

Models in expert systems

Brian Smith

Xerox PARC

All expert systems are based on models. The 'knowledge' embodied in expert systems, in particular, is usually encoded in a set of 'rules' that describe the problem and specify the behaviour that the system should manifest. These rules are always formulated with respect to a model of the underlying domain. This model must be determined in advance by the programmer, who may in turn have derived it from an analysis of the experts' performance on which the system is based.

Indeed, one of the prime tasks in building an expert system is to develop an appropriate model. There are various ways to do this: by analysing the desired behaviour, by building on underlying scientific theories, or by codifying the models apparently used by expert human practitioners. What phenomena are dealt with, what phenomena are ignored, and what patterns or regularities connect the phenomena that are dealt with -- all these decisions are made at the level of the model.

For example, a medical expert system designed to administer drugs might model drug absorption in terms of a scalar quantity proportional to the square of a patient's height, or proportional to the weight (neither model, of course, would be expected to be entirely accurate). An expert system for the office might model a secretary as a customer, producer, and processor of information, with a complex internal state. A defense warning system might model incoming missiles as point masses on parabolic flight paths, and model the atmosphere as a linear retarding force. And so on and so forth: the use of models permeates formal systems of all sorts.

When expert systems are actually deployed, however, they interact with the world itself, not with models. For example, when drugs are actually administered, or when offices are actually equipped with expert systems intended to work alongside people, we have full, thick situations to deal with, of at least potentially arbitrary complexity. Furthermore, the success of expert systems ultimately depends on their ability to deal with these rich, embedded situations. Their success, in other words, isn't exhausted by their ability to deal appropriately with the model used in their construction, or encoded in their knowledge bases.

In fact the only ultimate point of the models in expert systems is to help them succeed in the 'real world'. RCA, for example, is primarily interested in whether their satellites will actually get into orbit and stay there; they have only a derivative interest in whether the programs guiding them are proved correct with respect to a particular orbital model.

It is clear, therefore, that in order to analyse expert systems we need to understand the appropriateness of the models on which they are based. Analysing expert systems, in other words, comes in two parts: understanding the behaviour of a system in terms of the model on which it is based, and understanding the relationship between that model and the embedding world. At the present state of the art, we have a variety of techniques that enable us to study the former relationship, between system and model: formal semantics, model theory (hence its name), program verification. We have virtually no techniques, on the other hand, with which to study the latter relationship, between model and world. We are largely unable, therefore, to assess the appropriateness of models, or to predict when models will fail. All that we do when we prove a program 'correct' is to prove that it will behave as specified with respect to a model. It would be something quite else -- something we don't know how to do -- to prove that a system will in fact do the 'correct' thing once embedded into a real situation.

Two conclusions. First, we should develop and use expert systems only in those domains where we have confidence in the accuracy and appropriateness of our models. Second, we should develop a 'theory of models' with which to understand better how models work, and how they make sense of the infinite complexities of the worlds they represent. Although such a theory may strain at the edges of what can be formalized, or even pass beyond strict formal limits, we can still develop rigorous tools, and use clear-headed thinking, in analysing this important relationship.

The trivialization of expertise

Terry Winograd

Stanford University

Professor Moto-Oka of the Japanese Fifth Generation project [3] predicts that: "*Fifth generation computers are expected to function extremely effectively in all fields of society. ...totally new applied fields will be developed, social productivity will be increased, and distortions in values will be eliminated.*" Feigenbaum and McCorduck [2] proclaim that: "*We are now at the dawn of a new computer revolution... [leading] to computers that reason and inform. ...the engine that will produce the new wealth of nations. ... Perhaps equally important to all the economic advantages the Fifth Generation promises is that intangible thing called quality of life. A society where knowledge is quickly and easily available to anybody who wants it will... be an alluring place.*"

We might dismiss these statements as merely naive self-serving propaganda, but in doing so would we fail to

recognize the background in which they could be made sincerely and taken seriously by intelligent and educated readers. The computer is a powerful embodiment of a 'rationalistic' tradition that equates certain limited modes of rational description and inference with the full scope of how people think and what they do with language. This tradition is both demonstrated and further promoted by inflated claims for the potential benefits of 'expert systems.'

Rationalistic understanding has been extremely powerful in the creation and expansion of the physical sciences and in increasing our mastery over the physical world. At the same time, its power in dealing with these domains of reality has blinded us to its weakness in dealing with those more related to human life and society. The rationalistic orientation often promotes a wrong understanding of what constitutes a 'problem' in a human domain, and what computers can do to 'solve problems.' The failure of the rationalistic approach has led to a crisis in the practical areas where it has been applied seriously. The recognition of this crisis has emerged in the past few years in works on management theory, military systems, medicine, energy policy, and many other fields.

The problem is illustrated by the illusion promoted by the label 'expert system.' A human 'expert' is someone whose depth of understanding serves not only to solve specific well-formulated problems, but also to put them into a larger context. We distinguish between experts and idiot savants. What, in contrast, do computer systems do?

Expert systems are built on the basis of a relatively small, precisely defined set of object types and properties, together with a (possibly large and disorganized) collection of rules relating them. In pre-selecting the relevant elements out of which to build rules, one must cut out the role of context and background (which are at the heart of AI's theoretical difficulties). This process by its very nature creates blindness -- a limit is set by the way the world has been articulated. There is always the potential for breakdowns that call for moving beyond this limit -- for returning to the context and reformulating the problem.

One might argue justifiably that this blindness is a problem for people as well. But (with the exception of idiot savants), people are not programmed for a particular task. A normally intelligent person can always step back and recontextualize. Expert systems, even those with 'meta-rules' do not have this openness. Further, the blindness inherent in representation leads to difficulty in understanding the range and limitations of a particular program. A human expert can enter into a dialog about his or her own range of knowledge and limitations, moving outside the putative domain and using ordinary language and ordinary common sense. Failures of AI programs will in part reflect the inability of people to understand what

the program is actually doing, as opposed to what it might appear to be doing if the metaphor of 'thinking' is accepted.

Although the assumptions of the rationalistic orientation may seem self-evident (to those within our modern Western society), they are indeed only assumptions and can be challenged (see [4] for a more comprehensive discussion of this and the other issues raised in this paper). Instead of treating 'data' as the objective representation of reality, we can recognize symbols (on paper, or in a computer) as a medium for language, which is based on human commitment and is always relative to an unarticulated shared background. Instead of trying to get the machine to have 'understanding' through a collection of 'rules,' we can recognize that its manipulation of symbols is grounded in the background and interpretation of people who interact through it. We can identify and articulate 'systematic domains' of symbolic manipulation for which appropriate rules can be generated, and we can integrate these into a broader appreciation of human knowledge and expertise.

References

- [1] Dreyfus, Hubert L., and Stuart E. Dreyfus, *Mind Over Machine*, New York: MacMillan/The Free Press, 1985.
- [2] Feigenbaum, Edward, and Pamela McCorduck, *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World*, Reading, MA: Addison Wesley, 1983.
- [3] Moto-oka, T., Keynote speech: Challenge for knowledge information processing systems, in Moto-oka, T. (Ed.), *Fifth Generation Computer Systems: Proceedings of International Conference on Fifth Generation Computer Systems*, Amsterdam: North Holland, 1982.
- [4] Winograd, Terry, and Fernando Flores, *Understanding Computers and Cognition: A New Foundation for Design*, Norwood, NJ: Ablex, 1985.