People working in related research can find ideas in the book about complementary approaches, or even hints how to solve their own problems. But it is unlikely that they will find detailed solutions, because approaches described are often rather application-specific and specialized for particular software and/or hardware. Also, research in this field is in an early stage, and the techniques are often presented and tested using only toy-examples.

A substantial part of the book deals with problems of qualitative simulation and the use of "landmarks" in qualitative representations. The intrinsic ambiguity of qualitative methods can be reduced by introducing new landmarks, constraints based on confluences, or semi-quantitative ranges. At least a whole chapter is given to each of these three approaches.

Promising topics in simulation modelling presented in the book comprise inference of sensitivity, and different levels of abstraction. The straightforward approach to sensitivity analysis requires running a simulation many times, perturbing individual parameters and evaluating the differences of the results. This is extremely expensive, and as a result is seldom properly carried out. The alternative approach is based on representation of sensitivities of all functions in a model and then the desired global sensitivity can be assessed by propagating the individual sensitivities through the computation.

Another technique which will be important for future systems is learning. Reinforcement learning, for example, which can be defined as the process by which the response probability of a system to a stimulus is strengthened by a reward or weakened by a penalty, has been applied in a hierarchical framework for process control. It can be successful even if the process dynamics and process disturbances are unknown.

Two chapters deal with parallelism which may be very important for complex simulations, but currently techniques for effective parallelising simulations are limited as it is extremely hard to achieve substantial speed-ups because, for example, of the heterogeneity and asynchronicity of many complex systems.

This is the first comprehensive book on this subject. It presents contributions of leading researchers, and shows the development of the field which will have definitively increasing importance in the future. It can be recommended to researchers in both fields, AI and simulation, as a good background and as a source of ideas for problems which may arise in this combined field, and it may facilitate attempts to search for a solution.

The electronic design studio by M McCullough, W J Mitchell and P Purcell (eds.), MIT Press, Cambridge, MA, pp 493, £44.95.

On coming to this book, I knew little more about the use of CAD tools for architectural design (CAAD) than could be gleaned from the glossy pictures of 3D graphics workstations in the CAD/CAM press. But the book itself provides all that is needed by way of introduction. CAAD tools have apparently been in existence for almost 20 years. During this time they have evolved in a number of ways in attempts to support more of the creative design process, and are now beginning to make use of a number of knowledge-based system techniques.

This review will attempt to provide a brief overview of the field of CAAD and the characteristic problems in this field which are being tackled by the introduction of KBS techniques. The book is divided into four sections: an introduction to the theoretical foundations of CAAD, a section on "Information Delivery Systems for Design", a collection of case studies on the use of CAAD in real design situations, and a set of more theoretical papers on the use of KBS techniques. The approach described means concentrating on the first and last of these sections (which constitute together about half the book), though other sections contain material which may be of more interest to teachers and practitioners of architectural design.

Architectural design is defined as "the computation of shape information that is needed to guide fabrication or construction of an artifact". This computation involves recursively transforming and combining shapes according to rules defined by an appropriate "shape algebra" until it can be shown that a set of pre-defined constraints on the final design have been satisfied. However, even those propounding this essentially computational view of design must admit the existence of at least three characteristics of creative design, as practised by architects using pencil and paper, which cause problems for computer-based support tools. Firstly, shapes, or combinations of shapes, are ambiguous in the sense that they may be open to interpretation in a number of ways. This idea of "emergent shapes", or shapes which somehow emerge unexpectedly as a result of combining other components in a design in a particular way, is one which seems very important in the business of architectural design. Secondly, the rules of a shape algebra are often unstable or context-dependent. Thirdly, critical reasoning about architectural designs is essentially non-monotonic, in that judgements about whether some aspect of a design has been satisfactorily completed may later be reversed in the light of further considerations. This last factor makes it hard to decide when a design is finished.

Early tools for CAAD emphasized rigid structures for designs by specifying shapes in terms of points, edges and surfaces. This was at first felt to be an advantage, but later came to be recognized as unnecessarily restrictive. The trend in more recent years has been to attempt to provide semantically richer and more flexible descriptions of both the initial shape vocabularies and the designer's knowledge about how elements of these vocabularies can be transformed or combined. This trend is exemplified in a number of papers in the book. For example Stiny and Tan suggest more structured and flexible ways of describing shapes in terms of schemata and high-level relationships between their components, and Schmitt et al. suggest attempting to code a number of different types of expert designer knowledge into a system for CAAD. Rules, frames and scripts all make an appearance, as do Lisp and Prolog in roughly equal measures. Approaches based on the use of neural nets and hypermedia are also discussed in two papers.

These later tools have begun to tackle the problems of ambiguity or discontinuity in shape interpretation and the need for flexibility in rules about design to provide more useful support to the expert architectural designer for a wider range of activities than was previously possible. There is still, however, much progress to be made, and in particular, the problem of non-monotonicity has apparently not been cracked.

The book as a whole is well-produced and has a good index, though one or two of the papers seem to have suffered a little in translation (contributions from Europe, Israel and Japan appear alongside those from Canada, Australia and the US). Its subject matter is at first sight rather esoteric (and this is reflected in its price!), but on slightly closer inspection, yields an interesting insight into the applicability of KBS techniques in an ususual but important area of application.

Approximate reasoning models by Ramon Lopez de Mántaras, Ellis Horwood, Chichester, 1990, pp 109.
Search, inference and dependencies in artificial intelligence by Murray Shanahan and Richard Southwick, Ellis Horwood, Chichester, 1989, pp 140.

A distinguishing feature of intellectual man* is his ability to solve a diverse range of problems. However, in modern professional life the need to solve problems of such complexity, or requiring such wide domain knowledge, or with such frequency, is making computer-based assistance an increasingly necessary tool. An expert in a given field will have access to relevant domain knowledge and theories to enable him to identify solutions to a given problem, and make a judgement on the efficacy of acting on those solutions. He will, in general, be able to reason with uncertain and incomplete information, hypothesise and revise solutions as further information becomes available or if the solutions are found to violate constraints imposed on the solution space.

The difficulty is to find a model of human reasoning which is sufficiently rich to capture all these aspects. To date, in AI there has been a long tradition of using numerical techniques to cover uncertainty and vagueness in reasoning. More recently, there has been an increasing use of Reason Maintenance Systems (RMSs) to model a more symbolic style of problem solving and hypothetical

* Out of laziness, and in want of a better word, I use the male term to represent the generic of personkind.