Zeitschrift: IABSE journal = Journal AIPC = IVBH Journal

Band: 9 (1985)

Heft: J-29: New developments in project management and engineering

Artikel: Knowledge-based expert systems in civil engineering

Autor: Fenves, Steven J. / Maher, Mary Lou / Sriram, Duvvuru

DOI: https://doi.org/10.5169/seals-28292

Knowledge-based expert systems in civil engineering

«Systèmes experts» en génie civil

Wissengegründete Expertsysteme im Bauwesen

Steven J. FENVES
University Professor
Carnegie-Mellon Univ.
Pittsburgh, PA, USA

Dr. Steven Fenves received his Ph. D. from the University of Illinois, where he taught until 1972. His research activities center around computer-aided engineering, with an emphasis on expert systems, standards processing and databases.

Mary Lou MAHER
Assistant Professor
Carnegie-Mellon Univ.
Pittsburgh, PA, USA

Dr. Mary Lou Maher received a B.S. from Columbia University, a M.S. and a Ph. D. from Carnegie-Mellon University. Her research interests include expert systems, robotics and computer-aided instruction.

Duvvuru SRIRAM
Research Assistant
Carnegie-Mellon Univ.
Pittsburgh, PA, USA

Duvvuru Sriram received a B. Tech from I. I. T. M., India and a M. S. from Carnegie-Mellon University. He is currently pursuing a Ph. D. in civil engineering at C-MU under the direction of Dr. Fenves.

SUMMARY
Knowledge-Based Expert Systems (KBES) are emerging as an important tool kit for the solution of many engineering problems. This paper discusses important differences between KBES and conventional programming languages, along with the applications and implications of this technology to civil engineering problems.

RÉSUMÉ
Les «Systèmes experts» (SE) sont en train de devenir un outil important dans la solution de plusieurs problèmes en génie civil. Cet article compare les SE et les langages traditionnels de programmation, et décrit les applications et les conséquences de cette technologie en génie civil.

ZUSAMMENFASSUNG
Die wissengegründete Expertsysteme (WGE) zeigen sich als eine wichtige Instrumentengruppe für die Lösung vieler technischen Probleme. Diese Arbeit beschreibt wichtige Unterschiede zwischen den WGE und den konventionellen Programmierungssprachen sowohl auch die Anwendungen und Beziehungen dieser Technologie in den Problemen des Bauwesens.

1. INTRODUCTION

The use of computers in civil engineering to date has been extensive, but limited to areas in which algorithmic solutions are available. While these programs have provided invaluable help in the *well-formed* aspects of civil engineering, notably in analysis based on incontrovertible physical laws, they have been largely inadequate in the *ill-structured* aspects, such as design, because of the constraints imposed by conventional programming on the problem solving process. This paper examines the nature of conventional programming and the restrictions it places on problem solving. It is shown that a broader range of problem solving strategies can be addressed by the application of knowledge-based expert systems. The nature of knowledge-based expert systems is described, followed by a discussion of the range of their applications and illustrations of some existing and potential applications to civil engineering. The paper concludes with a discussion of the implications of knowledge-based expert systems on civil engineering education, practice, and research.

2. ANATOMY OF CONVENTIONAL PROGRAMS

In order to put knowledge-based expert systems into proper context, the nature and limitations of conventional computer programs must first be examined. A conventional program can be viewed as consisting of a set of sequentially executed *rules* in the form:

IF *condition* THEN *action*.

Thus, IF *input* THEN *compute* is the overall model of any program, where *compute* is further subdivided into more detailed rules, e.g., a steel design program may contain the rule:

IF *section is compact* THEN *allowable stress* = $0.66\,F_y$.

The conditions and actions are deeply intertwined, as the action of one rule becomes a component of the condition of another rule, thus

IF *a and b and c ... are satisfied* THEN *section is compact*

IF *actual stress* <= *allowable stress* THEN *stress constraint is satisfied*
etc.

