presents an overview of other projects that have attempted to apply AI to teaching a second language.

Multilingual multimedia should provide interesting reading for students and practitioners of computer-assisted instruction and artificial intelligence. The authors present highly readable and well illustrated material. No other book in the market is as focused on the same set of approaches to intelligent computer assisted language learning.

Reviewed by Claude Chaoui and Professor Roy Rada, Computer Science Department, University of Liverpool, UK

Eco-logic: logic-based approaches to ecological modelling by David Roberston, Alan Bundy, Robert Meutzelfeldt, Mandy Haggith and Michael Uschold, MIT Press, Cambridge, MA, 1991, £31.50. ISBN 0-262-18143-6.

This addition to the MIT series on Logic Programming stands out as being, as far as I am aware, the only one dealing primarily with the application of logic programming to a major real world problem. It essentially summarizes the results of a programme of work undertaken by the authors as part of the Alvey Programme, at the Department of Artificial Intelligence, Edinburgh. The top level goal of their project was "to make simulation modelling accessible to people with no previous knowledge of programming." As a focus for the project, they chose the target domain of ecology. And as a vehicle to achieve their goal, they chose logic programming. Hence the name of the book, which is also that of the project.

Simulation modelling is currently very much in the public eye, especially in the context of ecological simulation. Important decisions need to be taken, for example, on the basis of models of expected rates and effects of acid rain precipitation, of global warming due to increasing emissions of greenhouse gases, and so on. The trouble is, there are a number of competing models. These are often encoded using mathematical/statistical modelling techniques, and/or as computer programs. The problem then arises of communicating the assumptions upon which these models are based, and enabling them to be critiqued by non-experts of simulation modelling. It is vitally important to be able to do this, as these models are being used to support arguments for courses of actions which are warranted by the prediction of the model. For example, that we should reduce CO_2 emissions because the current rate of increase of CO_2 emission will (according to certain models) result in an increase in mean global temperature, which will (according to certain other models) affect the growth patterns of terrestrial vegetation in such a way that (according to yet other models) the future survivability of the human race will be seriously compromised. Unless the grounds of all these arguments are made clear to those who are being asked to make decisions, there will be no way for the decision maker to make an informed judgement.

Logic programming to the rescue! If the simulation model can be encoded as a set of logical statements, then Prolog may be used both to specify the model, and to run an animation of the model to generate predictions. If a clean logic programming style is used then the assumptions upon which models built in this way are based should be obtainable by inspection. The Eco-Logic group found they had one or two problems with maintaining a pure declarative style, however. This was mainly because they were finding it hard to express the things they wanted to without introducing rather a lot of "cuts" into their Prolog statements. As a consequence, the declarative meaning of their programs was being contaminated with a strong procedural element. (For non-Prolog people, a "cut" is an extra-logical symbol which may be introduced into a Prolog clause. It can be read as an instruction to the Prolog interpreter to commit in the search for solutions to a goal at the point at which the "cut" occurs, and cut out any alternative paths in the solution space.)

There were perhaps two main reasons for this apparent need to control the search strategy of the Prolog interpreter. Firstly, Prolog is a relational language. For example, a predicate such as `father(tony, paul)` expresses an instance of the father relationship. A query to the Prolog interpreter, such as |?–`father(tony, Son)` (`Son` is a variable—"what Sons is tony a father of?") will succeed with a solution `Son = paul`. But there may be other solutions, and Prolog will endeavour to search for them. However, we may wish to write a general predicate such as `mass (Population, Time, Mass)` which we know should only have a single solution for a given query; at any one time, a population of animals can only have one mass. In this case, the predicate represents a function, and not a relation. This is a different *type* of thing, and we do not want it to be interpreted in the same way. In particular, we know any query involving a function can only have a single solution, and we do not want the Prolog interpreter to waste time searching for additional solutions (there are one or two other considerations for wanting to distinguish a function from a relation).

A second enhancement is that we really want to recognize that specific functions and relations are only defined for entities of a specific type. Again, this is basically a matter of wanting to stop the Prolog interpreter from casting its net unreasonably widely in its search for solutions. For example, we would not want it to look to see if there were any vegetables or minerals which satisfied the father relationship.

The approach which the Eco-Logic group took to solving these problems was to use order-sorted logic as their basic representation language. This enables the definition of functions and relations as distinct types, with their associated properties (namely that a function must be single valued). It also enables a structure to be placed on the universe of discourse so that relations or functions may be defined as being between specific classes of entities, with the classes of entities being ordered into subclass hierarchies (for example, that "zebra" is a subclass of "mammal" is a subclass of "animal" is a subclass of "organism" is . . .). This is a "conservative extension" of logic, as it does not actually add to its expressiveness; anything that can be written in order-sorted logic can also be written in classical logic. It merely enables statements to be expressed in a more succinct style.

