

# Book Reviews

## Knowledge-Based Systems

*Simon Y. Blackwell*

The two-volume set entitled Knowledge-Based Systems (Volume 1, *Knowledge Acquisition for Knowledge-Based Systems*, 355 pp., and Volume 2, *Knowledge Acquisition Tools for Expert Systems*, 343 pp., Academic Press, San Diego, California, 1988), edited by B. R. Gaines and J. H. Boose, is an excellent collection of papers useful to both commercial practitioners of knowledge-based-systems development and research-oriented scientists at specialized centers or academic institutions. The set is the result of a call for papers to support the first American Association for Artificial Intelligence Knowledge Acquisition for Knowledge-Based Systems Workshop, held 3–7 November 1986 in Banff, Canada. Although the conference was held three years ago, these volumes are still timely and sorely needed. Few books dedicated to knowledge acquisition exist.

The first volume, *Knowledge Acquisition for Knowledge-Based Systems*, begins with a paper whose title sounds appropriate: "An Overview of Knowledge Acquisition and Transfer" by the editor B. R. Gaines. The first three pages of the paper are clear and concise. Unfortunately, it then evolves into a rambling philosophical discussion of humanity as a distributed anticipatory system. Obscured by this discourse are several useful observations about the knowledge-acquisition process, for example, that knowledge can be transferred through cultural, verbal, technical, metaphorical, mathematical, and transcendental means.

The next nine papers in the volume present a variety of knowledge-acquisition techniques and training methods. All are good, the most notable being "Generic Tasks for Knowledge-Based Reasoning: The Right Level of Abstraction for Knowledge Acquisition" by T. Bylander and B. Chandrasekaran. Chandrasekaran's papers are usually illuminating, and this one does not fail: He and Bylander re-examine such traditional beliefs as knowledge

should be uniformly represented and controlled and the knowledge base should be separated from the inference engine.

The final 10 papers in volume 1 discuss generalized learning and rule-induction techniques. They are interesting and informative, particularly "Generalization and Noise" by Y. Kodratoff and M. Manago, which discusses symbolic and numeric rule induction. Most rule-induction techniques focus on the use of examples and numeric analysis such as repertory grids. Kodratoff's and Manago's exploration of how the two complement each other is refreshing. Because of their technical nature and the amount of work it would take to put their content to use, most of the papers in this section of the volume are more appropriate for a specialized or research-oriented group.

For those just getting involved in knowledge-based-systems development, *Knowledge Acquisition Tools for Expert Systems* is the more useful volume. In addition to discussing the tools themselves, most of the papers contain details of the knowledge-acquisition techniques that are automated, thus providing much of the same information which is available in the first volume. As an added benefit, they also often discuss the underlying architectures for solving domain-specific problems. For instance, the details of the medical diagnostic architecture laid out in "Design for Acquisition: Principles of Knowledge System Design to Facilitate Knowledge Acquisition" by T. R. Gruber and P. R. Cohen are almost as useful as the discussion of how to build a knowledge-acquisition system.

Volume 2 is particularly germane given the rise in commercial interest about automated knowledge acquisition following this year's introduction of Neuron Data's NEXTRA™ product and last year's introduction of Test Bench by Texas Instruments. Test Bench is actually discussed in "A Mixed-Initiative Workbench for Knowledge Acquisition" by G. S. Kahn, E.

H. Breaux, P. De Klerk, and R. L. Joseph. This volume provides the background necessary to evaluate knowledge-acquisition tools such as NEXTRA, Test Bench, and AutoIntelligence (IntelligenceWare). The vendors of knowledge-based-systems development tools, for example, Inference, IntelliCorp, Aion, AI Corp., and IBM, would do well to pay heed to these books because they point the way to removing the knowledge bottleneck from knowledge-based-systems development.

Overall, the papers in both volumes are comprehensive and well integrated, a sometimes difficult state to achieve when compiling a collection of papers resulting from a small conference. The collection is comparable to Anna Hart's *Knowledge Acquisition for Expert Systems* (McGraw-Hill, 1986), but it is broader in scope and not as structured.

The arrangement of the papers is marred only by an overly brief index. Few readers can be expected to read a collection from beginning to end, and a better index would facilitate more enlightened use. Less important—but nevertheless distracting—is the large number of typographical errors in both volumes.

In conclusion, the set is recommended for both the commercial and research knowledge-based-systems practitioner. Reading the volumes in reverse order might be more useful to the commercial developer given the extra information available in volume 2.

## Neurocomputing: Foundations of Research

*Terry Rooker*

*Neurocomputing: Foundations of Research* (The MIT Press, Cambridge, Massachusetts, 1988, 729 pp., \$55.00), edited by James A. Anderson and Edward Rosenfeld, is a collection of 43 seminal papers and book sections, each prefaced by several pages of introductory comments. The book

traces the development of foundational ideas in neurocomputing. In the process, the editors outline how these ideas influenced and were influenced by developments in neuroscience.

