The social impact of loss of industrial competitiveness has been huge, as continuing unemployment testifies, and failure to capitalise on the commercial advantages offered by KBSs will only compound Britain's industrial problems. It is important that Britain's industrial managers are made aware of the significance to their businesses of KBSs. The report does not address this issue.

There may be more subtle social consequences of KBS implementation in industry. Will KBSs bring about a change in the power hierarchy of the workplace, as workers are provided with decision support systems to enable them to make decisions once referred to a manager? Will management find many of their responsibilities for planning and budgeting taking over by planning systems? Will the corporate mainframe pass its processed sales figures to an expert system for analysis, rather than the present legions of accountants? How will industry cope with these changes?

Excepting the lack of material of direct industrial relevance, this is a valuable report which will interest anyone who wishes to be acquainted with the forces which will shape their future. It should be compulsory reading for all those who have a responsibility for controlling those forces, especially those in politics and the military, whose ignorance of the consequences of their actions could be catastrophic.

Reviewed by J. A. G. Hawkins, Johnson Matthew Technology Centre, Blount's Court, Sonning Common, Reading RG4 9NH

Benefits and risks of knowledge-based systems Report by a Working Party, UK Council for Science and Society, OUP, Oxford 1989, pp 76, £6.95.

The aim of this report, sponsored by the Council for Science and Society, is to alert the public to the consequences of working with knowledge-based systems. The council was right to select this as a topic for study, since much of the work in this field is likely to affect the way in which society will react to advances in science and technology.

Because of the high incidence of errors in computer systems, we now all know that both hardware and software can give incorrect or misleading answers. It is obvious that people will be concerned when they realise they have been under the influence of mechanisms they do not understand and can not control.

When a *knowledge-based system* goes wrong, the effects are likely to be much more serious than for conventional applications. One reason is that knowledge-based systems are deliberately designed to appear natural and human—this makes their output more plausible than conventional programs, and hence more disturbing when they are in error. Another reason has to do with our still imperfect ideas of what is correct (and incorrect) about such programs.

Anyone who has written a simple program in Basic or Pascal knows that computer arithmetic can result in errors, that mistakes in addressing can result in the wrong variable of an array being accessed, and so on. It is worth considering what kinds of error can appear in a knowledge-based system. They include the same kinds of error that appear in conventional programs, but in addition there are the more subtle and pervasive errors that come from the contributions of an "expert". These may be errors of judgement, and perhaps encoding, and are much more subtle than the errors we have become used to in computer systems. Some systems, when they give results, allow the user to see the chain of reasoning used in the system. Few users will be able to evaluate this kind of output, and it will serve merely to re-enforce the apparent correctness of the system, rather than its fallibility.

The report starts with a short and necessarily sketchy account of the working of knowledge-based systems. This is followed by a survey of five application areas: finance, law, education, military and health. This is a potentially valuable part of the book, since it covers the selection of applications, one of the most critical parts of building a system. The accounts of what is being attempted, and in particular how these applications can fail, will be alarming to those unaware of directions in this field.

AI people have always seemed to try and separate themselves from the rest of the computing community. This can be seen from the very terms they use: Artificial Intelligence, Expert Systems, Knowledge-based, all seem to be making a point. Who could be *against* intelligence, experts, 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