Only a person knowledgeable in the application domain, an *expert*, can define the appropriate conditions and the resulting actions. This is particularly true in practical engineering programs, where many rules are not based on the *causality* of physical laws, but represent the *heuristics* (assumptions, limitation, rules of thumb or "style") of the expert or his organization. Thus, the program sketched may contain a rule such as

IF *trial section not compact* THEN *choose another section*

because the expert may want to insure that yielding is the governing limit state.

Developing a complete set of rules for any engineering application is a major undertaking. However, a person writing a conventional algorithmic program, a *programmer*, incurs three additional major professional responsibilities:

1. He must determine the *sequencing* of the program, that is, the fixed order in which the rules will be executed;
2. He must guarantee *completeness* of the program, that is, that the program performs the correct actions for every possible combination of conditions -- even if the action is only an error message for combinations not specifically provided for; and
3. He must similarly guarantee *uniqueness* of the results, that is, that every combination of conditions leads to one action only (this is not an independent requirement, since the programmer's choice of sequencing automatically assures a unique set of actions -- although they may not necessarily be

the *correct* ones for each combination of conditions).

Even when the above criteria are met, there are further assumptions and limitations in conventional programs. A conventional program assumes all data has been input and is without error, while in many cases only incomplete and uncertain data are available. Also, the program functions as a *black box* with no mechanisms to explain how it arrives at the computed results. The user must refer to external documentation, or the code listing itself, to determine the problem solving approach. And finally, the program contains only limited mechanisms for controlling the problem solving approach. It solves each problem in only one predefined and preprogrammed way -- an all or nothing situation.

Applications based on mathematical models and those requiring intense numerical computations may be conveniently built using conventional programming techniques. However, many problem solving strategies, such as interpretation and design, are *ill-structured* or *ill-defined* [22], and are not well suited to the rigid algorithmic format. The use of knowledge-based expert system techniques, as described in the next section, provides a means to computerize such applications.

3. KNOWLEDGE-BASED EXPERT SYSTEMS

Knowledge-based expert systems (KBES) have recently emerged from research in artificial intelligence as practical problem-solving tools that can reach a level of performance comparable to that of a human expert in some specialized problem domain. Surveys of expert system concepts are given in [6], and in recent books [10, 12].

A number of formalisms have been developed to represent the knowledge in a domain. One such representation is the production system (PS) model [4] illustrated in Figure 1. The principal feature of expert systems based on the PS model is that a clear distinction is made between the *knowledge-base*, containing the model of an expert's problem-solving knowledge, and the *control strategy* or inference mechanism which manipulates the knowledge base. In addition to the heuristic surface knowledge, which consists of IF-THEN production rules encoding empirical associations based on experience, the knowledge-base can incorporate fairly deep knowledge comprised of algorithmic procedures. Whereas the knowledge-base is specific to a given domain, the control strategy is completely general.

The process of using a KBES is as follows. The user enters some known facts about the problem into the context, the part of the KBES that contains the knowledge about the particular problem at hand. Following its control strategy, the inference mechanism locates the potentially applicable rules -- those whose condition portion is *matched* by the facts in the context -- selects one of these and *fires* it, that is, causes its action to be executed. The result of any action is to add to or modify some aspect of the context; thus, new rules become candidates to be fired, and a cylce of matching and firing is repeated in an "infinite loop" until a goal is satisfied or there are no more rules remaining to be fired.

All KBES possess, in some form, a knowledge-base, a context, and an inference mechanism. Some KBES contain one or more of the following additional facilities.

1. <u>Treatment of imprecise or incomplete knowledge.</u> Many expert systems provide the ability to deal with imprecise or incomplete knowledge, where rules are represented as

   IF *condition* THEN *action* WITH CERTAINTY

   e.g.,
   IF *eccentricity* > *10%*
   THEN *limit state is beam-column failure* is very likely (0.9)

   The inference mechanism can then propagate certainty measures about the inferences along with results of the inferences.
2. <u>Explanation facility.</u> It is very important that the expert system explain to the user its inference

process. A good explanation facility can provide explanation both a-priori (why a certain fact is requested) and a-posteriori (how a certain conclusion was reached).

