EXPERT SYSTEMS IN CIVIL ENGINEERING, CONSTRUCTION AND CONSTRUCTION ROBOTICS

by

D.R. Rehak and S.J. Fenves

DRC-12-18-84
December, 1984

Expert Systems in Civil Engineering, Construction and Construction Robotics[1]

Daniel R. Rehak[2] and Steven J. Fenves[3]

1. Introduction

The civil engineering profession covers of a wide spectrum of tasks in the planning, design, construction and management of constructed facilities, including structures, tunnels, environmental protection systems and transportation systems. Computers are an essential tool for the profession, but they are only applied to limited aspects of the total process. In practice, tasks such as finite element analysis of structures, surveying computations, construction accounting, and construction project activity scheduling, are all performed using computer based tools on a regular basis. In fact, some of these tasks are computationally intractable without the aid of computer based tools. However, there exist numerous interesting, abstract and "difficult" problem solving tasks which have not been computerized, or where the available computer tools do little to attack the intelligent aspects of the problem solving process. To this end, researchers in the Robotics Institute's Civil Engineering and Construction Robotics Laboratory are exploring knowledge-based expert systems (KBES) as an approach to extending the range of computer applications in the civil engineering and construction domains.

Knowledge-based expert systems are computer programs, based on artificial intelligence (Ai) techniques, designed to reach the level of performance of a human expert in some specialized problem solving domain. Expert systems have a great potential for practical use in ill-structured problem solving domains where explicit algorithms do not exist or where traditional computer programs provide only restricted problem solving capabilities.

As the civil engineering and construction processes and the construction site increase in complexity, expert systems will increasingly augment or replace conventional algorithmic programs in which all decisions have to be anticipated and pre-programmed. Expert systems are applicable to a wide range of tasks, and many problems in the civil engineering domain, such as structural design, structural modeling, structural diagnosis, site investigation, estimating, scheduling and contingency management, are potential expert system applications. The development of construction robots provides many new applications for expert systems including tasks such as sensor interpretation, equipment diagnosis, operational planning and operational monitoring. Finally, design for robotic construction will be a new domain, and one particularly suited for expert system application.

A variety of potential expert system applications in civil engineering, construction and construction robotics are described below. In addition, research work in progress, developing prototype expert systems for several civil engineering domains, is presented.

---

[1] Prepared for *1984 Annual Research Review,* Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, March 1985.

[2] Assistant Professor, Department of Civil Engineering, Co-Director, Civil Engineering and Construction Robotic Laboratory, Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA 15213, ARPAnet: DANIEL.REHAK@CMU-RI-CIVE.

[3] University Professor, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA 15213.

2. Expert Systems Versus Algorithmic Applications

In order to put expert systems into proper perspective, the nature and limitations of conventional computer programs must first be examined. Any computer program can be viewed as consisting of a recipe of *rules* that completely specifies the problem solving sequence and actions:

IF **condition**$_1$ *THEN* **action**$_1$
IF **condition**$_2$ *THEN* **action**$_2$
⋮
IF **condition**$_i$ *THEN* **action**$_i$
⋮
IF **condition**$_n$ *THEN* **action**$_n$

Each rule contains a premise or condition which is evaluated. If the condition is *true*, the corresponding action is performed and execution continues with the next rule. If the condition is *false*, the action is skipped.

The basic model of any program is a triplet of rules:

```
IF starting a new problem   THEN input problem description
IF input is available       THEN compute results
IF results are computed     THEN output results
```

A process such as **compute results** is further subdivided into thousands of more detailed rules. As illustrated, the rules are intertwined; the action result of one rule becomes the premise of another rule. Traditional computer programs are developed by explicitly stating all applicable rules and their precise sequence of execution. Moreover, often only the action parts of the rules are explicitly stated. Such programs are called *algorithmic.*

Only a person knowledgeable in the domain of the program, a *domain expert*, can define the applicable conditions and corresponding actions. This is particularly true in any practical engineering program, where a very large proportion of the rules are not necessarily based on the *causality* of physical laws, but represent the *heuristics* (assumptions, limitations, rules of thumb or *style*) of the expert or his organization. Developing a complete set of rules for any engineering application is a major undertaking. The rules must satisfy the following criteria:

- *Completeness.* The set of rules must provide an action for every possible combination of conditions. Often, many of the conditions are not explicitly stated. They may be "second nature" to the programmer or user, or the combination of conditions represents a case which is unusual, and thus not expected to occur. However, failure to consider all possible cases results in a program which will not perform in an acceptable manner over the complete range of potential applications.

- *Uniqueness.* The set of rules must provide one and only one unique outcome for every possible combination of conditions. As outlined above, this is difficult to achieve, and if not achieved it results in a program which is flawed.

- *Correctness.* The set of rules must provide a correct outcome for all possible conditions. In addition, the sequence of rules must be correct. Misinterpretation problems occur when the program (programmer) invokes the wrong action for a given condition. Either the action is wrong, or the proper action is not associated with the proper premise. Alternatively, an incorrect sequence applies rules at the wrong time.

