
WORKSHOP REPORT

Coupling Symbolic and Numeric Computing in Knowledge-Based Systems

C. T. Kitzmiller and J. S. Kowalik

Presented is a discussion of several issues raised during the workshop sponsored by the American Association for Artificial Intelligence on Coupling Symbolic and Numeric Computing in Expert Systems, which was held on 27 to 29 August 1985 in Seattle, Washington. Issues include the definition of coupled systems, motivations for coupling, coupled system architectures, and key factors in the design of coupled systems.

Two major reasons for the recent interest in coupling numeric and symbolic computing emerged during the workshop. The first reason is a need to help the computer user solve problems that require specialized knowledge or expertise. In many situations, users need guidance and counseling in order to solve the problem at hand. The solutions to many problems in business, science, and engineering depend on the application of sophisticated numeric algorithms or techniques. In such situations, users often need help in determining which specific algorithm or technique should be employed and in interpreting any computed results. In other situations, the need is more basic--for guidance in determining whether the problem at hand can be solved and, if so, whether the resources that can be brought to bear are sufficient.

Historically, however, traditional computing environments have stressed precision and speed and have offered the user little real guidance in using the computing resources at their disposal. Until recently, users have been left, for the most part, to their own devices when determining how to apply computing to the task at hand or when interpreting the results of traditional programs. Coupled systems promise to integrate the explanation and problem-solving capabilities of expert systems with the precision of traditional numeric computing.

The second major reason is a need for computing capabilities of increased power and usefulness, capabilities that transcend the limitations of traditional approaches. Among those who need such capabilities are scientists and

engineers who deal with problems involving ambiguous, contradictory, and imprecise data. Even though many elements of the design and analysis tasks they perform can be modeled mathematically or computed using numeric methods, many elements are not sufficiently defined or understood to be amenable to traditional algorithmic or symbolic techniques. A fully autonomous robot, for example, requires a truly intelligent control system--one that combines techniques from artificial intelligence (AI), control theory, and operations research (Kowalik et al. 1986). Although such a robot could utilize traditional techniques to perform many routine tasks, sophisticated techniques are needed to handle many of the humanlike functions. In these instances, computing capabilities more robust than those traditionally available are needed.

We believe that integrating formal mathematical methods and methods based on symbolic knowledge--the coupling of symbolic and numeric computing techniques (see figure 1)--is key to the development of computing methods capable of solving some of the problems currently deemed intractable.

What Is a Coupled System?

Any system linking both numeric and symbolic computing processes could be considered a coupled system. In fact, many of the expert systems with which we are familiar couple symbol-

ic and numeric computing techniques to some extent. MYCIN, PROSPECTOR, DENDRAL, HEARSAY-II, and most of the vision- and image-processing systems are coupled in some sense. However, a significant difference exists in how expert and knowledge-based systems utilize the numeric computing processes embedded within them. In many, the numeric processes are simply functions blindly invoked to provide numeric values. The system has no knowledge of the numeric processes and needs none. In other processes, the referenced systems, the numeric processes are recognized as special processes, and the knowledge-based system component reasons about the application or results of these processes.

MYCIN, for example, utilizes a numeric algorithm to compute certainty factors associated with asserted facts. As implemented, MYCIN needs no special knowledge of the numeric algorithm, nor does it reason about how or where the algorithm should be applied. If, however, MYCIN needed to select an algorithm to use from among several and determine when it should be applied, then it would require special knowledge of each algorithm it is to manage.

In order to establish this distinction, the workshop adopted a definition of coupled systems that required the coupling to be special. Coupled systems, by definition, link symbolic and numeric computing in a manner not found in conventional expert or knowledge-based systems. Coupled systems must have some knowledge of the numeric processes embedded within them and must reason about the application or results of these numeric processes.

According to this definition, MYCIN and systems that invoke numeric processes in a similar manner would not be considered coupled systems. MYCIN would, however, be considered a coupled system if it utilized symbolic techniques to select one certainty factor algorithm from several to use in a particular instance.