3. Knowledge acquisition facility. Since the knowledge-base is built up incrementally, it is important to have a facility which permits the addition of new rules. Ideally, the expert should be able to add to or modify the knowledge-base as sessions on the expert system reveal gaps in the knowledge-base.

A number of languages and domain independent frameworks (DIFs) are available for building KBES. The languages commonly used are Lisp, Prolog, and OPS5 [9]. These languages require that the expert system be structured entirely by the KBES builder and they can address a wide range of problems. A DIF is an inference mechanism with an empty knowledge-base; a knowledge acquisition facility is usually available to aid in the development of the knowledge-base and to structure the context for a particular application domain. Many frameworks also provide facilities for treatment of imprecise data and explanation. A number of frameworks have been developed from domain dependent expert systems by providing a similar inference mechanism with an empty knowledge-base. Examples of such systems are EMYCIN [26], which evolved from work on MYCIN [21] and KAS [18], which developed from PROSPECTOR [5]. These DIFs may be used for problems similar to the problem that the original system addressed. KBES are not limited to mainframe computers. Several frameworks are now available for personal computers, such as INSIGHT, DECIDING FACTOR, and M1. The personal computer frameworks are restricted in their applicability; they can typically address problems such as diagnosis and classification. However, they provide an environment for rapid prototyping of KBES, thus avoiding costly investment in the development of KBES.

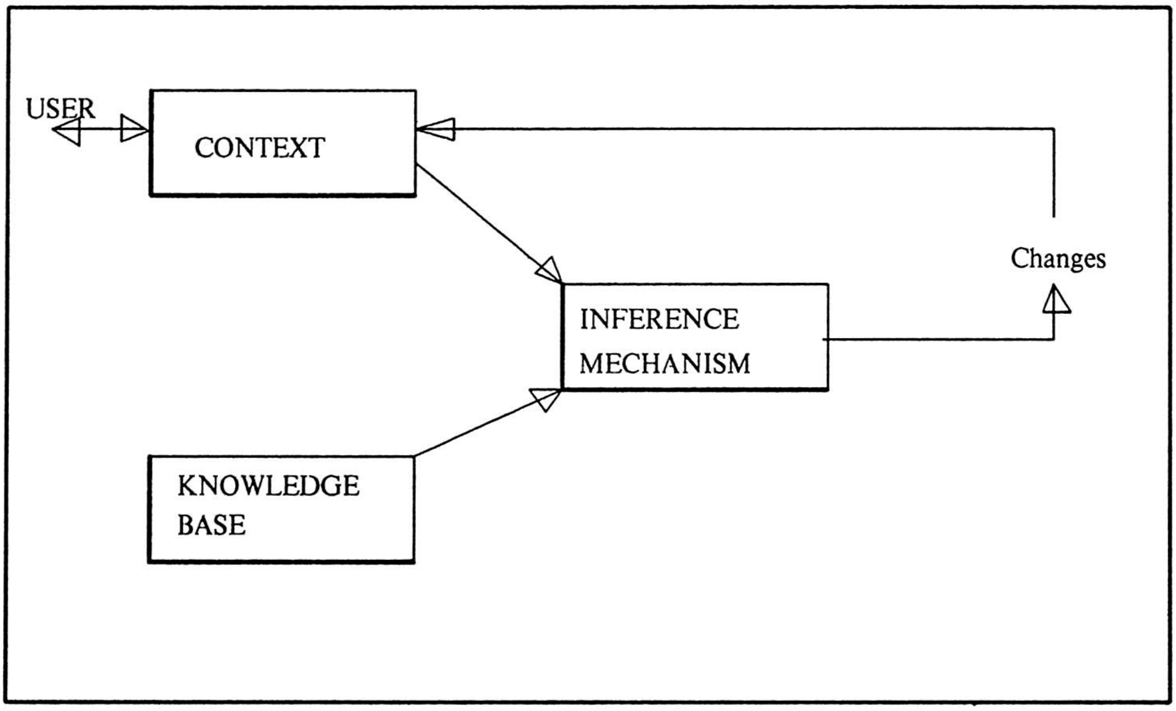

Figure 1: A Schematic View of a Production System Model

A KBES differs from a conventional program in the following aspects:

1. Sequencing. Unlike the instruction sets in a conventional program, the rules in a PS model are not executed sequentially. The sequencing is the responsibility of the control strategy adopted. There are many control strategies in common use, such as forward chaining, backward chaining, means-ends analysis, and agenda control. In forward chaining the system works from known facts to a

goal state. In backward chaining the systems starts from a hypothesis or goal, checking the facts to see if the hypothesis can be supported. Agenda control processes tasks according to an agenda, or list of goals, ordered by a priority rating. The control strategy adopted may depend on the framework used.

2. Completeness. Completeness need not be guaranteed in an expert system, at least not in the development stages. If there are no rules in the knowledge-base for the situation at hand, new rules can be acquired from the user.
3. Uniqueness. Non-determinism can be incorporated so that a given set of facts may lead to one or more conclusions. If a mechanism for treating imprecise knowledge is included in the KBES, the possible conclusions can be ranked on the basis of their likelihood or certainty.

4. RANGE OF EXPERT SYSTEM APPLICATIONS

The range of potential KBES applications covers a spectrum bounded by *derivation* problems and *formation* problems at the ends [1]. In derivation problems, known solutions are present in the knowledge base and the KBES derives the best solution for the specific problem at hand. On the other hand, in formation problems, components of the solution are present in the knowledge-base and the KBES forms a solution from these. components for the specific problem at hand. In real life, most problems fall between these two extreme categories. Problems normally encountered at the derivation end of the spectrum are:

1. Diagnosis. The problem consists of finding the state of a system based on the interpretation of data (which may be noisy, i.e., imprecise). Some examples in this category are programs such as MYCIN for infectious diseases and DELTA/CATS [3] for troubleshooting diesel electric locomotives.
2. Interpretation. The given data are analyzed to determine their meaning. Examples are the Dipmeter Advisor, for interpreting dipmeter data and extracting information about geological patterns and trends [23], and PROSPECTOR, for identifying ore-bearing geological formations.
3. Monitoring. Signals are interpreted continuously and required changes are made depending on the state of the system being monitored. An example is the Ventilation Manager [7] for monitoring patients' ventilatory therapy.

Formation problems are usually examples of the *generate-and-test* paradigm: a possible candidate solution is generated by one part of the system and is then tested for suitability by another part of the system. Formation problems fall into two subclasses: *constraint satisfaction* and *optimization* problems. In constraint satisfaction the solution need only to satisfy a set of constraints, while in optimization an attempt is made to find the optimal solution. Problems usually encountered at this end of the spectrum are:

1. Planning. The objective is to set up a program of actions that are required to achieve certain goals. An outstanding example in this category is MOLGEN for planning experiments in molecular genetics [25].
2. Design. Design involves the selection of a physical system to perform a certain function. A very successful example, in a limited definition of design, is R1 [16], that designs VAX computer configurations, including selecting the needed components and determining their physical layout and interconnections.

5. EXISTING APPLICATIONS IN CIVIL ENGINEERING

Expert system methodology is still in its infancy. This section briefly presents some existing expert systems developed for civil engineering problems.

SACON [2] is a KBES to aid a structural analyst in selecting appropriate modeling options in MARC, a large, general purpose finite element program. SACON was developed using the EMYCIN framework. A similar KBES is SesCon [8], a prototype front end for the Seasam-69 structural analysis package.

SPERIL [13] is being developed at Purdue University for the assessment of damage to buildings after a hazardous event, such as an earthquake. It is being implemented in the C language.

HI-RISE [15] is a KBES developed to aid the structural engineer in the preliminary structural design of buildings. Using a three dimensional grid as input, HI-RISE designs alternative structural systems for lateral and gravity loads. HI-RISE is developed in PSRL [20], a frame-based production system language.