The program developer has the responsibility to insure that the three criteria listed above are met. Due to the complexity and size of any real program, this goal is almost impossible to attain, and this makes conventional program development expensive. Additional costs are incurred when programs are updated; the program developer must not only modify or add rules, he also must locate the affected rules in the sequence and modify the sequence itself.

Even if the completeness, correctness and uniqueness criteria are met, there are additional problems with traditional programs:

- The program assumes all data has been input and is without error, while in many cases only incomplete and uncertain data are available.

- The program functions as a *black box* with no mechanisms to explain how it produces the computed results. The user must refer to external documentation, or the code listing itself, to determine the problem solving approach.

- 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 as algorithmic programs. However many problem solving strategies, such as interpretation or design, are *ill-structured* or *ill-defined* [Simon 81], and are not well suited to the rigid algorithmic format. To date, such applications have not been successfully computerized. Numerous civil engineering tasks fall into this category. The use of expert systems for such applications is discussed in the next two sections.

3. Knowledge-Based Expert Systems

3.1. Overview

Knowledge-based expert systems are designed to overcome the shortcomings of algorithmic computer applications. They may be based on the same type of premise-action rules as algorithmic programs, but the sequence of selecting and applying the rules is not specified a priori. The rules are represented symbolically and are treated as *knowledge* which is used by the knowledge processing component of the expert system. This knowledge processor determines which rules are applicable in any situation and invokes the corresponding actions. By design, the set of rules need not be complete or unique; the knowledge processor knows how to deal with such circumstances. Expert systems have additional capabilities for selecting rules based on incomplete or uncertain data, and explaining why rules are selected and how they are used.

3.2. Background

Knowledge-based expert systems have emerged from decades of research in artificial intelligence. They are practical problem solving tools that can reach a level of performance comparable to that of a human expert in some specialized problem solving domain. Rather than being a program with general problem solving knowledge that can be applied to any task, an expert system contains a large body of domain-specific knowledge gathered from human experts. This knowledge is used to perform problem solving tasks in a manner similar to experts. The expert system mimics the behavior of the expert whose knowledge is encoded in the program, and does not pretend to learn or to be innovative. Furthermore, an expert system is highly idiosyncratic, in that it reproduces the knowledge of one domain expert (or, at most, the consensus of a small group); it does not pretend to generate or condense the combined knowledge of several experts.

Expert systems have been developed in a number of disciplines, including: medical consultation *(MYCIN)* [Shortliffe 76], hypothesizing molecular structure from mass spectrograms *(DENDRAL)* [Buchanan 69], computer configuration *(R1, XCON)* [McDermott 80], mathematical formula manipulation *(MACSYMA)* [Moses 71], oil well logging *(Dipmeter Advisor)* [Davis 81, Smith 83], and mineral exploration *(Prospector)* [Duda 79]. Numerous applications are moving from the research laboratory into production (Duda 83]. Many civil engineering problem domains are candidates for expert system formulation [Sriram 82], and several prototypes are under development, as described in Section 6.

3.3. Architecture of an Expert System

The principal distinction between expert systems and algorithmic programs lies in the use of knowledge. A traditional algorithmic application is organized into data and program. An expert system separates the program into an explicit *knowledge base* describing the problem solving strategy and a control program or *inference machine* which manipulates the knowledge base. The data portion or *context* describes the problem being solved and the current state of the solution process. Such an approach is denoted *knowledge-based* [Nau 83].

A variety of expert system architectures exist. Various domain independent systems have different inference procedures and different knowledge representation schemes, including: production systems [Forgy 81], semantic inference networks [Reboh 81], and frame representations [Wright 83]. More complex blackboard systems, which are based on multiple experts operating at different levels of abstraction, have also been built [Balzer 80, Erman 80, Nii 82]. In the production system formalism for domain knowledge representation, the knowledge is represented directly in terms of IF-THEN rules illustrated above. Other types of expert systems use different knowledge representation formalisms. Some of the problem solving strategies incorporated in expert systems are discussed in Section 3.4.

A schematic of an expert system using the production system formalism (IF-THEN rules) is shown in Figure 1. It is to be emphasized than only the knowledge base is domain specific. All the other components are parts of a general purpose expert system building framework applicable to other application domains. The components of the expert system are:

- *Knowledge Base.* The knowledge base is the repository for all the knowledge and rules used by the system in problem solving. This information can be divided into two classes: the factual or *causal* knowledge of the application domain, and the *empirical* associations or rules. The knowledge base may also contain long term historical information and facts. All the information in the knowledge base is organized so that it may be effectively utilized by the other components of the system. .

