AI Magazine Volume 12 Number 1 (1991) (© AAAI)

Book Reviews

Artificial Intelligence, Simulation, and Modeling

Mark E. Lacy

As a system scientist doing modeling and simulation, I have been interested for some time in ways that modeling and simulation and AI could be of value to each other. After all, both areas have their roots in putting knowledge into useful representations. I have speculated (*AI Magazine*, summer 1989, pp. 43–48) that the scientist of the future, in applying computing to his(her) work, could benefit from a virtual laboratory environment that provides an integration of mathematical and statistical tools with AI methods to assist in modeling and simulation. One learns from reading *Artificial Intelligence, Simulation, and Modeling* (John Wiley and Sons, New York, 1989, 556 pages, $44.95) that a significant amount of research relating to the integration of AI and simulation is under way. This integration will not only help those doing modeling and simulation go beyond what currently available tools allow, it will also give those developing knowledge-based systems the opportunity to draw on the advantages of modeling and simulation to arrive at some conclusions. Editors Lawrence E. Widman, Kenneth A. Loparo, and Norman R. Nielsen succeeded in providing us with a broad view of current research along these lines.

The many authors contributing to this book explore, on several levels, the relationships between AI and simulation and what challenges and promises these relationships suggest. On the one hand, certain similarities stand out: Both AI and modeling and simulation are concerned with the representation of an external reality. Predicting an outcome using simulation is analogous to forward chaining, and mathematical optimization is analogous to backward chaining. On the other hand, AI and simulation are concerned with representations that are used in different ways. Numeric simulation is used for what-if questions, but diagnosis and explanation by symbolic reasoning are used for why questions. Execution on computing machinery can be different as well: AI tools do not lend themselves to the advantages of parallel architectures as well as mathematical tools do, and systems that attempt to integrate AI and mathematical approaches pose a real challenge to the development of fast software and hardware.

There is great potential, however, for AI and simulation to take advantage of each other's strengths. Because modeling and simulation is a complex undertaking, requiring expertise in several fields, AI tools might be able to help the "simulationist" better handle the complexity of the modeling and simulation process, leading him(her) through the iterative cycle of devising and testing new models. Likewise, AI systems might be able to use modeling and simulation to arrive at decisions for complex systems where simple heuristics are not appropriate or feasible.

Artificial Intelligence, Simulation, and Modeling includes many chapters on these concepts and others, including the use of qualitative reasoning and qualitative simulation, the execution of integrated AI-simulation systems on machines of advanced architecture, formalisms and languages for integrated systems, and specific examples of systems under development. My focus here, biased by my work in modeling and simulation, is on the work described on qualitative reasoning and simulation.

One of the key challenges to integrating AI and simulation is the successful integration of qualitative and quantitative information. Several chapters concern qualitative reasoning and qualitative modeling and simulation, including the use of mathematically based models by qualitative reasoning in AI systems and the use of qualitative approaches in a modeling situation. The qualitative approach is not as commonly applied in modeling and simulation as the quantitative approach (for example, minimizing an objective function that describes how well a model fits the available data). Typically, the systems challenging us are complex, and we are lucky if we can make any qualitative statements about the behavior of a system other than, perhaps, statements regarding stability or long-term steady-state behavior. Only the simplest of systems can be analyzed in terms of its qualitative behavior. This reason is one of the main ones for performing simulation: Intuitive analysis can seldom answer the important questions about complex systems. We tend to obtain quantitative impressions of a system first, then use these impressions to infer qualitative aspects of system behavior. Therefore, the introduction of tools that assist in the qualitative analysis of dynamic systems will greatly benefit the modeling community. Currently, applying the qualitative reasoning approach even to simple systems is not for the faint-hearted, as these chapters demonstrate. Major advances are necessary to make the tools more easily and widely applicable.

Do the research projects described in this book meet the needs of end users? From several of the descriptions of the projects under way, it was not clear to me whether the teams that are tackling these projects include domain experts or whether opinions from domain experts are being solicited. I presume that it will often be a domain expert who wishes to use an integrated AI-simulation system to help him(her) at work. Some kind of evaluation by domain experts is a must; otherwise, much time and effort can be wasted in inventing the wrong wheel. End users should have input from the beginning and be given frequent opportunities to provide direction to an AI-simulation integration team as a project progresses. However, this book does express some caution about the use of powerful integrated systems by naive users.

