knowledge? Superficial as this comment may seem, it is at the heart of a long-standing division between those who work in AI research and those in computer science, software engineering, and data processing. I recall arguments made at the time of the Lighthill report, when opponents of AI said it would be better to regard such systems as just another form of automation, rather than investing them with quasi-human properties such as intelligence and the like.

Unfortunately, parts of this report show some of the same tendencies. For example, right at the beginning in Chapter 1, we read that a knowledge-based system can "solve very difficult problems", "reason", "use judgemental rules", "contemplate" (*sic*), "explain", and "justify". This is the kind of language that has offended many people in the past. If, as proposed in the report, primary schools are encouraged to teach KBS, it is to be hoped they will not use this kind of language. To be fair, the report lists these properties of systems, but immediately qualifies the statement by saying these are a wishful, rather than an accurate, list of properties; however, the damage has been done.

The final part of the report contains sections on regulation and good practice. These give warnings about how KB systems should be controlled and presented to the users. If they are obeyed, many of my own fears would be reduced, but the nature of people working with AI suggests that, as in the past, they may be ignored in the rush to get new systems installed. Some of the recommendations in the final section seem too tentative, too vague and in some cases too impratical to be adopted.

Although the report lacks the cohesion and unity that would have come from a single author rather than a committee, it contains an extremely useful collection of information and advice and would form an excellent starting point for the discussions and debates that should take place when knowledge-based systems are being considered.

Reviewed by J. E. Nicholls, Programming Research Group, Computer Laboratory, Oxford University, 8–11 Keble Road, Oxford OX1 3QD

The reliability of expert systems by Erik Hollnagel (Ed.), Elllis Horwood, Chichester, 1989, pp 243, £25.00.

Readers of J. F. Sowa's classic textbook *Conceptual Structures* may have been surprised to find, prominently listed among the suggested further readings, the *Tao Te Ching.* Was this to say that students should start to learn Zen koans along with the calculus, and Buddhist logic along with numerical analysis? Such a programme has not, to my knowledge, been taken up; but the present volume indicates that the idea is really not so far-fetched. It is based on a one-day seminar on "safety and Risks in the Use of Expert Systems" held in Copenhagen in May 1988, with a heavy representation from the firm Computer Resources International; and one might say that the more senior the participants the more they dared to propound riddles and paradoxes.

Out of all these contributions, we can see that "knowledge engineering" has paradoxes packed in tightly enough to satisfy any Zen master. In many respects (and not simply to the mystified layman) computers embody logical precision. The physical events at their core must possess a reliability that is unique in its degree. And at the interface with human operators the syntax of communication permits no ambiguity or vagueness. Yet, as experienced practitioners know, all software requires craft skills in its writing and even more in its testing. With expert systems, the paradoxes are built into their very definition: "we want them to do perfectly things that we don't really understand" (p. 170). Those who reflect on how this task is to be accomplished must then decide on the best mixture of formal methods, communicable craft skills, and perhaps something else beyond.

To some extent, what we see here in "knowledge engineering" is common to any branch of engineering. The elements of "applied science" and craft skills are always in a complex dialectical interaction. The gap between teachers and theorists on the one side, and practitioners on the other is wide and fraught with misunderstandings. It may be seen as the conflict between two ways of knowing, the "knowing-that" which can be mastered by immature minds out of textbooks and lectures, and the "knowing-how" that is built up out of long personal experience and is only partly communicable. The one way can tend to be overly-general, abstract and unrealistic; the other can be excessively particular, unsystematic and perhaps even misinformed. Their fruitful union can never be a simple affair; indeed, it may be this that defines greatness in either a theorist or a practitioner.

It might be thought that "knowing-how" is a sort of subliterate skill, appropriate for carpenters and plumbers; but this book shows that even the most sophisticated, science-based technology, requires such a knowledge for its effective functioning. This need becomes even more urgent when the tasks are the most sophisticated and reflective, namely the reliability problem or quality-assurance in general. Considering the attempts made here to solve the reliability problem by formal means within the "knowing-that" tradition, we are led to an enhanced appreciation of the importance of the complementary approach.

The application of formal methods to the management of uncertainty is taken to its limits of effectiveness by D. Dubois, a recognized master in the field. He deploys all the concepts that generalize probability, including fuzzy-sets and "possibility". But at the end he states the limits beyond which such tools are not applicable; the unconquered area includes cases which require "fusing conflicting inferred facts, or to justify reinforcement effects". Since these tasks are all too common in the situations where expert systems are dependent on real data, the need for a complementary approach is implicitly conceded.

The other authors in the volume are closer to practice, and their solutions to the realiability problem are generally descriptive and taxonomical, including precepts based on their working experience of the complexities of real problems. It is the editor who lets himself go into the realm of paradox, as a means of instruction and illumination. For this he has a precedent in the words of welcome to the symposium by a distinguished colleague, who characterized the standard teaching languages hardly (e.g. "teaching BASIC should be considered a criminal offence"), and queried whether expert systems belong more to the problem set than to the solution set.