- *Context.* The context contains all of the information which describes the problem currently being solved, including both problem data and solution status. The problem data may be divided into facts provided by the user and those derived or inferred by the program. The use of an expert system program begins with the user entering some known facts about the problem into the context

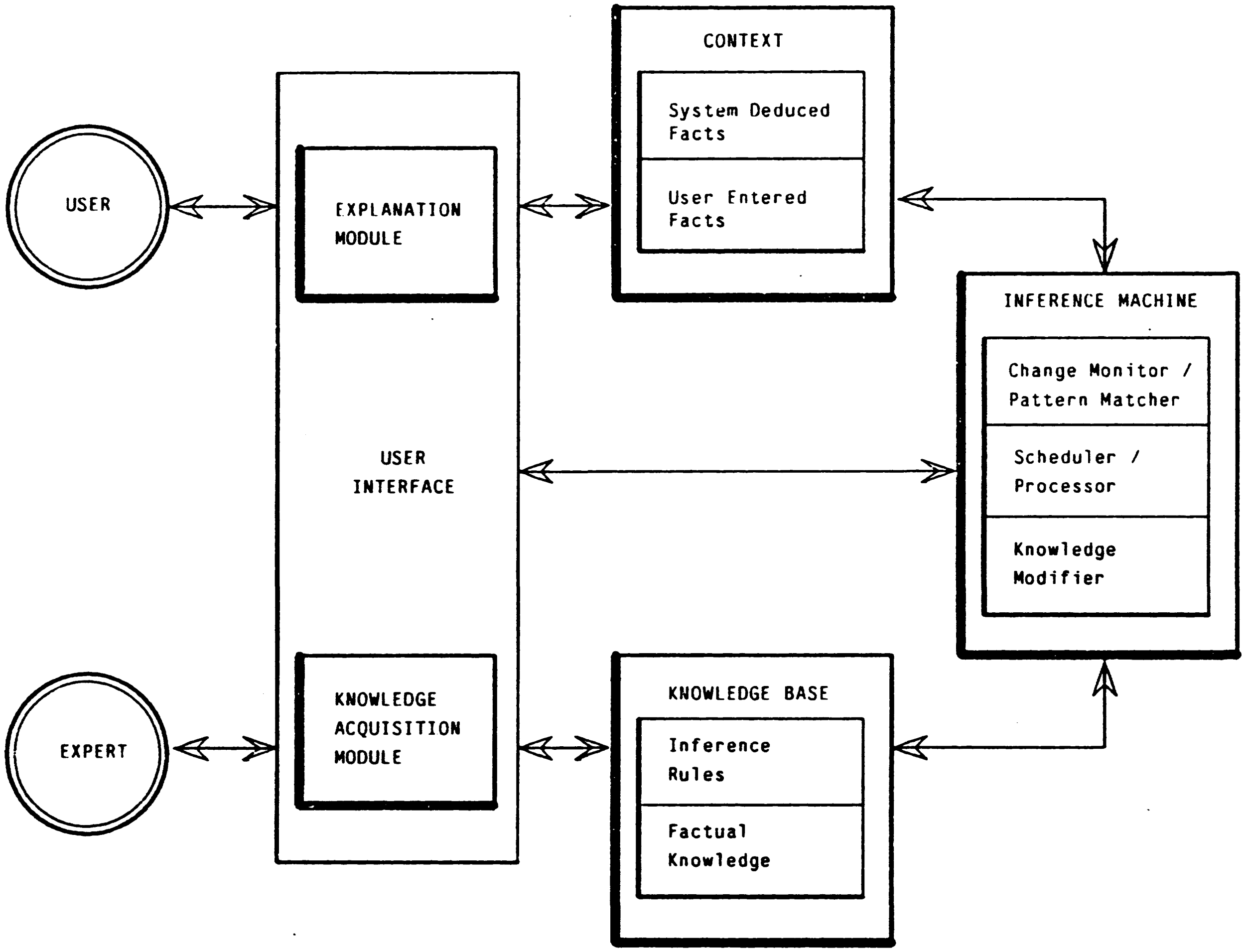


Figure 1: Schematic View of an Expert System

- *Inference Machine.* The inference machine or inference engine is the knowledge processor. It operates on the context, utilizing the rules in the knowledge base to deduce new facts which then can be used for subsequent inferences. The basic operation of a *forward chaining* inference machine, discussed in Section 3.4, is an infinite loop performing three steps:

  1. Examine the premises of rules in the knowledge base and determine which of these currently evaluate to *true*, based on the current problem data maintained in the context. This step, performed by the change monitor or pattern matcher, yields a set of candidate rules.

  2. Select one of the applicable rules. The rule is chosen by the scheduler or processor.

  3. Invoke or *fire* the corresponding action, which will change some items in the context. The context is updated by the knowledge modifier.

  The objective of the inference machine is to arrive at a global conclusion (*goal*), and the process continues until the problem is solved and the context is transformed into the desired goal state, or when there are no more rules remaining to be fired.

Many expert system inference machines can deal with imprecise or incomplete knowledge. Associated with all data are certainty measures indicating a level of confidence in the data. Rules conditionally fire based on the certainty of the premise. The inference mechanism can then propagate certainty about the inferences along with results of the inferences.