CONE [17] is a KBES developed in OPS5 to interpret data from a cone penetrometer (CPT), a field exploration device used in to provide soil stratigraphy information. CONE interprets the raw data from CPT test data to determine the soil type and shear strength of soil strata. CONE represents knowledge from multiple experts by utilizing concepts from fuzzy set theory.

CHINA [11] is a KBES that provides an intelligent interface to a FORTRAN design model. CHINA is a rule based system that embodies the design experience of leading noise control engineers. The expert system interacts with the state of the art noise barrier design model through initializing the execution of the program, interpreting the results, and, if necessary, generating new input data for an additional run.

TRALI [27] is a KBES developed to assist in traffic signal setting for isolated intersections. In contrast to existing computer aids, this system can be applied to intersections of highly irregular geometries. Algorithmic processes to evaluate signal settings and decision tables to identify traffic flow conflicts are invoked by the expert system; phase distribution of flows is performed by applying heuristic rules.

HOWSAFE [14] is a microcomputer-based espert system, using the DECIDING FACTOR as a framework, developed to evaluate aspects of a construction firm's organization and procedures that bear on the firm's safety performance.

DAPS [19] is an expert system for the assessment of structural damage to buried facilities subjected to intense impulsive pressures. This code is written in Lisp and takes advantage of the inference engine of SPERIL-I as the model for the reasoning process.

6. POTENTIAL APPLICATIONS IN CIVIL ENGINEERING

A number of potential applications of KBES in civil engineering are outlined below. The list is by no means exhaustive and is provided to illustrate the potentials of KBES. Potential applications are described for some of the problems in the *derivation-formation* spectrum.

6.1 Interpretation

6.1.1 Existing conditions

The gathering and interpretation of existing conditions is the first step of any civil engineering design. KBES may be used for interpretation of structural condition and load capacity of structures based on observations and non-destructive tests in the absence of adequate design documents, interpretation of traffic conditions and demands for transportation improvements, etc.

6.1.2 Intelligent modelling tools

The model of CHINA, SACON, and SesCon are applicable in many areas where powerful analytical tools are available, but where KBES can serve for *problem identification*, to assist users in generating the appropriate analytical model, and in *result interpretation*, to assist in evaluating the analytical results for appropriateness . With proper facilities, these tools can be intergrated directly with the analysis programs to become intelligent pre- and post-processors.

6.2 Diagnosis

6.2.1 Failure diagnosis

Landslides, rockslides, building failures, water and sewage treatment plant failures, failure of construction schedules, etc. all require collecting and evaluating data of various degrees of certainty to identify the most likely cause of failure. KBES could materially assist in such diagnoses.

6.2.2 Remedial diagnosis

The diagnosis of existing civil engineering systems to determine potential failures and dysfunctions is even more appropriate for KBES because causes of deficiency could be ranked and appropriate remedial and preventive measures recommended.

6.3 Monitoring

6.3.1 Performance monitoring

With the increasingly available microprocessors and sensors providing input to expert systems, real-time monitoring and time-dependent diagnosis may be performed on structures, foundations, reservoir systems, etc. as well as on construction equipment.

6.3.2 Process monitoring

Monitoring of design and construction processes to control costs and duration could be provided by expert systems incorporating knowledge about alternative schemes, reactions to contingencies, etc.

6.4 Planning

6.4.1 Project planning

KBES may assist in the planning of design and construction projects where there is a large range of potential actions, many conflicting requirements, and possible contingencies to consider.

6.4.2 Macro-planning

Planning of large capital projects, such as in infrastructure rehabilitation plans, could similarly be assisted by expert systems, especially where public reaction and preferences strongly influence the plans.

6.5 Design

In all branches of civil engineering, there are potential applications of KBES for the initial synthesis of system function or configuration, for the selection of initial design parameters, for the evaluation and ranking of candidate solutions, and for modification and redesign of unsatisfactory components.

7. EXPERT SYSTEM DEVELOPMENT PROCESS

The development process of an expert system is incremental; a prototype system may be implemented and tested by the expert, and as a result new knowledge added to the knowledge-base. The main reason for this evolutionary development is that a great deal of time must be spent on *formalizing* the expert's knowledge to a much larger extent than he or she had ever done previously.