With respect to breadth of coverage and potential readership, *Artificial Intelligence, Simulation, and Modeling* does provide a broad survey of current research, but it is written from an AI perspective and will find a greater readership among AI researchers than simulationists. Because the field of integrated AI-simulation approaches is rapidly developing, future survey texts will be needed, and hopefully, they will be assembled for workers both inside and outside AI. The work of long-time simulationist David Garfinkel at the University of Pennsylvania and his colleagues on the integration of AI and simulation in the study of enzyme kinetics (Garfinkel et al. 1987; Soo et al. 1988) is an example of work that should receive more attention in future texts.

The AI perspective is also shown in the references provided at the end of each chapter; they are up to date and useful in pointing to additional sources. However, the reference lists do not include enough references to reports that have appeared outside the AI and simulation literature, such as many of Garfinkel's.

The material in this book is well organized but uneven in its accessibility to the reader. Some chapters are a joy to read, but others are needlessly technical and contain many pages of unnecessary detail. This result is not unusual when many different authors contribute to a book, but it is an important factor that must be addressed to meet the needs of the reader. The most appropriate readership for a book of this type includes AI researchers, simulationists, and graduate students in AI or simulation. However, because of the approach chosen by the editors and the level of detail given by many of the authors, this book would not make a good resource for many readers other than AI researchers.

Despite these shortcomings, this book succeeded in making me aware of a great deal of research. It has also made me anxious to see the first commercially available integrated systems, so I can begin using one myself. *Artificial Intelligence, Simulation, and Modeling* is a welcome effort that shows us some of the exciting computing technologies we have to look forward to.

References

Garfinkel, D ; Kulikowski, C. A ; Soo, V.-W.; Maclay, J.; and Achs, M. J 1987. Modeling and Artificial Intelligence Approaches to Enzyme Systems. *Federation Proceedings* 46:2481–2484.

Soo, V.-W.; Kulikowski, C. A.; Garfinkel, D.; and Garfinkel, L. 1988. Theory Formation in Postulating Enzyme Kinetic Mechanisms: Reasoning with Constraints. *Computers and Biomedical Research* 21:381–403

Mark E. Lacy is manager of computational biology at Norwich Eaton Pharmaceuticals, Inc., Norwich, NY 13815. ■

Expert Systems in Business: A Practical Approach

John Musgrove

The cover to *Expert Systems in Business: A Practical Approach* by Michael L. Barrett and Annabel C. Beerel (Ellis Horwood Limited, Chichester, England, 1988, 259 pages, $36.95, ISBN 0-7458-0269-9) contains an abstract design in colors of violet, brilliant green, and dark magenta. Thus, the book is difficult at first to take seriously as a technical book. After seeing other Ellis Horwood books, however, it appears that the use of technicolor covers is the publisher's approach to product differentiation on the information technology bookshelf.

Although this book presents many of the ideas and issues previously covered by others, particularly Paul Harmon and David King in *Expert Systems: Artificial Intelligence in Business* (Wiley, 1985), the authors present their own experience in Great Britain. In a short preface, Barrett and Beerel list 10 principal aims, among which are to identify the business benefits that can be obtained (from applying expert systems), provide a step-by-step strategy for identifying potential expert system applications, and show how an organization can get started in expert systems and achieve early returns. Aims that the authors believe are unique to their presentation include providing insight into how experts participate in an expert system project and how they can accomplish their tasks and giving a detailed account of knowledge engineering in practice, with a focus on what the knowledge engineer actually does and how s/he must behave.

All the aims were addressed in the book, although the chapters on software and hardware were not tied to the aims and were not presented in sufficient depth. This problem might in part result from the availability in Great Britain of fewer of the expert system shell products that are commonly available in the United States.

The authors begin by providing a short but competent introduction to the field of expert systems. Of particular value are the ideas regarding the relevance of expert systems to business and the idea that knowledge is know-how. Throughout the book, the authors frequently tie their concepts to their relevance in a business setting. The emphasis on human know-how and its application in expert systems, as opposed to the less specific concept of human knowledge, sets this presentation apart from others.

This book is clearly concerned with business applications for expert systems as opposed to research applications. A strong emphasis on application selection is made to assure that developers focus their efforts on the benefits to an organization's core business unit. Four good checklists for application evaluation are addressed. Although these checklists are good, they are similar to other sources' criteria for selection and do not provide any new concepts or considerations. The authors make a good presentation for the use of outside services for developing initial applications to limit the costs and frustrations of starting a new endeavor within the organization. They also recommend developing an in-house knowledge engineering team for the long term but give it little attention.

