand the practical nuts and bolts of expert systems, you will certainly gain from the broader strategic overview which the author presents and will be able to compare the competitive advantage which one type of expert system might provide over another.

If you are interested in management strategy in its own right, the book contains much of value as well; for example, several sections are devoted to the management concepts involved in handling change and innovation in general, which is of considerable interest.

Delia Venables is a Fellow of the British Computer Society and a committee member of the Expert Systems Specialist Group. She has worked in systems analysis and operational research and is now an independent computer consultant.

Algorithmic methods for artificial intelligence by M Griffiths & C Palissier, Kogan Page 1986
Formal methods in artificial intelligence by J-P Delahaye, North Oxford Academic 1987

Writers and philosophers have toyed with the idea of artificial intelligence for centuries, but it was only with the advent of computers that the idea was discussed in computational terms.

Modern artificial intelligence (AI) could be said to date from Turing's work. In his *imitation game* a person would interact remotely with something and would have to guess whether it was human or machine. If the person was not able to tell the difference, then the machine could be said to be intelligent. The aim of the researcher and implementor of such a machine is to discover what humans can discuss, and then emulate this activity. This leads to the direction of research which could be described as a replacement strategy. For example, early work in the more specialized field of expert systems attempted this approach—to replace experts. It also implies that the machine will be indistinguishable from a human and hence that we must assume that humans can only do that which is done by computer. For whatever reason, computer models are now used to test theories of the mind—a large part of the work of people in cognitive studies.

Turing's test can also be interpreted as an injunction to design and build something and only then discuss what properties of intelligence it has, rather than merely sitting around philosophizing about intelligence. In this way it fits more into our daily work patterns. Most computing professionals, whether academic or not, whether working in AI or not, are involved in creating usable and useful systems. To do this we need to explore what computers can do and what humans can do, and create systems which appropriately mesh these two components. The aim is hopefully, to enhance human lives.

From this second viewpoint, Turing and Church's earlier mathematical work is most important. For they discussed the mathematical constraint on computers. This is expressed succinctly in Delahaye's book:

> The only problems that a computer program is able to solve are the *decidable* problems, that is, problems that can be expressed as a decidable predicate; and this term has a strict interpretation that is independent of the programming language.

Delahaye takes us along the path followed by older courses in logic and computation. Through decidability and recursive functions to formal systems and then propositional and predicate logic, that is, logics that can be used to model collections of simple statements. Thus "only criminals commit crimes" would be written as $\forall\, y\, \forall\, x\, ((cr(x) \wedge co(y,x) \rightarrow cm(y))$, where cr(x) is an abbreviation for "x is a criminal", co(y,x) is an abbreviation for "x commits y" and cm(y) for "y is a crime". There are several important issues being discussed here. Firstly, notation used for abbreviating logical statements whether used formally or informally, the intuitions that a particular logic system is trying to represent, the syntax of a particular formalization, the semantics of that formalization