Many selections in the book are classics and are already considered standard references. Because many of them are out of print or were originally published in relatively inaccessible sources, this book does a great service by collecting them in one place. Although one can always second-guess the editors of any collection and argue for or against the inclusion or omission of a paper or book section, overall the editors did well in making their selections. Their choices are consistent with the theme of the book.

Care must be taken when compiling book excerpts to avoid taking ideas out of context.... By staying within the framework of the book and including chapter prefaces, Anderson and Rosenfeld avoided this problem. Their comments on Minsky and Papert's book *Perceptrons*, for example, help dispel the myth that the book was solely responsible for killing neural network research in the 1960s. In reality, the problems with neural models were well known, and Minsky and Papert only formalized these difficulties and provided the final blow.

The prefaces are the strength of this book. The general introduction is an excellent, brief presentation on neural networks or neuromorphic systems. The editors explain the neurobiological inspiration of neural models and how these ideas are carried over into neurocomputing. The details of organizing and implementing neural models are outlined. This overview provides a good explanation of basic concepts for a newcomer to the field and is a good summary for others. This approach of explaining the basic ideas and their effect on the field continues throughout the book.

Interestingly, the first selection is a chapter from a book by William James that was published in 1890. The chapter summarizes *associationism*, one of the spiritual forefathers of neural modeling. Although other older selections, some centuries older, could have been used, the use of this selection makes an important point—that connectionist ideas did not start in the 1940s.

Many chapters emphasize the influence of neurophysiology and biology on at least some researchers. For example, for many neurophysiologists, the brain is not designed as a computation device but as a survival mechanism. Further, the brain is designed to function in a particular environment, and this environment has a great impact on both the form and function of the brain. At the same time, not all neuromorphic systems are intended to model the human mind. The introductory comments to these chapters are insightful and provide the necessary perspective for each selection as well as trace the development of the field.

In the afterword, the editors draw some conclusions about the current status and future possibilities of neural networks. These comments are more speculative than those given in the various introductory statements. The value and potential of neural networks are current topics of heated debate. The editors have attempted to make safe comments, that is, comments which are optimistic but conservative enough to escape the criticism of skeptics. A few of these comments are particularly intriguing.

The editors imply that connectionism is taking cognitive science by storm. It is true there is currently a great deal of interest, but only part of the field is getting on the bandwagon. There is still active interest in rule- or symbol-processing theories. There is still interest in more conventional noncomputational theories of mind. Classical conditioning is experiencing a modest revival. Anderson and Rosenfeld say the mind is proof of connectionist models. However, supporters of these other models make the same comment. It is yet to be demonstrated how much of the brain's behavior can be accounted for by connectionist models.

The editors raise the interesting idea that neural models will intrinsically make mistakes: "This is the dark side of 'creativity' and 'generalization'." Many neural models perform some kind of hill climbing and can get stuck in local maxima. Such local maxima can be interpreted as mistakes. It is not at all certain whether any such mechanism is responsible for mistakes of the mind. The claim that two algo-

rithms are identical because they both make mistakes seems weak. It is possible that mistakes and slips are the consequence of other factors, such as the size and complexity of the brain. Current connectionist models use as many as  $10^4$  fully interconnected nodes. The brain has  $10^{11}$  neurons with as many as  $10^4$  connections from a single neuron. It is an open question whether connectionist models can be meaningfully applied to such massive networks.

Anderson and Rosenfeld state that many concerns about scaling will be relieved with the development of special-purpose hardware. Unfortunately, not everyone working on this hardware shares this rosy forecast. Dan Hammerstrom at the Oregon Graduate Center is investigating the connectivity problems of constructing neural units on very large-scale integrated chips (D. Hammerstrom, *A Connectivity Analysis of a Class of Simple Associative Neural Networks*, Technical Report, CS/E-86-009, Oregon Graduate Center, 1988). Such problems include routing the connections between units and increased transmission delays to distant units. As the number of neural units and their connectivity increases, these problems get worse.

A related issue is the weak computational power of current neural networks. The editors point out that the study of simple networks might provide insights into human cognition and that these simple networks might be grouped into larger functional groups for more powerful computations. Some evidence exists that the brain functions similarly, but it has not been demonstrated.

The editors make an important point that often gets lost in the debate over the utility of connectionist models compared to rule-based models. AI theories (and other cognitive theories) are high level; that is, they are concerned with the structure of the knowledge. Neural models are low level; that is, they are concerned with the structure of the machinery. There is great potential in pursuing both approaches and integrating them. Too many researchers treat these approaches as antagonistic. Such attitudes inhibit investigating the potential of integrating these approaches.