I disagree with the authors' contention that an expert system group should not be made part of the data processing department to avoid the risk of having expert systems used as just another software technique. I believe that expert systems are a computer technique for solving problems just as other programming methods are. Also, truly useful expert systems are not wholly self-contained but operate in conjunction with databases and other computer information systems. Earl Sacerdoti, formerly of Teknowledge, recently observed that the expert system industry has moved away from the Ptolemaic view that the universe revolves around expert systems. The industry has moved to a Copernican view that expert systems are one solution among many that revolve around business problems.

The technology section of this book provides a good explanation of the inferencing process. Only Ken Pedersen's *Expert Systems Programming* (Wiley, 1989) provides a better discussion of the processes involved. The authors give a good description of the necessary criteria for software tool selection, focusing on the importance of the user interface, the developer interface, and the interfaces to external computer systems. They review the published methods of building systems. They present concise, three-paragraph discussions of five approaches to knowledge acquisition. Barrett and Beerel include a substantial discussion of uncertainty, but the concept is dismissed because of the variation in implementation for different software systems. The authors' emphasis on know-how systems leads them to preferentially recommend rule-based production systems.

The discussion of software tools is the weakest part of this book. The chapter on computer software is short, the product descriptions are minimal, and the authors make no comparisons. This situation might reflect the status of software availability in Britain; it is not applicable to the American market. In contrast, Barrett and Beerel provide good criteria for software tool evaluations. The discussions of computer hardware are also generic and limited. The authors seem to contradict themselves by alternately saying "in the future we expect to see all workstation vendors offering delivery... via personal computers" (p. 130) and "a likely pattern for the future is development on personal computers and delivery via mainframes" (p. 131). The relative roles of the large workstations and mainframes versus personal computers is left unresolved. Potential developers would do well to supplement this section with *Expert Systems: Tools and Applications* by P. Harmon, R. Maus, and W. Morrissey (Wiley, 1988).

By contrast, the section on building effective systems is the strongest part of the book. The implementation overview is particularly good, having been drawn from the authors' experience in system building. The different development styles available, based on the goal of the effort, are well presented. The presentation and discussion of spider diagrams to represent areas of knowledge differs from other published presentations. The authors further define and develop this concept in the chapter on building the system. This chapter presents a systematic and reasoned approach to developing the problem and the knowledge and implementing the solution. This presentation is useful if you choose not to use an outside consultant, as recommended. The two chapters on being a knowledge engineer and being an expert are unparalleled and effective. The authors discuss the selection and training of knowledge engineers. Also discussed are issues concerning how the knowledge engineer should interact with the expert. Tricks of the trade for handling typical knowledge engineering situations are presented. The authors' discussions about being an expert are unique and unavailable in similar books. The beneficial characteristics of an expert are discussed as are the expert's possible emotions throughout the project.

The final chapter is short but effective. A much longer discussion of project management can be found in *Crafting Knowledge-Based Systems* by J. Walters and N. Nielsen (Wiley 1988). The authors provide a good description for a project team for project control and documentation. The possible distribution of the team effort—the cost in time and person-days—is presented for a small project.

The chapter "Conclusions and Recommendations" comprises many good summaries. Here, as throughout the book, the use of bulleted lists helps readers focus on important concepts and items. Although the authors used extensive cross-referencing to other chapters within respective sections, greater referencing to chapters in other sections would have been beneficial.

The list of references is short but includes both old and new material. The appendixes include the typical Glossary, Applications, Sources of Information, and Example Costings. The glossary is more than sufficient for a book of this kind. The list of well-known applications is shorter than that of other books, but the broad list of potential application areas is singularly good. The sources of information are limited and not current for U.S. periodicals. Again, this limitation might result from the authors' location in Great Britain. Additional evidence of the British source of this book is in the example costings; they are given in pounds sterling.

In summary, *Expert Systems in Business: A Practical Approach* is a good addition and companion to other books on the subject. The sections on expert system technology and the building of effective systems are valuable to anyone expecting to consider using expert systems in a business environment.

John Musgrove is group leader for expert systems in the Houston regional office of Bechtel Corporation He evaluates expert system applications and directs expert system development projects He received a B.S. in mechanical engineering from Southern Methodist University in 1968 and a M S in engineering from the University of Houston in 1973. ■

Genetic Algorithms in Search, Optimization, and Machine Learning

Terry Rooker

Genetic algorithms are one of those ideas that have been germinating for some time. They can trace their roots to the mid-1960s.

Genetic algorithms provide an alternative to traditional search techniques by adapting mechanisms found in genetics. The basic idea is simple: The parameters of a problem are translated into some encoding, usually represented as a binary string. A population of strings is then randomly generated. Some measure of fitness is then applied to each of the strings. The *goodness* of the fit determines the string's chances to continue influencing the search. The more fit the string, the more likely it is to be selected to create part of the next generation.