- *Explanation Module.* The explanation module provides the expert system with the capability to explain its reasoning and problem solving strategy to the user. At any point the user may interrupt the system and inquire what it is doing and why it is pursuing the current line of reasoning. In addition, the program can explain, in an a posteriori fashion, how any fact was deduced and how knowledge was applied.

- *Knowledge Acquisition Module.* The information in the knowledge base is in a rigid format, and the translation of knowledge obtained from experts to the required internal format may be tedious. The knowledge acquisition module aids in this task. Although it is desired that eventually the human expert be able to enter directly knowledge into the system, this goal is currently not achieved.

- *User Interface.* The user accesses the system through a friendly interface, often using a problem oriented subset of English or computer graphics. The interface provides capabilities for the user to monitor the performance of the system, volunteer information, request explanations, and redirect the problem solving approach used by the expert system.

The basic concepts of a knowledge base, knowledge acquisition, explanation, context and inference mechanisms are common to the different types of expert system architectures. Details of system organization, knowledge and data representation, and inference method vary among the different approaches.

3.4. Problem-Solving Strategies

Problem solving involves the search for a solution through a state-space by the application of operators, where the state-space (the possible states in the problem solution) consists of an initial state, a goal state and intermediate states. A solution path consists of all states that lead from the initial state to the goal state. Domain independent problem solving strategies are commonly referred to as *weak methods* and may lead to combinatorial explosions; the number of states to be explored grows combinatorially with the number of variables describing the problem. Expert systems can be considered *strong problem solvers* since they employ domain knowledge in the solution strategy to reduce the search space. A number of problem solving strategies used in current expert systems are briefly presented below. More detailed descriptions of a number of problem solving strategies can be found in [Nilsson 80, Rich 83, Stefik 77].

- *Forward Chaining.* A system exhibits forward chaining (bottom-up, data-driven, antecedent-driven are all equivalent to forward chaining) if it works from an initial state of known facts to a goal state. All facts are input to the system and the system deduces the most appropriate hypothesis or goal state that fits the facts. The main drawback of this strategy is that it is extremely wasteful to require as input data all the possible facts for all conditions; in many circumstances all possible facts are not known or relevant. This strategy is useful in situations where there are a large number of hypotheses and few input data. Sometimes the problem solving mechanism is guided by the events occurring during the solution process; this type of forward chaining is called *event-driven.*

- *Backward Chaining.* A system exhibits backward chaining (also referred to as consequentdriven, top-down, goal-driven and hypothesis-driven) if it tries to prove a goal or hypothesis by checking known facts in the context. If the known facts in the context do not support the hypothesis, then the preconditions that are needed for the hypothesis are set up as sub-goals. Essentially, the process can be viewed as a search in the state-space going from the goal state to the initial state by the application of inverse operators.

- ***Means-ends Analysis.*** In means-ends analysis the difference between the current state and the goal state is determined and used to find an operator most relevant to reducing this difference. If the operator is not directly applicable to the current situation, the problem state is changed by setting up subgoals so that the operator can be applied. After an operator has been applied, the current state corresponds to the newly modified state. Means-ends analysis utilizes both the forward and backward chaining techniques. However, this strategy is only applicable to those tasks where the measures of differences between the various states and the operators to reduce these differences can be formulated a priori.

- ***Problem Reduction.*** Problem reduction involves decomposing problems into smaller sub-problems. The problem is represented as a tree, in terms of an AND-OR graph. An AND node is a parent of a number of child nodes, all of which must be solved for the AND node to be *true.* For the OR node, it is sufficient for one of the sibling child nodes be solved; an OR node indicates that a number of alternate solutions exist for the problem. In many cases, backward chaining is used to solve the AND-OR graph. A detailed description of an algorithm (AO*) for finding solutions in an AND-OR graph is given in [Rich 83]. This technique is very useful in tackling large complex problems.

- ***Backtracking.*** Backtracking is used when it is determined that an infeasible solution path has been followed. Backtracking withdraws or revises previous decisions. Backtracking is incorporated in many AI languages, such as *PROLOG* [Clocksin 81]. Backtracking, in its pure form, can be very inefficient. Stallman and Sussman [Stallman 77] developed the concept of *dependency-directed backtracking (DDB)* as an efficient way to backtrack from wrong decisions. In DDB, a record of all deduced facts, their antecedent facts along with their support justifications and the relevant rules are maintained; these records are known as *dependency records.* Support justifications are justifications for any assumptions made during the search. When the problem solver identifies an infeasible solution path, it retrieves the antecedents of the *contradiction.* Those dependency records are used to determine which alternative path to pursue. This strategy involves a lot of book-keeping. However, this additional book-keeping helps in a number of ways. For example, explanation of the program behavior can be extracted from the dependency records. This concept was further extended by Doyle [Doyle 78] for systems that incorporate *nonmonotic reasoning.*

- ***Plan-Generate-Test.*** The generate-and-test strategy, in its purest form, generates all possible solutions in the search space and tests each solution until it finds a solution that satisfies the goal condition. The plan-generate-test sequence restricts the number of possible solutions generated by an early elimination of inconsistent solutions. Elimination is performed in a planning stage, where the data is interpreted and *constraints* are formulated and evaluated; these constraints eliminate solutions that are inconsistent with the data and the goal state.