Although this definition appears to require that a symbolic process always be "on top," controlling the numeric process, the possibility of a numeric process being in control of a symbolic

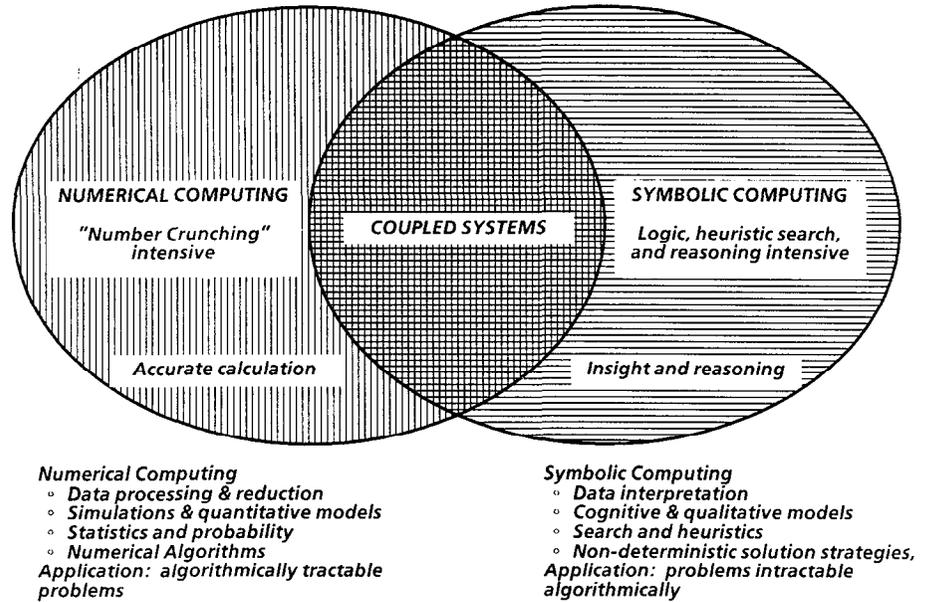


Figure 1 Computing Techniques

process cannot be ruled out. For example, a numeric routine could invoke a symbolic process or expert system to determine the value of a variable. In order to have knowledge or reasoning capabilities, however, a numeric process could not be a pure numeric or mathematical algorithm but must possess symbolic capabilities. We believe that numeric algorithms alone are only able to procedurally invoke symbolic processes. In any event, such structures present several interesting problems, for example, methods within a numeric algorithm to handle uncertainty or incomplete results.

Why Couple?

Often, there is no choice. Separately, neither symbolic nor numeric computing can successfully address all problems in design and analysis. Complex problems such as the control of a fully autonomous robot or the design of an aircraft cannot be solved by purely symbolic or numeric techniques. In such cases, a mix of numeric and symbolic techniques is needed to obtain a solution.

In order to automate the design of an aircraft, for example, numeric computing techniques such as computational fluid dynamics, probability and statistics, and aircraft subsystem modeling need to be integrated with sym-

bolic manipulation techniques such as database management, test generation and planning, and constraint propagation. Without a robust coupling of symbolic and numeric programs, the aircraft design process cannot be fully automated.

Many tasks less complex than the design of aircraft, such as solving a set of nonlinear simultaneous equations, can also prove intractable to traditional symbolic or numeric techniques. The limitations of individual numeric techniques, however, can often be overcome if several different techniques are applied in the proper sequence. In many instances, such a capability can be accommodated by a knowledge-based system that is able to manage the numeric techniques.

Often, coupling is not required, but either the symbolic or numeric processes are enhanced by it. For example, intelligent interfaces might not be necessary, but often they greatly improve the utility of numeric programs. Numeric programs can be enhanced by the incorporation of nondeterministic solution strategies; explanation capability; and the use of programming techniques such as constraint propagation, semantic nets, and frames. Symbolic computing can sometimes be enhanced by the acquisition of procedural knowledge in the form of numeric algorithms (algorithms capable of

replacing computationally intensive search routines) and by the improved precision of numeric processes.

It has been pointed out by Chandrasekaran (1983) that even if we have a complete mathematical model of a situation, this model by itself might be insufficient for many tasks because qualitative reasoning is required to interpret the numeric values of the various problem variables. Thus, the solution of complex problems often compels us to switch back and forth between formal analysis and qualitative reasoning.

Shallow Versus Deep Coupled Systems

Two basic approaches to coupling symbolic and numeric computing processes emerged at the workshop. Each approach parallels a general approach to building expert systems.

The first and more common approach is to develop coupled systems that essentially treat the numeric routines as "black boxes." These systems, referred to here as shallow coupled systems, have little knowledge of the involved processes. Typical uses are managing the application of numeric algorithms to achieve a solution and interpreting the results of numeric routines. Because little is known about each numeric process, the functions of such shallow systems are accomplished through analysis of the problem's state variables. The sequence in which individual routines are applied during the solution process, for example, is often determined by observing the effect each process has on the state variables. The knowledge that is represented might take, depending on the application and performance considerations, the form of shallow rules defining the relationships between different state variables and the acceptable values each state variable can assume.