A series of operators is used to create this next generation. Three common operators demonstrate the selection procedure. First is *reproduction*. This operator selects the strings to be operated on by the other operators. The selection should be related to the measured fitness of the strings. *Crossover* is then used to combine parts of different strings. In its simplest form, a random point is selected, and the parts of the two strings beyond this point are simply exchanged. *Mutation* is the rare occurrence of changing one of the bit values.

As the number of fit strings increases in the population, the more similar the strings are likely to be. In this case, the reproduction and crossover make ever-finer distinctions in the overall fitness of the strings. Eventually, a string is found that is good enough for the problem at hand. As with many such methods, if it is left running long enough, an optimal solution can be found.

There are numerous open questions. There is research on the operators to use, coding schemes, and evaluation functions (measures of fitness). This system is only a simple one to outline to concept.

The interesting point about genetic algorithms is that they are essentially variations of blind searches. There is no operator or evaluation based on the next likely step. All evaluation of the current state is based simply on how well this state satisfies some measure of fitness.

As the search through the genetic algorithm space proceeds, the operators are building short segments of strings that each contribute to the solution. These segments are appropriately called *building blocks*. As they are discovered, they increase the overall fitness of their including string. If the crossover operator destroys this building block in a string, the overall fitness of the offspring should decrease. As the incidences of this building block increase, it is more likely to be included in new offspring. As this process continues, the strings get closer to some optimum configuration.

Genetic algorithms have some advantages over other optimization or blind-search techniques. Most optimization methods evaluate individual points and then try to evaluate the best direction for the next move. As David Goldberg points out in *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Reading, Mass., 1989, 412 pages, ISBN 0-201-15767-5), considering only a single location at a time leaves such algorithms susceptible to getting caught in local minima (or maxima). Genetic algorithms avoid part of this problem by using only an objective function (the measure of fitness). Because there is no notion of direction in the search, genetic algorithms do not need derivative information or any of the complex methods for calculating the next-best move.

Genetic algorithms also have *implicit parallelism*: As the different building blocks are generated, the algorithm is essentially searching a large space of possible building blocks. As the best building blocks are found, they are incorporated into more successful strings. In this sense, the search is proceeding in parallel. Goldberg mentions that this parallelism is two orders of magnitude greater than the size of the input. That is, for n strings in the population, n^3 building blocks are evaluated. Such a speedup would have obvious effects on the efficiency of the search.

The computer science field in general and AI in particular have heard many of these "too good to be true claims." The important question in this case is whether genetic algorithms can live up to their promise. They have some theoretical advantages over traditional techniques. To truly evaluate their potential, a series of comparative experiments are needed. Under controlled circumstances, genetic algorithms and other algorithms can then be compared. Unfortunately, such a comparison is probably not possible. As previously explained, genetic algorithms only use an objective function. Most other techniques use some other information to determine the direction of the search. There is a problem with comparing such different methods. It is never conclusive whether differences in performance are a function of the underlying functions or the different algorithms.

Goldberg discusses the performance of genetic algorithms for the traveling salesman problem, a standard benchmark for optimization. He points out that the genetic algorithm does not perform as well as other methods but reminds us that the genetic algorithm is also solving the problem without information such as the city distances. Once again, there is only a measure of fitness and not any function to determine the next-best direction for search. Any such comparison of genetic algorithms with other approaches is always open to such comments.

This comparison highlights the main strength of genetic algorithms. They do not need domain-specific information, just some measure of how well each generation of strings fits some criterion. There is no need to develop mathematical models so that there is some function to use to evaluate the next step. There is no need for the computational overhead of artificial neural networks that implicitly perform such calculations inside the network. In such cases, even if not superior, the genetic algorithms might prove easier to implement.

Although genetic algorithms have been around for 25 years, there is a small group of individuals researching the various questions. They have tremendous promise given such limited development. There has been a recent surge of interest in genetic algorithms, and maybe this renewed interest will produce solutions to more of the questions.

Goldberg's Genetic Algorithms has proven to be the de facto handbook of the field. He has written a concise, detailed description of genetic algorithms. He uses the right mix of examples and theory. He starts with simple descriptions and works up to the open questions. He is also frank about the limitations of the approach and equally enthusiastic about its promise. If you are interested in genetic algorithms in any capacity, then Genetic Algorithms is the best place to start.

Terry Rooker is a system engineer at the Naval Surface Warfare Center in Dahlgren, Virginia. He is working as a system analyst on a command and control system. ■