

bibliography includes 184 references up to April 1985, which are in addition to others quoted in the text; 44 are by T.B. Cross, 17 of these being apparently unpublished presentation addresses. An additional feature is a listing of AI resources, principally but not entirely tool vendors; the single non-US source is the Turing Institute, of "Glasgow, Ireland" (p 247). The index is quite satisfactory. The general standard of production is excellent (only 6 typographical errors).

This book may be of interest to management information systems strategists (primarily for its discussion of knowledge networking), but this reviewer cannot recommend it as worthwhile serious reading for knowledge engineers. The argument is not with the idea that future strategic issues of information technology and computer communications are irrelevant. Current awareness in this fast-changing world is vital, of course. The problem is that Cross does not appear to be either sufficiently accurate, precise or up to date in his information, in areas familiar to knowledge engineers, to engender the reader's confidence to persevere through the jumbled structure of the book. There are far too many statements lacking in meaning to allow the reviewer to recommend more than perhaps scanning a library copy.

These two books in fact tackle "knowledge engineering" from radically opposing perspectives. Greenwell is concerned with the actual practicalities involved in the earlier stages of building expert systems. Cross takes a strategic view of a much broader area, from a height at which Greenwell's subject is scarcely visible as a distinct issue. This is very much a superset of that defined in the glossary, whereas Greenwell confines his attention to a subset of his definition of knowledge engineering. Greenwell deals with a small, well-bounded area and does it well; Cross attempts a large, ill-defined domain with less successful results. The fact that the latter chooses to write about a much broader interpretation of the term "knowledge engineering" than that which is conventionally held should not be taken as a criticism in itself, however. It does raise the valid point of whether confining the term to the building of expert systems is still appropriate; particularly now it is increasingly recognized that this is one of a family of related and in practice integrated technologies. Even more closely should we consider whether "knowledge engineering" ever has been a suitable appellation for the art of knowledge acquisition. The reviewer is inclined to propose that one who is "skilled in knowledge" should be described as a knowledgician, just as a logician is skilled in logic.

References

Neale, IM, 1988. "First generation expert systems: a review of knowledge acquisition methodologies", *Knowledge Engineering Review* 3(2) 105-145.

What every engineer should know about artificial intelligence by William A Taylor, MIT Press, 1988, pp 331, £19.95.

I found the title of the book very attractive and I really wanted to know what was it that every engineer should know about Artificial Intelligence (AI). The book begins by giving some reasons why engineers should consider AI, but not a lot of the rest of the book is specific to engineering. In fact it can be read profitably by anyone who wants to gain an overview of AI and access the relevance of the technology but is not interested in reading very technical material. The book is written in a style that is very easy to read and it does not assume much computing background from the readers.

The author presents a very practical view of AI. He writes, "For engineering purposes artificial intelligence is a few software ideas that work well enough for commercial use". The few software ideas include different programming styles, languages, and advanced programming environments that stemmed from AI research. The author does not advocate a particular style or language. His own view is that "I have been in engineering long enough to know that no single tool, no matter how wonderful, can perform *all* jobs well. I am happy to adopt ideas from logic programming or object-

oriented programming or function-based programming, ...” Besides being practical, he is also personal. The word “I” appears in the text from time to time, as the previous quote illustrates. There are a few places where I do not fully agree with his conclusion or relate well to his experience. For example, he writes that “Productivity in a good Lisp environment is a factor of 10 better than in any other environment—I get 10 times more work done per unit of effort than using any other set of tools”. No doubt, a programmer would be very productive if he was using a good Lisp environment that was familiar to him, but would he be better by a factor of 10 than in other environments? Besides, getting to know a Lisp environment well would take a substantial amount of time initially.

Having given a flavour of the style of the book, let us now consider its content. The main text is 306 pages thick with sixteen chapters. Chapter one gives a brief description of the motivation behind AI work, its derivative results and the promise that they hold. Chapter two contrasts the differences between human thinking. It ends with the assuring note that we do not want computers to compete with us, but want them to complement our skills so we can do our jobs better.

The next two chapters describe the results of AI research and applications in some details. Chapter three concentrates on different aspects of AI like natural language processing, robotics, vision, etc. Taylor’s assessment of the results is very down to earth. Chapter four describes thirteen successful applications mostly taken from engineering areas.

Chapters five and six introduce the programming language Lisp. Different aspects of a good software tool, using Lisp environment as an example, are described in Chapter seven. It emphasizes that a good software tool should support source code control, electronic mail and provide features that allow a user to debug, change, recompile and retest a program quickly. Chapter eight contrasts function-based programming, as in Lisp, with object-oriented and rule-based programming. This chapter gives particular attention to rule-based programming to prepare the reader for the next chapter on expert systems.

The chapter on expert systems begins with giving some reasons why people should consider building expert systems and then gives some hints on choosing problems for expert systems. Finally, some details of two well known expert systems, Mycin and R1, are described. The description is at a very high level and is not technical.

Chapters ten to thirteen are on logic programming, rule-based programming and Prolog. These are not there to teach one how to program but to explain the essential features and concepts of Prolog like unification and backtracking.

Chapter fourteen gives some practical help on how to get started in AI; it tells you where to buy your Lisp and Prolog systems and how to go about providing training for your staff. Chapter fifteen speculates about the future of AI and the influence that it may have on society. Chapter sixteen is about the Japanese—their fifth generation project and their culture. It is interesting but not entirely relevant.

In summary, *What every engineer should know about Artificial Intelligence* is a very useful book for anyone who wants to find out about the relevance of AI in practice.

Artificial intelligence and expert systems—case studies in the knowledge domain of archaeology* by J-C Gardin et al., Translated by Richard Ennals, Ellis Horwood Ltd, Chichester, 1988, pp 232, £29.95, ISBN: 0-7458-0431-4.

This is an interesting and potentially important book which is not, however, what the unwary purchaser might expect it to be. It is certainly not, for example, a general introduction to artificial intelligence and expert systems as its main title (in English) might suggest. Its extended title refers to case studies in the domain of archaeology and is more accurate though still somewhat misleading. In reality the book is: (a) a presentation of the method of “logicist analysis” for the humanities as

* Originally published in French under the title “Systems experts et sciences humains” 1987 Eyrolles, Paris.