Shallow coupling is utilized by a Carnegie-Mellon University (CMU) expert system to solve nonlinear algebraic equations (Talukdar et al. 1986). This system manages the application of several different numeric programs during the solution process. The

expert system component determines which program to apply depending on how each affects convergence of the state variable. This system is reportedly able to solve sets of equations that none of its constituent numeric programs can solve individually.

A second coupling approach is to develop systems utilizing extensive knowledge of each process. Such deep coupled systems explicitly represent, depending on the objective of the application, the process's function, inputs and outputs, purpose, usage constraints, side effects, limitations, and the like. Knowledge of each process is integrated with other information and used directly by the knowledge-based system component during problem solving. As might be expected, deep coupled systems can be applied to a wider range of problems than shallow systems. The deep system approach, for example, can be used to provide an intelligent interface to numeric routines.

A system being developed by North American Rockwell to help analysts configure and utilize simulations (Abernathy et al. 1985) is such a deep coupled system. The engagement analyst's apprentice (EAA) is intended to assist the user in constructing a simulation from simulation modules and to evaluate the ensuing results. EAA employs extensive knowledge of each simulation module in order to achieve its objectives.

To date, few deep coupled systems are fully operational. As a result, the relative advantages of the deep and shallow coupling approaches have not been positively established. However, each approach has perceived advantages, stemming from the extent of knowledge explicitly represented.

Among the potential advantages of the deep coupled approach noted during the workshop are performance and robustness. Because of the explicit representation of each process's usage requirements and limitations, deep coupled systems should be more robust and provide a higher level of performance than shallow coupled systems. Knowledge of each process's operating envelope should enable an intelligent selection of the most appropriate process to apply. Similarly, because key information about each

process is explicitly represented, deep systems should be easier to extend and maintain than shallow systems.

Major disadvantages of the deep coupled approach are the overhead associated with the deep coupling and the time required to develop the initial application. Both disadvantages result from the amount of knowledge that must be explicitly represented.

The advantages and disadvantages of shallow coupled systems are the direct opposites of those for deep coupled systems. Shallow coupled systems are expected to be most useful in those situations where a few different processes (two to four) need to be coupled, little is known about individual processes, or the ability to rapidly develop an initial application is needed.

Applications

Within the larger context of counseling the user or creating a new problem-solving capability, most applications have as a specific objective one or more of the following four elements.

One objective is to extend and enhance the capabilities of existing numeric programs. A primary goal for many systems is to provide new problem-solving capabilities by extending the functionality of existing numeric programs. As noted previously, even complete mathematical models are inadequate for many tasks. Most of the applications presented augment the traditional (numerically oriented) computing environment with rule-based shells to manage the application of numeric routines.

In addition to systems such as the CMU expert system presented by Talukdar et al. (1986), several applications utilize symbolic techniques to integrate independent numeric programs and methods (Campbell and Olsen 1986; Chalfan 1986; Nachtsheim et al. 1986; Simmons and Dixon 1986). Other applications utilize symbolic techniques to couple causal and qualitative models with numeric simulations, yielding expert or knowledge-based simulations (Ferguson, Siemens, and Wagner 1986; Hart, Barzilay, and Duda 1986; Lounamaa and Tse 1986).

A second objective is the intelligent user interface. Many systems are aimed at providing guidance and access to users of numeric algorithms and techniques. Typical functions are helping the user select and use numeric processes (intelligent front ends) and assisting the user in interpreting the results by numeric processes (Abernathy et al. 1985; Hart, Barzilay, and Duda 1986; Love et al. 1986; Sivasankaran and Jarke 1986). The EAA system (Abernathy et al. 1985) discussed previously has as one of its major roles that of an intelligent interface to numeric simulations.

A third objective is learning and induction. Several systems have as a goal the extraction of new knowledge from numeric processes and data. Typical applications are extracting classification and relational rules and structures from test and model data (Briggs 1986; Cooper and Kornell 1986; Pao and Bozma 1986) and extracting knowledge from a sensitivity or parametric analysis of numeric simulations and algorithms (Gladd and Krall 1986; Wellman 1986).