- ***Hierarchical Planning & Least Commitment Principle.*** Hierarchical planning involves developing a plan at successive levels of abstraction. For example, in design of complex systems the design space is divided into a set of levels, where the higher levels are abstractions of details at lower levels; the problem is hierarchically decomposed into loosely coupled subsystems. A number of solutions may exist for each subsystem. However, enough information may not be available to ascertain variables describing a subsystem. Further, the solution to one subproblem may depend on the decisions made in the solution of another subsystem. To minimize this dependency, it is important to defer decisions as late as possible. This principle is called the *least commitment principle* because problem description variables are not instantiated (decisions are deferred) until more information about the problem space is available.

- *Constraint Handling.* If the subgoals in hierarchical planning do not interact with each other, they can be solved independently. However, in practice these subgoals do interact. The interaction between subgoals can be handled by *constraint satisfaction* methods. Constraint satisfaction methods involve the determination of problem states that satisfy a given set of constraints. Constraints may be used to eliminate solution paths, to partially describe solutions, and to express interactions between subproblems. Stefik [Stefik 80] presents an extension to the classical constraint satisfaction method [Mackworth 77] by integrating it into hierarchical planning. This method, denoted *constraint posting,* involves three stages.

    1. *Constraint formulation* is the operation of adding new constraints representing restrictions on decisions and variable values. The constraints contain increasing detail as problem solving progresses.

    2. *Constraint propagation* is the operation of creating new constraints from formulated constraints. This operation handles interactions between subproblems through the reformulation of constraints from different subproblems.

    3. *Constraint satisfaction* is the operation of finding values for problem variables so that the constraints on these variables are satisfied.

- *Agenda Control.* When a human problem solver is required to perform a number of tasks at one time, he gives a priority rating to these tasks. A list of justifications and a priority rating can be associated with each task. The task with the highest priority rating is performed first; an *agenda* of tasks is prepared. This type of control can be used for complex tasks that require focusing attention on certain subparts of the problem solving process. Agendas can also be used in systems that require several independent sources of expertise to communicate with each other.

4. Development of Expert System Applications

4.1. Scope of Expert System Applications

Expert system applications are appearing in many disciplines. However, not all tasks are amenable to expert system formulation. The following is a partial list of criteria for the evaluation of promising potential applications [Hayes-Roth 83]:

- There are recognized experts in the field whose performance is better than that of novices.

- The factual component of domain knowledge is routinely taught to neophytes who become experts by developing their own rules and empirical associations.

- Typical tasks are performed by an expert in a few minutes to several hours.

- Tasks are primarily cognitive, requiring reasoning at multiple levels of abstraction.

- Algorithmic solutions are either impractical or result in overly constrained or specialized programs.

- There are substantial benefits in applying the expert knowledge to each occurrence of the task.

A resulting system should have the following characteristics:

- *Usefulness.* The expert system must be capable of performing useful functions. Usefulness depends on the domain and task for which the expert system is developed.

- *Performance.* The expert system must have a high level of performance, reliability and accuracy over a range of application cases. This requires that the expert system have the specialized knowledge that separates human experts from novices.

- *Transparency.* An expert system is transparent if it can be understood by people using the expert system. To have this characteristic, the expert system must be able to explain its actions and reasoning to the user.

4.2. Range of Expert System Applications

The range of potential expert system applications covers a spectrum from *derivation* or *interpretative* problems to *formation* or *generative* problems [Amarel 78]. In derivation problems, the problem conditions and description are given as part of a solution description *(goal).* The expert system completes the solution description by applying the available knowledge and rules such that the initial data and conditions are well integrated in the solution. As an example, in a derivation problem such as theorem proving, a solution hypothesis is formulated which the expert system attempts to prove by applying rules to the known data. Repeated application of rules transforms the problem statement (the initial state) to the solution state. In formation problems, conditions *(constraints)* are given in the form of properties that the solution as a whole must satisfy. Candidate solutions are generated and tested against the specified constraints. Two subclasses exist: *constraint satisfaction* in which the solution need only satisfy the governing constraints, and *optimization* where an attempt is made to find the optimal solution. The design of a plan, object, or system fits this paradigm. Most actual problems are not pure formative or derivation problems, but lie somewhere in-between and require that techniques from both categories be used in problem solving.

The following list of problem types covers the spectrum of expert system applications.

- *Interpretation.* An interpretation system takes observed data and explains its meaning by inferring the problem state which corresponds to the observed data. Examples are *Dipmeter Advisor,* a system for interpreting geophysical oil well log data [Davis 81, Smith 83], and *Prospector,* a system for identifying geological ore-bearing formations [Duda 79].

- *Prediction.* Starting with a given situation, a prediction system infers likely consequences.