There is still a difficulty. This is that few ecological modellers will be expert at writing, or interpreting Prolog programs at all, let alone have the expertise to write clean declarative programs (either in Prolog or in the order-sorted logic variant). The full EL system, however, provides some assistance to help users construct simulation programs. There is, firstly, a problem description subsystem which helps the user describe the problem which is to be modelled. Then, secondly, there is a program generator which can offer a range of high level program building blocks, or schemata, from which a complete simulation model can be built. The problem description is used to guide the selection of schemata which are proposed to the user as possible building blocks.

The above is a very incomplete outline of the total package which was developed, and which is described in this book, although it should at least give a flavour of the content. The main reason for going into this at some length is to emphasize that there is much material in this book which is of interest to a wide range of audiences. Certainly it will be of interest to ecological modellers, or anyone else who is into building simulation models. But it also contains a great deal of interest for those computing scientists interested in logic programming, automated program construction, artificial intelligence, automated inference and software engineering. The book has been carefully structured to make it as widely accessible as possible, with a short summary at the beginning of each chapter giving a succinct description of the main results of that chapter. Technical descriptions of algorithms are isolated from the main body of the text and may be skipped without loss of continuity. This format was specifically chosen to make the book accessible to a wide range of audiences, whilst still giving sufficient "meat" to satisfy the logic programming literati. However, it has the side-effect of being a very effective method of communication for the latter as well.

I must express some slight reservations about the ultimate extent of the applicability of the final system. The main examples given in the text are of small scale models; rather simple systems involving rabbits, sheep, grass, and so on. These obviously need to be very simple to suit the tutorial nature of most of the text. But how easy it will be to apply the Eco-Logic approach to large, complex, ill-defined problems? There may well be a class of problems for which modellers in environmental consultancies, or biology and ecology researchers could use Eco-Logic. But the sorts of problems with which I introduced this review are orders of magnitude more complex than the tutorial examples. They needs must be drawn into the mainstream of scientific research and review, with the assumptions and methods of solution presented for peer review in the appropriate scientific domains. There is a fairly smooth progression of complexity (and real world applicability) from models of rabbits breeding in geometric ratios to global models of climate change. Exactly where on this scale lies the limit of applicability of Eco-Logic remains to be seen. Nevertheless, the motives behind the project are laudable, and maybe advances in logic programming technology will push the frontiers of applicability further.

In summary, a very nice book with much to commend it.

Reviewed by Paul J. Krause, Advanced Computation Laboratory, Imperial Cancer Research Fund, 61 Lincoln's Inn Fields, London WC2A 3PX, UK

From natural language processing to logic for expert systems edited by André Thayse, John Wiley & Sons, Chichester, 1991, pp 535, £29.95, ISBN 0 471 92431 8.

Two of the central issues in artificial intelligence research are knowledge representation and inference. Among the many formalisms and inference mechanisms researched and used, probably the first used formalism still occupies the central platform. That formalism is logic. The advantage of using logic is its clean semantics, well developed methodology and techniques of making inferences. Effective use of logic, as with any other discipline, requires considerable thought. Practical examples which bring together the theory and practice would be very beneficial for beginners as well as experts. The successful application of logic in different fields is the main theme for the trilogy *A Logic Based Approach to Artificial Intelligence*. The first two volumes are devoted to basic logic, concepts and various AI techniques. The volume under review is the last of the series, which emphasizes the use of the various logic languages and formal grammars described in the previous volumes for the most significant application areas of AI and computer science. The book consists of eight self-contained chapters, organized according to the general theme, as the editor states:

spoken natural language → written natural language → logic languages → application-specific language → programming language

The first chapter starts with speech recognition. It first discusses the general problems involved, and then moves on to hidden Markov models and the related problems in acoustic decoding. The major point of the chapter is how to use a language model based on a recursive transition network of a context-free grammar to improve the recognition algorithm. The strength of the proposed language model results from incorporating syntactic knowledge into the algorithm. Surprisingly, the very first chapter is not really about using logical formalism.

Chapter 2 continues the main theme. The first part overviews the general issues in natural language understanding and develops a logical language in an intermediate logic form. It is instructive to see that various results and tools of mathematical logic can be used to increase the competence of natural language processing systems. The rest of the chapter explains how to translate the form described in the first part into Prolog form, a chosen target application language for a particular realization of the proposed formalism.

Chapter 3 is on building knowledge bases for expert systems. It starts by illustrating the advantage of a logic-based approach for the construction of problem solvers for expert systems. Then the discussion focuses on a particular formalism and the different aspects in using it to represent various parts of a knowledge base. Much effort is devoted to plausible reasoning by an inheritance hierarchy in an object-oriented manner.

Chapter 4 introduces two classical realizations of truth maintenance systems. The two paradigms of Doyle's TMS and de Kleer's ATMS are explained superbly in a clear and concise way. This chapter also includes the discussions about why these two techniques can make knowledge based systems more efficient.