At the present state of expert system methodology, the generation of an expert system involves the cooperative effort between one or more experts who possess the domain-dependent knowledge and a knowledge engineer (KE) who elicits the expertise and converts it into the system's knowledge-base (akin to the early days of computer programming, where an intermediate programmer was needed).

The major stages of expert system development are:

1. Identification. The first stage is to identify the problem domain to be addressed. Some of the criteria of a "good" domain are:
    - algorithmic solutions are inappropriate, overly constrained or over-specialized;
    - there are experts who perform tasks in the domain notably better than novices;
    - knowledge components are available in textbooks - novices become experts incrementally through practice;
    - typical tasks take an expert several minutes to several hours to complete; and
    - the process could be significantly improved if an expert could work on each occurance of a task.
2. Formalization. This stage involves the identification of key concepts, relations and information-flow characteristics of the problem domain, and the mapping of these concepts and relations into the formal structure supported by an available expert system building framework.
3. Implementation. At this stage the initial knowledge-base is built up from more detailed interviews with the expert.
4. Testing. The knowledge-base is tested and validated using actual tasks, and modified and extended as needed.

The development of practical, operational KBES is not trivial (some of the systems mentioned have involved 5 to 10 man-years). The difficulties are partly procedural, in that current expert system development systems lack several desirable features, notably in interfacing with algorithmic procedures and existing programs. The second set of difficulties is in obtaining cooperation from experts: the experts may be unsure of the value of "programming" their hard-earned knowledge, and they may have trouble in formulating and expressing their knowledge in the form suitable for use in a KBES.

8. IMPLICATIONS

The emergence of expert systems as an additional component of the engineers' "toolkit" promises to introduce changes at least as far-reaching as the entire "computer revolution" to date. This new development has far-reaching implications for engineering practice, education, and research.

In practice, the most direct implication is that computer assistance to engineers can be significantly extended into application areas that were previously intractable. More importantly, engineers will be able to operate *as if* the most expert person in a particular problem area were looking over each engineer's shoulder, providing advice and guidance based on long experience. Also, organizations will be able to "capture" and constructively use the expertise of senior people for many years after their retirement.

The implications on education are equally powerful. Expert systems can serve as training or teaching tools, providing the student or novice "synthetic experience" in dealing with ill-structured problems.

The development of simple expert systems may be an excellent pedagogical tool, permitting the student to organize and formalize his thought processes. One such experiment has been reported in [24].

In civil engineering research, KBES provide for the first time an environment for conducting research on the practical aspects of the profession, namely, on how practitioners use, integrate, and combine elements of textbook knowledge to perform practical and innovative tasks. Beyond this, experimentation with expert systems, particularly with the formalization of concepts and thought processes, may eventually lead to theoretical insights and discoveries which may obviate the need for heuristics.

REFERENCES

[1] AMAREL S., "Basic Themes and Problems in Current AI Research," *Proceedings of the Fourth Annual AIM Workshop*, Ceilsielske, V. B., Ed., Rutgers University, pp. 28-46, June 1978.

[2] BENNETT J. S. and ENGELMORE R. S., "SACON: A Knowledge-Based Consultant for Structural Analysis," *Proceedings Sixth IJCAI*, pp. 47-49, August 20-23 1979.

[3] BONISSONE P. P., "DELTA: An Expert System to Troubleshoot Diesel Electrical Locomotives," *ACM*, New York City, pp. 44-45, October 24-26 1983.

[4] BROWNSTON L., FARRELL R., KANT E., and MARTIN N., *Programming Expert Systems in OPS5*, Addison-Wesley, MA, 1985.

[5] DUDA R. O. et al., *A Computer-Based Consultant for Mineral Exploration*, Final Report SRI Project 6415 SRI International Edition, 1979.

[6] DUDA R. O. and SHORTLIFFE E. H., "Expert Systems Research," *Science*, Vol. 220, pp. 261-268, April 1983.