- *Diagnosis and Debugging.* Diagnosis systems infer malfunctions or system state from observed irregularities and interpretation of data. *MYCIN,* an infections disease diagnostician [Shortliffe 76], and several other medical diagnosis programs fall into this category.

- *Monitoring.* A monitor observes system behavior and compares the observations to the planned behavior to determine flaws in the plan or potential malfunctions of the system. An example is *Ventilation Manager,* a program for monitoring a patient's ventilation therapy [Fagan 79].

- *Design.* Design is the process of developing a configuration for an object which satisfies all applicable constraints. *R1* (or *XCON)* is an example of a design system which is used to configure VAX[4] computers [McDermott 80].

- *Planning.* Planning is a design process that yields a set a actions intended to produce a desired outcome. An example is *MOLGEN,* an expert system for planning experiments in molecular genetics [Stefik 8 i].

- *Repair.* Repair systems plan remedies for malfunctions found through diagnosis and debugging.

- *Control.* A control system encompasses many of the characteristics of the other types of applications described above. It must interpret data, predict outcomes, formulate plans, execute the plans, and monitor their execution.

Interpretation, prediction, monitoring and diagnosis lie at the derivation end of the spectrum while design, planning, and control lie at the formation end.

4.3. Languages and Tools for Building Expert Systems

A number of languages and tools are currently available for building expert systems. These tools can be grouped into three categories [Hayes-Roth 83].

- *General Purpose Programming Languages.* Expert system may be implemented directly in a high-level, general purpose programming language. These high-level languages contain some special features, such as facilities for handling large chunks of knowledge and operators for developing planning and reasoning strategies. In addition, these languages contain powerful abstraction mechanisms with which other higher level constructs can be built, making programming flexible and easy. Current expert system have been built using a number of languages, of which *LISP* [Winston 81] and *PROLOG* [Clocksin 81] are very popular among researchers.

- *General Purpose Representation Languages.* General purpose representation languages are programming languages developed specifically for knowledge engineering. These languages are not restricted to implementing any particular control strategy, but facilitate the implementation of a wide range of problems spanning the derivation-formation spectrum. Some general purpose languages are: *SRL* [Wright 83], *RLL* [Greiner 80], *KEE* [Intelligences 84], *OPS5* [Forgy 81], *ROSIE* [Fain 81], *LOOPS* [Bobrow 83] and *AGE* [Nii 79].

- *Domain Independent Expert System Frameworks.* A domain independent expert system framework provides the expert system builder with an inference mechanism from which an application can be built by adding domain specific knowledge. Such systems also provide knowledge-acquisition and explanation modules to simplify the construction of the expert systems. Usually, these frameworks have evolved out of domain specific KBES. Hence, their control strategies are restricted to those provided in the original system. Systems in this category include: *EMYCIN* [vanMelle 79], *KAS* [Reboh 81], *HEARSAY-III* [Balzer 80], *EXPERT* [Weiss 79], and *KMS* [Reggia82].

A number of widely used languages and tools are listed in Table 1. A detailed description of these tools is beyond the scope of this paper and the reader is referred to Part V of [Hayes-Roth 83].

---

[4]VAX is a registered trademark of Digital Electronics Corp.

Tool or Language	Developer	Knowledge Representation	Implementation Language
OPS5	Carnegie-Mellon University	Rules	***LISP* and *BLISS***
EMYCIN	Stanford University	Rules	***INTERLISP***
HEARSAY-III	USC-ISI	Rules	***INTERUSP***
EXPERT	Rutgers University	Rules	***FORTRAN*** .
ROSIE	Rand Corporation	Rules	***INTERLISP***
KS300	Tecknowledge Inc.	Rules	***INTERLISP***
AGE	Stanford University	**Rules**	***INTERLISP***
KAS	SRI International	Rules and Semantic Nets	***INTERLISP***
KMS	University of Maryland	Rules and Frames	***LISP***
KEE	IntelliGentics Inc.	Rules and Frames	***INTERLISP***
RLL	Stanford University	Frames	***INTERLISP***
PSRL	Carnegie-Mellon University	Rules and Frames	***FRANZLISP***
LOOPS	Xerox PARC	Rules and Frames	***1NTERLISP-D***
KL-ONE	Rand Corporation	Semantic Nets	***INTERLISP***
C-PROLOG	University of Edinburgh	**Logic**	***C***

Table 1: Languages and Tools for Building Expert Systems

4.4. Building Expert System Applications

The process of building a complete expert system consists of two distinct steps. The first step is the selection of the appropriate language or framework (inference engine, knowledge base and context structure) from among those listed in the previous section. The second step is denoted *knowledge engineering* and consists of gathering expert knowledge and augmenting a domain independent expert system shell with domain dependent knowledge to provide a fully functioning system.

Knowledge engineering is an incremental cooperative process commonly performed by two people. The first is the *domain expert* who possesses the problem solving knowledge for the problem area being addressed. The second is the *knowledge engineer* who gathers expertise from the domain expert and translates this knowledge into the format required by the expert system. A knowledge engineer who is also literate in the application domain is desired, as he can understand the issues involved and the nomenclature used by the domain expert [Dym 84, Fenves 84a].