A fourth objective is intelligent processing. An aim of several systems being developed is a reduction in the expense of such tasks as computational aerodynamics. In some sense, these applications are the opposite of those being developed to extend system capabilities.

Many number-crunching intensive problems cannot be solved because of insufficient computing resources. Several factors have a direct effect on whether a computational bottleneck is encountered. One factor is the complexity of the computation (that is, size of program, number of iterations, and so on) that must be accomplished in order to obtain a solution. Another factor is the number and combination of inputs during the computation.

The applications presented at the workshop attempt to reduce the computing resources and time associated with expensive numeric processes by substituting simplified algorithms where appropriate or by approximating expensive numeric algorithms with stored results (Briggs 1986; Kant 1986).

Of particular interest are several systems being developed as generic coupling shells or languages (Borchardt

1986; Tompkins 1986). One system, Simple Tool for Automated Reasoning (STAR), is a general language to couple scientific application programs (Borchardt 1986). STAR is of interest because it employs an object-oriented interface to couple numeric routines to a rule-based expert system shell, it is implemented in the C language, and it is available through the National Aeronautics and Space Administration's (NASA) Computer Software Management and Information Center (COSMIC) office.

Architectural Considerations

As might be expected, no single architecture was judged best for all applications. The coupled systems presented during the workshop utilize a spectrum of system architectures and coupling mechanisms. At one end of the spectrum are systems such as the CMU system (Talukdar et al. 1986), which contain a few numeric routines and utilize simple coupling schemes. At the other end are expert simulation and other systems (Abernathy et al. 1985; Ferguson, Siemens, and Wagner 1986; Sivasankaran and Jarke 1985) that incorporate extensive numeric routines and require sophisticated coupling interfaces.

A large percentage of the coupled expert systems presented at the workshop are of the shallow coupling variety. Typically, the interface between the symbolic and numeric processes is simple-communication by way of a global memory and a clear separation of symbolic and numeric functions. Typical functions performed by the symbolic processes are the user interface, interpretation of computed results, and overall control of the problem-solving process. The predominant use of numeric processes is for numeric simulations and signal processing. In most of the applications presented, the numeric routines are coupled as stand-alone programs or routines.

It should be noted that in many instances, a shallow coupling approach is not only acceptable but preferable. Many applications need to couple numeric routines for signal processing

or other similar purposes and primarily involve data interpretation. Usually, such coupling does not warrant a deep coupling approach. Often, little is known about individual processes, and shallow rules that relate the problem's state variables to the execution of specific processes provide the needed coupling.

Other applications, however, do require a deep system approach. An intelligent front end, for example, must have knowledge of each process it is to utilize in order to select the most appropriate process from several candidate processes.

Although no universal architecture was presented, several programming concepts appeared very popular. One concept is the use of blackboard architecture as the overall system architecture (Abernathy et al. 1985; Ferguson, Siemens, and Wagner 1986; Guiller-mard and Lagache 1986; Sivasankaran and Jarke 1986; Talukdar et al. 1986). Among the advantages of this approach is the ability to stratify the problem-solving and process knowledge. This ability allows all problem-solving and other metalevel information to be uniformly represented in the blackboard independently of the processes. Individual symbolic and numeric processes can be incorporated as separate multilevel knowledge sources. All information specific to an individual process is represented within the corresponding knowledge source.

A second approach is the use of object-oriented programming techniques to buffer the expert system from the details of individual processes. Several applications (Borchardt 1986; Ferguson, Siemens, and Wagner 1986; Lounamaa and 'Ise 1986; Love et al. 1986; Tompkins 1986) utilize object-oriented shells to encapsulate individual numeric and symbolic processes. Functions typically handled within the shell (and hidden from the expert system component) are the loading of subordinate programs or routines, the instantiation and verification of the routine's inputs, and any other task of a mechanical nature required to invoke each process. Frame-based or record-based structures are often used to represent the attributes of individual processes and

their state variables. Key among the advantages stemming from these techniques are the abilities to distribute the system among parallel processors, to readily maintain and comprehend the system's operation, and to incrementally extend the system's capabilities.

Design Issues

Many issues enter into the design of a coupled system application. Key among these is whether a shallow or deep coupling approach is most appropriate. During the workshop, several factors having a direct effect upon this issue were identified and discussed. Seven factors were considered particularly noteworthy by the workshop attendees.

- The purpose and objectives of the application.

