

Selected Bibliography

- Brakto, I, 1986. *Prolog Programming for Artificial Intelligence*, Wokingham: Addison Wesley
 Clocksin, W and Mellish, C, 1984. *Programming in Prolog*, Second Edition, New York: Springer-Verlag
 Gallier, JH, 1987. *Logic for Computer Science: Foundations of Automatic Theorem Proving*, Singapore: John Wiley

Database applications using Prolog by Robert Lucas, Ellis Horwood Ltd, 1988, pp 159, £19.95.

The Alvey programme over the last few years has, to some extent, been an attempt to spread the new "fifth generation" methods out of the ivory towers and into industry and commerce. In many ways this has been successful, and numerous collaborations between academic research groups and commercial and public sector organizations have been established. Nevertheless there is still a widespread view that as yet the new technology has not been fully transferred. In particular, in the field of knowledge-based systems, and specifically Prolog-based ones, there seems to be a lack of significant "real" applications. This situation makes recent government cutbacks in funding for such collaborations difficult to counter, reinforces a tendency for the research community to be inward looking and encourages industrial partners to perceive that community as impractical symbol pushers. One aspect of *Database applications using Prolog* by Robert Lucas, the product of collaboration between the Department of Computer Science at Coventry Polytechnic and local industry, is that it makes a welcome contribution to the process of technology transfer. It has to be said, however, that the book is concerned more with the first stage of transfer, producing prototype systems, than the final stage of fully operational industrial systems.

We are told (p 12) that "the whole thrust of the work described was based on the observation that we can use a database as a repository for facts for the Prolog system". An important aim of the work was, therefore, to provide an efficient interface between a Prolog front-end and a conventional database. As well as a thorough discussion of the construction of this interface and three prototype applications developed using the system, a good part of this short book (about 40 pp) is devoted to reviewing the elements of Prolog, relational databases and expert systems. While this is moderately successful it suffers from attempting to cover in a very short space material which is accessible elsewhere. On balance, however, the completeness gained is probably worthwhile, particularly because it encourages the use of Prolog from a logical point of view.

The major focus of the book is the design and implementation of a tightly coupled interface between Prolog and a commercially available relational database (the Mimer database). The detailed account of the problems which arise in the construction of the interface provide a useful source of information for others involved in similar projects, even though the implementation, in Pascal, is specific to the Mimer database. The final implementation is tested for performance. This is clearly of crucial importance because an underlying assumption of the composite Prolog plus database design is that it can achieve efficiencies better than a pure Prolog system, and also that these efficiencies are acceptable in comparison to other database access methods. The tests carried out, although rather limited, show tentative support for the design. The real value of this part of the account, however, is methodological, some tests are much better than no tests at all.

The strength of this book is in the example applications. The most entertaining of these, devised in collaboration with the police, is concerned with the detection of burglaries. Apparently this activity is bogged down with paper work; scene of the crime reports are made on paper, background data is supplied on paper and so on. One aspect of the project was to provide a flexible database to replace this paper work. The other was to model a particular aspect of the detection task carried out by the burglary squad: the value of information on the modus operandi of burglars, for example whether they prefer upstairs windows or are deterred by dogs. We are suitably appalled by the fact that "only 12% of burglaries are investigated by the squad", in the remaining cases the police simply wait for the perpetrators to be picked up for something else, or not as the case may be. The completed system was set up in a police station for evaluation, but the burglary squad "showed little interest in it" even

though it was set up within twenty yards of them. Perhaps most depressing for the designers was that “no actual arrests were made . . . so at the end of the trial it was not possible to say conclusively that the system does detect the correct burglars”, nor, I would add, that it does not detect too many incorrect ones. In spite of these obstacles, which are instructive in themselves, the book gives an interesting and convincing account of the value of this prototype. The lessons learned are applicable to other areas, such as clinical diagnosis which has to deal with case histories and many sources of data within the hospital administration. Indeed this project addresses a huge potential market, the application of expert systems to the problem of administrative paperwork in any large organization.

The book also describes a fault-finding diagnostic system for a substantial network maintained by ISTE Ltd. I have to say that I did not get so much enlightenment from the description of this project, but the area of expert systems applied to the maintenance of large scale information systems clearly targets another market with enormous potential. The last system concerns a motor car assembly line. Apparently the robots in control of paint spraying sometimes spray the oncoming cars with the wrong colour and this has to be ‘rectified’ at extra cost. The point of this application was to show that the system could be used in a real time context. The final section of the book is concerned with a restricted form of natural language interface for rule acquisition. After a review of some aspects of natural language processing, an interface is described which is menu-driven but allows the expert to respond in natural language.

In summary I would recommend the book to the expert system practitioner and to students specializing in this field. It has the important merits of encouraging a clear logical way of using Prolog, a very systematic way of analyzing problem specifications in three important application areas, and it collects under one cover the necessary background material. There are also many points of interest for the researcher, especially in the treatment of the applications and the rule acquisition interface.

The rise of the expert company (1) by Edward Feigenbaum, Pamela McCorduck and H. Penny Nii, Macmillan 1988, pp. 322, £14.95.

It is difficult to come away from reading this book with other than a feeling of optimism and satisfaction. Here, it would seem, are a set of ready answers to those questions that the unconverted frequently pose—“After XCON and Dipmeter Advisor, then what? Where are the installed expert systems that really offer a payback?”. Well, take your pick from dozens of examples in this book—from steel furnaces to disk drives, from the early estimation of building costs to the evaluation of medical care in the NHS. The range is deliberately chosen to be wide and impressive, the claims for the savings in human time are often startling.

And yet it is also difficult to come away without a feeling of unease and frustration. Wouldn't it all be even more impressive if some failures were reported in the book too, and their causes discussed a little? What about all the prototypes that don't scale up into money-spinners? Whose fingers got burned in tackling the technology unwisely, and what can we learn from their mistakes? There is nothing of that to be found within these pages, and it results in a distinctly unbalanced tale.

Prof. Feigenbaum was recently quoted in an interview in a British computing newspaper as saying that the authors wanted to include some negative stories, but could find none. I doubt whether this means that failures cannot be found when one digs for them; more likely, it reflects a natural preference within companies for positive image making in a book like this. The authors clearly derived much of their material from interviews with the successful knowledge engineers, experts and technology champions in these companies. This is an entirely natural and reasonable thing to do, but it is as well to approach it with a healthy curiosity about the casualties of the learning curve, too. I don't regard it, for instance, as sullyng the reputation of my own employer, some of whose successful internal applications are well covered in the early part of the book, to comment that not every prototype started within the company has been successfully scaled up or has found a profitable niche.