Anderson and Rosenfeld wrote an interesting book. If you are familiar with neural networks, they have

some insightful comments. If you are not familiar with them, you can learn by tracing the development of some foundational ideas. This book should have a deserving place on any bookshelf.

## **Design Automation: Automated Full-Custom VLSI Layout Using the Ulysses Design Environment** *Marwan A. Jabri*

*Design Automation: Automated Full-Custom VLSI Layout Using the Ulysses Design Environment* (Academic Press, Boston, Massachusetts, 1988, 463 pages) by Michael L. Bushnell deals with an interesting and challenging idea in very large-scale integration (VLSI) computer-aided design (CAD): the implementation of a top-down design process based on a script. A system called Ulysses that implements such a script-based approach within a blackboard architecture is described. The primitive operations in the script are the actions of specialized design tools such as floor planners, automatic layout generators, and signal routers. Ulysses is destined to be used in a top-down design methodology, which achieves compact and efficient VLSI circuits, now primarily a manual, time-consuming process.

On the positive side, the book is generally well written and well presented. The chapters are well structured, and the flow of ideas within them is smooth. Paragraphs chain well, and arguments are generally well constructed.

On the negative side, I found it disruptive to have Chapter 3, "The VLSI Design Process," coming between Chapter 2, "Prior Design Environments," and Chapter 4, "Using AI in Environments." To appreciate prior design environments, readers need to have a feeling for the VLSI design process. Reading Chapter 3 before Chapter 2 is a solution. Also, few graphic supports are used in the presentation, which is surprising for a text on CAD and VLSI. The quality of the figures is mediocre when you consider what can easily be achieved with today's desk top publishing systems. Additional work on the (few) figures used would have greatly helped the reader.

Although the book deals with many important technical issues, I have a number of concerns. The biggest concern is that although the major

theme is interesting, the actual implementation is quite poor and disappointing. The idea of the top-down design script is to have an environment where CAD tools can easily interface and be used in the design process, without the burden of file and syntax transformations. Therefore, the contribution of the work described in the book would be based on the demonstration of such a script system, with a number of design tools to prove the usefulness of the idea. However, only two tools were integrated (the third was manually simulated, which itself is awkward) in the script environment, which considerably reduces the power and authority of the demonstration. This disappointing demonstration might be the result of the project's ambitious nature and the immaturity of the CAD tools as low-level operators to a high-level design system.

The book is misleading in its treatment of some key points. The text reads:

### **Student Volunteers Needed**

If you are an undergraduate or graduate student enrolled in a degree program at any college or university and would like to assist the staff members during AAAI-90 in Boston, Massachusetts (29 July-3 August 1990), please contact the AAAI office. In exchange for 12 hours of assistance, you will receive a complimentary conference registration, a copy of the AAAI-90 conference proceedings, and a special AAAI-90 T-shirt. For further information, please send your name, address, telephone, adviser's name, and E-mail address to Volunteer Coordinator, AAAI, 445 Burgess Drive, Menlo Park, CA 94025. E-mail: [NCAI@aaai.org](mailto:NCAI@aaai.org).

Also Weaver is not set up for interactive control of routing. The designer's input can be manually added to the blackboard before routing begins, but he cannot change this information once Weaver has begun to route. It would be highly desirable if the designer could interrupt the routing system and add his own manually-constructed routes of subnets when he observed that the router was floundering. Of course such interruption capability would require an additional user communication expert, so that the user would not have to edit raw blackboard structures directly. (p. 51)

The author is criticizing the capability of the Weaver system (a knowledge-based circuit interconnections router) to restart, continue (that is, to be interrupted), or accept that a user might specify some routing channels. This criticism implies a misunderstanding about what current state-of-the-art routing tools are able to achieve. The problem here is not the communication expert in the blackboard architecture. Any routing expert will have difficulty trying to deal with incomplete (partial) solutions while being interrupted and having its current solution modified. Expecting the router to continue under these circumstances is too much.

A big surprise in the book's technical content was that important work in AI-based VLSI CAD was not referred to, yet seemingly unrelated work was repeatedly cited. Here are some examples:

The work of the AT&T group of Brian Ackland was not even mentioned. This group has completed a number of AI-based VLSI CAD systems, including Cadre, which is the most relevant to the book. However, the book mentions the Redesign system (pp. 55-56), which deals with MSI TTL circuits, which are only distantly related to VLSI circuits. Bushnell also discusses the Vexed and Critter systems:

The work on Vexed and Critter looks extremely promising; however, the work is not really extendible to the VLSI design domain because these systems do not consider chip floor planning, cell placement and routing problems.

Bushnell's mention of SADD, also an MSI TTL target system, is another reference to an irrelevant system:

SADD appears to be successful, but like Vexed, it is difficult to extend to the VLSI IC design domain because SADD does not consider floor planning, cell placement and routing. (p.59)