Such a concern is obvious, but it ultimately dictates the approach selected. If, for example, the objective is to interpret the results of numeric processes, then a shallow coupling approach might be sufficient. If, however, the objective is to provide an intelligent interface, then a deep coupling approach is probably appropriate.

- How much knowledge about each process must be explicitly represented?

A deep or shallow decision does not necessarily result in an either-or situation. The decision to adopt a shallow approach should not preclude the incorporation of deep coupling techniques or knowledge.

Within the approach selected, the extent of the knowledge to be represented must be determined: for instance, whether the application requires knowledge of the process's architecture and structure. In general, management of individual processes or the interpretation of results normally only requires knowledge of the function and outputs of a process. Knowledge of a process's structure is usually only necessary if modification of the process's structure, function, or access is desired.

- Importance of flexibility or generality.

A high degree of flexibility or generality

appears to dictate the adoption of a deep coupling approach. For example, a general coupling language should be robust to provide coupling in diverse applications.

- Importance of robustness and performance.

If speed is more critical than robustness or performance, a simpler, less search-intensive coupling is probably desired. Speed and performance considerations might also dictate a need for parallel or distributed processing.

- What needs to be communicated between the symbolic and numeric processes?

Mechanisms enabling the use of approximate, partial, or incomplete solutions need to be considered. If the numeric processes need to deal with confidence factors or other measures of uncertainty, then mechanisms to enable this capability need to be considered.

- How the user and symbolic processes can determine the significance of computed numeric results.

Mechanisms to evaluate the precision of the results in light of the precision of subordinate routines and the user's overall requirements need to be considered.

- How hardware affects the system's design and, ultimately, its performance.

If the application involves the coupling of existing numeric programs or if a high degree of both performance and speed is needed, a heterogeneous coupling of numeric and symbolic processors should be considered. The coupling of large or resource-hungry programs might not be possible without parallel or distributed processing.

Conclusions

Many real-life problems encountered in science and industry require solution techniques that combine AI and conventional computation methods. Typically, these problems have some major subproblems that are amenable to formal quantitative modeling and to numeric computation techniques such as numeric analysis, statistics, and quantitative simulation. Often, however, the overall solution process

requires insight and qualitative reasoning in order to obtain a solution or interpret the results of the solution process. Thus, it is only natural to expect that the coupling of symbolic and numeric computing in knowledge-based systems is needed to solve complex problems. We emphasize that it is not reasonable to expect that symbolic computing, in the guise of conventional expert systems, can supersede or make obsolete the existing forms of numeric computing. A more reasonable approach is a gradual merging of symbolic and numeric computing tools to create a new, robust generation of software tools. These new tools, utilizing knowledge-based system architectures, will provide new problem-solving capabilities and help the user understand and employ quantitative analysis methods.

The methodology of developing coupled systems is still evolving. As a consequence, a general coupling methodology or language applicable in all situations has yet to be fully developed. Only within the last several years have systems begun to be developed with the expressed purpose of providing coupled system development environments or languages. Most applications now in operation have been "custom tailored" and employ a simple, shallow coupling approach. Although several deep coupled systems are under development, most systems are several years from full operational deployment.

Acknowledgments

The contributions of several colleagues are gratefully acknowledged. Of particular note are Elaine Kant (Schlumberger Palo Alto Research Center) and Sarosh Talukdar (Carnegie-Mellon University), whose workshop notes were indispensable, and Ted Jardine (Boeing Computer Services), who assisted in the initial preparation of this paper.

References

- Abernathy, M F.; Bernabe, M L.; Chandler, L J.; Wilkins, M. G.; and Wolfe, M 1985 A Conceptual Design of an Intelligent Front End to Scientific Application Programs Paper presented at 1985 Coupling Symbolic and Numerical Computing Workshop.
- Borchardt, G C 1986 STAR: A Computer Language for Hybrid AI Applications. In *Coupling Symbolic and Numerical Com-*

puting in *Expert Systems*, ed J. S. Kowalik, 169-177. Amsterdam, The Netherlands: North-Holland.

Briggs, R. 1986. How a Complex Numerical Algorithm Can Be Transformed into a Simple Symbolic Algorithm. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed J. S. Kowalik, 137-145. Amsterdam, The Netherlands: North-Holland.

Campbell, S. D., and Olson, S. H. 1986. WX1--An Expert System for Weather Radar Interpretation. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 329-348. Amsterdam, The Netherlands: North-Holland.

Chalfan, K. M. 1986. An Expert System for Design Analysis. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 179-190. Amsterdam, The Netherlands: North-Holland.