In his own survey of the background, he lists the ways that expert systems can *increase* the risks that their introduction is intended to reduce. This theme is amplified in the next contribution (by J. Robert Taylor). He discusses the limitations of expert systems applied to plant safety, with a reminder that a system with a few hundred rules is the largest manageable, and this may cover only one or a few components in a large plant! A later chapter discusses adequacy of models, an issue "which is still scarcely considered in expert system applications" (p. 157), in spite of the fact that an untested system may easily produce more harm than good.

Finally, the editor reviews the problem systematically, with some complementary principles, such as that these tools belong to *both* the problem set and the solution set; and that expert systems do best in the undramatic cases of "routinely knowledge intensive tasks that keep developing in the information society" rather than with the few striking cases. He dissects the problems of uncertainty, and also distinguishes between that and imprecision, as in the "unhappy marriage of symbols and numbers". And in the best master-craftsman fashion, he concludes his discussion with five types of failures, which can be understood as common pitfalls of work.

Speaking personally, I found it all quite illuminating in its way, but not at all surprising; but then my work over years has centred on the management of uncertainty in information, quality-assurance and craft skills and pitfalls (Funtowicz and Ravetz, 1990). The book is then of considerable philosophical and methodological interest; but if knowledge engineering has been able to get by for so long without such a budget of paradoxes, does it really need them now? As an outsider, I cannot comment on ordinary practice in the field of knowledge engineering. But as an observer of the scene in technology, I have been aware of the way that only a minority of knowledge engineers made a public criticism of the expert system that was so gargantuan in its size, and so grotesque in its specifications, that only a totally mystified public could believe the propaganda for it. I refer, of course, to "Star Wars". In that, fantasy and engineering were inextricably intertwined, in the context of an already corrupted American military procurement system and a rather special political situation. What could be more mind-blowing than such a multi-billion dollar hi-tech extravaganza? Those who over the decades had assured the public and themselves that "computers cannot err" might now reflect on the merits of an approach to technology that draws on paradox as the essential complement to linear logic. The present work makes a good start.

Reference

Funtowicz, SO and Ravetz, JR, 1990. *Uncertainty and Quality in Science for Policy*, Kluwer Academic, Dordrecht.

Reviewed by J. R. Ravetz, Council for Science & Society, 3/4 St Andrews Hill, London EC4V 5BY

Knowledge acquisition for expert systems (2nd Ed.) by A Hart, Kogan Page, London 1989, pp 196, £14.95. ISBN 1 85091 830 9.

Among the advances in the expert systems world since this book first appeared in 1986 has been a growing understanding of why it is that successful deployment of a working system is a rare achievement. We are now more likely to appreciate the complexity and subtlety of human expertise. Systems are more commonly labelled "knowledge-based systems"; not least because few programs can be said to perform at the level of an expert, who displays intuition and judgement on sub-optimal or inadequate evidence, and furthermore learns from experience. Clearly-defined boundaries, within which the knowledge base can be confidently assumed to be complete and correct, are one factor critical to success. No longer cloaked in mystery, KBS technology is increasingly viewed as another branch of information systems. "Integration" and "embedding" have been buzz-words in recent years, indicating that commercial or practical benefits are significantly improved by linking KBSs with other technologies (such as relational databases). This insight has been accompanied by recognition of the essential importance of validation and of maintenance. Another advance, as expert systems came into the fold, was increased attention given to identifying and involving the user in (and even throughout) the project. A certain level of relevant knowledge must always be assumed on the part of the user—so the earlier and more precisely we can define that knowledge, the better. Computers lack common sense. Human interpretation is usually necessary to some degree between the external world and the inputs to the system, and also between advice or decisions conveyed by the system and actions performed by the human user as a result. The context can never be completely defined for the computer.

One perception which has remained largely unchanged, on the other hand, is that knowledge acquisition is difficult: we still hear references made to the "Feigenbaum bottleneck". It certainly seems usually true that where the knowledge engineer is not already familiar with the task or domain to be modelled, the whole process of knowledge acquisition takes more time than implementation. Whether in fact it is also so difficult as to merit its image (among some) of a "black art", depends partly on the objective of the knowledge engineer (knowledgician?) and partly on the nature of the knowledge. If, at one extreme, the aim is to accurately and completely capture the private knowledge of the one person regarded as significantly knowledgeable about the problem, without access to any verified documents or other realiable sources, then knowledge acquisition may be a failure. On the other hand, some (successful) so-called expert systems have been designed as replacements for technical manuals: acquiring the knowledge from such a source may be tedious, but hardly difficult. Most KBS projects lie somewhere between these two paradigms, where the knowledge engineer following "guided curiosity", with an eye (and an ear) for detail and a degree of common sense, should succeed.

Knowledge acquisition must, however, be disciplined and structured: these days, at any rate, there will usually be a fixed budget and a project plan, incorporating deadlines. There is a considerable gap to be bridged between the mass of disparate sources drawn upon by people in solving a problem, and writing a computer program which generates (or advises on) a solution. Often, too, those who are expected to use the KBS have significantly less relevant knowledge and experience than the expert who acts as the primary source for the knowledge base. Ensuring that the system's user interface is comprehensible for the ultimate user is a part of the knowledge engineer's job, often