[7] FAGAN L. M., KUNZ J. C., FEIGENBAUM E. A., and OSBORN J. J., "Representation of Dynamic Clinical Knowledge: Measurement Interpretation in the Intensive Care Unit," *Proceedings Fifth IJCAI*, pp. 1014-1019, 1979.

[8] FJELLHEIM R. and SYVERSEN P., *An Expert System for Sesam-69 Structural Analysis Program Selection*, Technical Report CP-83-6010, Division for Data Technology, Computas, Norway, January 1983.

[9] FORGY C. L., *OPS5 User's Manual*, Technical Report CMU-CS-81-135, Carnegie-Mellon University, July 1981.

[10] HARMON P. and KING D., *Artificial Intelligence in Business: Expert Systems*, John Wiley & Sons, 1985.

[11] HARRIS R. A., COHN L. F. and BOWLBY P. E., *An Intelligent Interface To a FORTRAN Design Model*, unpublished , 1986, [Submitted to the ASCE session on Expert Systems in Transportation Engineering, April 1986].

[12] HAYES-ROTH F., WATERMAN D. A., and LENAT D. B., Eds., *Building Expert Systems*, Addison-Wesley Publishing Company, Inc., 1983.

[13] ISHIZUKA M., FU K. S., and YAO J. T. P., *SPERIL I-Computer Based Structural Damage Assessment System*, Technical Report CE-STR-81-36, Purdue University, November 1981.

[14] LEVITT R., *HOWSAFE: A Microcomputer-Based Expert System for Evaluating the Safety of a Construction Firm*, unpublished , 1986, [Submitted to the ASCE symposium on Expert Systems in Civil Engineering, April 1986].

[15] MAHER M. L. and FENVES S. J., *HI-RISE: An Expert System for the Preliminary Structural Design of High Rise Buildings*, Technical Report, Department of Civil Engineering, Carnegie-Mellon University, 1984.

[16] MCDERMOTT J., *R1 : A Rule-Based Configurer of Computer Systems*, Technical Report CMU-CS-80-119, Carnegie-Mellon University, 1980.

[17] MULLARKEY, P., *CONE: An Expert System for Interpretation of Geotechnical Data from Cone Penetrometers*, unpublished Ph.D. Dissertation, Civil Engineering Department, Carnegie-Mellon University, Pittsburgh, 1985.

[18] REBOH R., *Knowledge Engineering Techniques and Tools in the Prospector Environment*, Technical Report 8172, SRI International, June 1981.

[19] ROSS T.J., SAVAGE S.J. and WONG F.S., *DAPS: An Expert System for Damage Assessment to Protective Structures*, unpublished , 1986, [Submitted to the ASCE symposium on Expert Systems in Civil Engineering, April 1986].

[20] RYCHENER M. and FOX M., *PSRL: An SRL-Based Production System*, Technical Report, Robotics Institute, Carnegie-Mellon University, 1984.

[21] SHORTLIFFE E. H., *Computer-Based Medical Consultations: MYCIN*, American Elsevier, New York, 1976.

[22] SIMON H. A., *The Sciences of the Artificial*, MIT press, Cambridge, MA, USA, 1969.

[23] SMITH R.G. and BAKER J.D., "The Dipmeter Advisor System," *Proceedings Seventh IJCAI*, pp. 122-129, 1983.

[24] STARFIELD A. M., BUTALA K. L., ENGLAND M. M., and SMITH K. A., "Mastering Engineering Concepts by Building and Expert System," *Engineering Education*, pp. 104-107, November 1983.

[25] STEFIK M., "Planning with Constraints (MOLGEN 1)," *Artificial Intelligence*, Vol. 16, pp. 111-140, 1981.

[26] VAN MELLE W., "A Domain Independent Production-Rule System for Consultation Programs," *Proceedings Sixth IJCAI*, pp. 923-925, August 1979.

[27] ZOZAYA-GOROSTIZA C. and HENDRICKSON C., *An Expert System for Traffic Signal Setting Assistance*, unpublished , 1986, [Submitted to the ASCE session on Expert Systems in Transportation Engineering, April 1986].