Chandrasekaran, B. 1983. Expert Systems: Matching Techniques to Tasks. In *Artificial Intelligence Applications for Business*, ed. W. Reitman, 41-64. Norwood, N.J.: Ablex.

Cooper, D., and Kornell, J. 1986. Combining Symbolic and Numeric Methods for Automated Induction. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 115-122. Amsterdam, The Netherlands: North-Holland.

Ferguson, J. C.; Siemens, R. W.; and Wagner, R. E. 1986. STAR-PLAN: A Satellite Anomaly Resolution and Planning System. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 273-282. Amsterdam, The Netherlands: North-Holland.

Gladd, N. T., and Krall, N. A. 1986. Artificial Intelligence Methods for Facilitating Large-Scale Numerical Computations. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 123-136. Amsterdam, The Netherlands: North-Holland.

Guillermard, J., and Lagache, J. C. 1986. PMS-1: A Real-Time Advisor for the Boron Control in PWR Nuclear Reactors. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 283-294. Amsterdam, The Netherlands: North-Holland.

Hart, P. E.; Barzilay, A.; and Duda, R. O. 1986. Qualitative Reasoning for Financial Assessments--A Prospectus. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 193-208. Amsterdam, The Netherlands: North-Holland.

Kant, E. 1986. Coordinating Multiple Reasoning Techniques in Algorithm Design and Program Synthesis. Paper presented at

1985 Coupling Symbolic and Numerical Computing Workshop.

Kowalik, J. S.; Chalfan, K. M.; Marcus, R. I.; and Skillman, T. L. 1986. Composite Software Systems. Technical Report, Boeing Computer Services.

Lounamaa, P., and Tse, E. 1986. The Simulation and Expert Environment. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 83-100. Amsterdam, The Netherlands: North-Holland.

Love, P. L.; Shuey, R. T.; Spindler, D. A.; Murray, K. S.; and Anderson, K. R. 1986. Geoplanner--A Prototype Expert Assistant for Numerical Processing. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 307-314. Amsterdam, The Netherlands: North-Holland.

Nachtsheim, P. R.; Gevarter, W. B.; Stutz, J. C.; and Banda, C. P. 1986. A Knowledge-Based Expert System for Scheduling of Airborne Astronomical Observations. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 217-230. Amsterdam, The Netherlands: North-Holland.

Pao, Y. H., and Bozma, I. 1986. Quantization of Numerical Sensor Data for Inductive Learning. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik. Amsterdam, The Netherlands: North-Holland.

Simmons, M. K., and Dixon, J. R. 1986. Reasoning about Quantitative Methods in Engineering Design. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 47-57. Amsterdam, The Netherlands: North-Holland.

Sivasankaran, T., and Jarke, M. 1986. Coupling Expert Systems and Actuarial Pricing Models. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 263-272. Amsterdam, The Netherlands: North-Holland.

Talukdar, S. N.; Elfes, A.; Cardozo, E.; Joobani, R.; Rychener, M.; and Benares, R. 1986. A System for Distributed Problem Solving. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 59-68. Amsterdam, The Netherlands: North-Holland.

Tompkins, J. W. 1986. Using an Object-Oriented Environment to Enable Expert Systems to Deal with Existing Programs. In *Coupling Symbolic and Numerical Computing in Expert Systems*, ed. J. S. Kowalik, 161-168. Amsterdam, The Netherlands: North-Holland.

Wellman, M. P. 1986. Reasoning about Assumptions Underlying Mathematical Models. In *Coupling Symbolic and Numerical*

Computing in Expert Systems, ed J. S. Kowalik, 21-36. Amsterdam, The Netherlands: North-Holland.

The Intelligent Choice

ALAN FARLEY & ASSOCIATES, INC., The premier search firm on the west coast, is now available nationwide to staff specific, individual positions or entire projects.

AFA's excellent management team, research and superb counseling staff is available at a moments notice to provide:

- Effective search/qualifying
- Outstanding marketing/research strategies
- 100% satisfaction or money back guarantee protection through Bank of America
- 17% flat rate fee on all positions

Give Bob Stratton a call collect at (818) 986-9865 for any hiring need, or Carol Soriano at (818) 789-9100 if you are looking for that success bound opportunity.

ALAN FARLEY & ASSOCIATES, INC.

14011 Ventura Blvd.,
Ste 307
Sherman Oaks, CA 91423
(818) 990-3110