The knowledge engineering process of building an expert system application is outlined below [Reboh 81, Hayes-Roth 83]. This process is similar in nature to building an algorithmic program or producing the design of an object.

- *Problem Identification.* The first step in building an expert system is to identify the problem to be solved and the characteristics of the solution. Identification of resources, domain experts, and computer facilities are made and overall goals for the project are set.

- *System Design.* The overall structure of the system is selected. Based on processes used by the expert, available data, strategies, information flow, etc., a preliminary model of problem solving to be used by the system is developed. From this, a problem solving strategy and a candidate domain independent expert system development tool is selected and key concepts are formalized. If the selected system appears to have the correct formalisms for problem solving and knowledge representation, a more detailed analysis is undertaken to produce a detailed design of the system.

- *Knowledge Acquisition.* Knowledge acquisition is the process of gathering the expert knowledge from the domain expert. This process may be difficult. In some instances the cognitive portions of the problem solving process used by the expert have never been verbalized and are difficult to extract (the expert is not conscious of how he solves problems). In other cases the expert may feel threatened by a computerized replacement and will be reluctant to cooperate.

- *Implementation.* The knowledge engineer's tasks is then implementation; the expert's knowledge is encoded in the format required by the expert system framework. This yields a prototype operational program.

- *Testing.* Once a prototype or partial system is developed, it is tested. The domain expert and the program are given the same problem and their problem solving behavior is compared. Flaws in the system are detected and corrected by adding or modifying the knowledge used by the program.

The knowledge engineering process is not sequential, and a number of iterations in the last three steps may be necessary to correct and tune the system's behavior. Adding depth of knowledge, breadth of capabilities, and improved interfaces and explanations to the prototype yields the final version of the system.

5. Potential Expert System Applications in Civil Engineering, Construction and Construction Robotics

Today, there are no production expert systems in the civil engineering or construction domains. In addition to a general survey of the field [Sriram 82], potential applications in construction [McGartland 83, Rehak 84], construction robotics [Fenves 84b], geotechnical engineering [Stanford

83], automated tunnel boring [Sriram 83a] and water resource management [Rehak 83] have been examined by researchers in the Civil Engineering and Construction Robotics Laboratory. Prototypes and studies indicate that the following areas, covering the spectrum of applications discussed in Section 4.2, are excellent candidates for expert system development in the civil engineering, construction and construction robotics domains. However, the list is by no means exhaustive. Development of prototypes in these areas is described in more detail in Section 6.

5.1. Diagnosis

- *Equipment Diagnosis and Repair*. Construction equipment is becoming more complex and harder to maintain. Robotic and automated construction equipment will present even more complex diagnosis and repair problems. Expert system diagnosticians can keep track of the equipment's internal conditions by diagnosing its operating state, make "connections" between spatially and temporarily dispersed signals and events, e.g., distinguish between transient and hard failures, and revise their operational expectations based on previous history. Such systems can act as expert mechanics or mechanic's assistants in suggesting possible fault causes or repair and preventive maintenance procedures based on operational behavior and failure symptoms. A prototype equipment diagnostician is *MOVER* [Fenves 83, Fenves 84c].

- *Structure Diagnosis*. Systems which assess the safety and serviceability of various classes of structures (bridges, buildings, dams, etc.) would take data from tests and descriptions of a structure to assess its integrity (load carrying capacity, potential life, etc.), identify likely causes of deficiencies in the structure, and provide recommendations for remedial actions. Such systems would be particularly valuable in repair, reconstruction and restoration work, where detailed plans are often not available and where many possible modes of deterioration must be considered. An example of such an application is *SPERIL*, an expert system for the assessment of seismic damage to buildings [Ishizuka 81].

5.2. Interpretation

- *Site Investigation*. Much of the information which is the basis for substructure design of constructed facilities and for field operations comes from site investigations. Information is obtained from field and lab tests, visual and other inspection devices, and observations. The data is then presented to an expert geotechnical engineer who interprets the data to characterize the site and makes recommendations concerning possible actions and substructure design. Since such tasks rely almost wholly on human experts using their empirical knowledge, they are excellent candidates for expert system formulation.

  Such applications could be built to perform portions of the expert's post-facto data interpretation, or could be built into intelligent, automated site investigation equipment. The most exciting possibility offered by expert systems is to completely bypass certain exploratory or evaluation operations. For example, with appropriate sensing, interpretation, theoretical correlation and empirical knowledge, one should be able to infer as much geotechnical information from a production rock-drill as one obtains today from geotechnical explorations. An example of such a system for the interpretation of cone penetrometer data is being built [Mullarkey 83]. Another prototype for the inference of a three-dimensional model of site stratigraphy based on borehole and sample test data is being designed [Norkin 84]. Similar pre-interpretation expert systems can be developed for other aspects of the construction environment.

