The Knowledge Engineering Review, Vol. 9:1, 1994, 83–94

Book reviews

Multilingual multimedia: bridging the language barrier with intelligent systems, by Masoud Yazdani (Ed.), Intellect Books, UK, 1993, pp 210, £14.95, ISBN 1-871516-30-7.

At first sight of the title of the book, and as a multilingual person myself who does research in the field of hypermedia, it seemed to be a very attractive book to read. Networking and electronic mail systems facilitate communication between people throughout the world. Nowadays, speaking two or three different languages is invaluable in Europe and worldwide. Multimedia and hypermedia modernize different aspects of teaching and learning. It is amazing what people do, when they do not speak the same language, in order to make communication possible. Every one of us has experienced such things when we travel in the world. On some occasions, people use body language, on others they use drawings or icons to signal a meaning. An interesting aspect that this book addresses is language learning and interactive communication through icons, an approach that suits people who do not speak one another's language. Another aspect that this book deals with is extending systems to cope with multi-languages.

It is great to have more choices. Personally, I would like to be able to communicate in my own language as well. Being able to communicating in one language and receive responses in a different language will be great fun (just imagine!).

Although the title of the book uses the term 'multimedia', some of the systems described are actually hypermedia systems. So, to make things easier for readers who are new in this field, it is worthwhile defining briefly the terms *hypertext*, *hypermedia* and *multimedia*. Hypertext systems refer to systems that allow nonlinear accessing and browsing of information. The information in this case is mainly textual. Hypermedia systems have the same features as hypertext, but in addition to textual information, it handles video, audio and animation. Multimedia systems handle different types of information such as text, audio, video and animation. In multimedia systems, the storage and access of information is not necessarily nonlinear.

This book is a collection of nine papers, referred to as chapters in the book. Some of these papers report on work related to a European DELTA project which developed multilingual systems for education. Each chapter is between 10–15 pages long on average, and is followed by a list of references. A small number of references are provided, and most are several years old. (This is not a criticism, as multilingual multimedia is still a new topic.)

The topics covered by the book can be divided into two main categories: systems and prototypes; and survey of development approaches and design issues. Chapter 1 describes *Restaurant*, a prototype system developed in HyperCard, a hypermedia system, for practising foreign languages in a restaurant situation. It incorporates text, animated graphics and digitized speech. Chapter 2 presents a survey of approaches to the development of multimedia software and relevant multilingual issues. It describes multilinguility in the context of multi-culture, which is interesting to consider: How people, who do not speak the same language and who belong to different cultures, actually communicate? Chapter 3 presents design consideration for a visual language and how it could be developed to such a level as to enable presentation and understanding without much training. Chapter 4 presents a prototype hotel-booking system developed in HyperCard which uses icons to allow a potential guest and hotel manager to communicate. Chapter 5 describes a system for the teaching of Modern Languages that incorporates 'human-like' knowledge of the teaching domain. Here a system called 'LINGER' (a language-independent bug-finder) is described that was extended to cope with the Spanish language. Chapter 6 presents a critical evaluation of LINGER performed at Purdue University in the USA. Chapter 7 describes enhanced LINGER which uses some AI techniques for syntactic error analysis and corrections. Chapter 8 assesses the performance and architectures of a number of grammar checkers. Finally, Chapter 9

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