- *Sensing.* Construction is often performed in an environment characterized by unseen and unanticipated conditions. Constructors, both manned and robotic, must transform raw sensor data in a variety of ways to properly interpert them, i.e., build a reliable and usable *world model* of the environment. Many "low-level" transformations such as signal conditioning, smoothing, imaging, feature extraction, etc., can be performed algorithmically — in fact, dedicated hardware or firmware is available for several of these processes. However, "high-level" interpretations which extract *meaning* from the transformed sensor data are more likely to be implemented as expert systems.

  As an example, REX (Robotic Excavator) is a robotic gas pipe excavator under development in the Civil Engineering and Construction Robotics Laboratory for the hazardous task of gas line excavation and repair. A variety of sensors will be used to determine subsurface features such as locations of utilities, prior to excavation [Ray 83]. The data from the various sensing modes will be separately transformed to a common, high level representation (e.g., bounding planes of features "seen" by each sensor). It is anticipated that an expert system will be needed to interpret and integrate the separate sensor images and to build a composite world model of the subsurface features. Such a model would then be used as part of the input to the planning process for excavation. The expert system will have to deal with issues such as:

  o discrepancy between expected position taken from utility maps and sensed position — an obvious heuristic is:

  IF sensed diameter, material, etc. agree with expected
  THEN accept sensed position;

  o discrepancy or conflict between sensing modes;
  o discrimination between pipe valves, tees and other attachments.

- *Quality Control.* In construction, there are major uncertainties in the condition and quality of the emplaced components. A prototype system which uses magnetic imaging to determine the location and size of concrete reinforcement in structures such as bridge decks or nuclear reactor containment vessels is being built [Motazed 84]. Similar to pre-interpretation, there is the exciting possibility to integrate post-interpretation into the process itself. Thus, eventually one should get as much information about the quality of a weld from an instrumented automated welder as one obtains from a post-facto weld inspection.

5.3. Design

- *Design with Conventional Components.* The design of constructed facilities is a complex and ill-defined task. Computer-aided structural design processes today primarily consists of algorithmic analysis, proportioning or optimization programs. The "creative" component of the design task has been largely ignored. *HI-RISE* [Maher 84a] is an example of an expert system for the preliminary design of high rise buildings. Similar expert design systems could be developed for other constructed facilities where preliminary design or synthesis essentially involves the selection, combination and *customization* of relatively standard components (e.g., small-to-medium span bridges, prefabricated buildings and, possibly, certain water treatment and wastewater processing plants).

- *Design with Robot-constructed Components.* The constructed facilities built with the first generation of construction robots will not look significantly different than those built presently, nor will their design process be significantly affected. Eventually, however, the existence of construction robotics will provide major influence on the form and functions of the facilities, and will alter the design process iiself in significant ways. In other industries, this feedback is already evident; "design for manufacturability" has become a major concern in manufacturing industries, reflecting the clear need to explicitly address manufacturing concerns and constraints at the earliest, most abstract levels of conceptual design. The same trend, "design for constructability," can be expected in civil engineering design, as the means of manufacturing, i.e., constructing, facilities change drastically [Warszawski 84].

It is too early to speculate on the specific forms that this feedback from robotic construction to civil engineering may take. What is clear, however, that the feedback will be largely heuristic: certain forms, techniques, materials, etc., will be shown by experience to be more appropriate, efficient, economic, etc., for robotic construction than others. Expert systems are the most appropriate computer aids developed so far for representing and processing such heuristic knowledge. It is to be expected, therefore, that computer-aided civil engineering will increasingly use knowledge-based systems to integrate knowledge about robotic construction into the design process.

5.4. Planning

- *Operation Planning.* Some of the first pieces of robotic construction equipment are being designed to operate in an automated mode in hazardous environments such as mining and excavation. As outlined above, expert systems have a role in the automated interpretation and sensing of the environment of the machine. This environmental information is needed to plan the operational strategies of the machine so it can realize its goal (excavating a pipe, digging a tunnel, bolting a mine roof, etc.). An expert system can augment and eventually replace the operator to provide such control strategies. Such an application has been proposed for the strategic and tactical control of an automated tunnel boring machine (TBM) [Sriram 83a]. *MOVER* also is an example of a prototype operational planning system. Fully autonomous construction robots, such as REX, will require such operational planning expert systems.

- *Construction Planning.* Construction estimating is an example of a task which is computerized at the micro level, but which requires expertise at a macro or strategic level. Once a quantity take-off has been prepared, the computation of unit prices and determination of total estimated cost is an algorithmic task. However, true construction planning, the process of determining what tasks must be undertaken to execute the project and the sequencing and resource requirements of the tasks must be completed before their costs are known. The process of allocating equipment and crews, determining the sequencing of construction tasks and determining the items for the estimate are planning tasks which require expertise. Expert systems could be developed for such applications.

The planning of robotic construction processes and the design of robotic equipment is itself an area for expert system applications. Problem solving strategies such as hierarchical planning and problem reduction, discussed in Section 3.4 are directly applicable to the complex task of planning construction processes involving robotic equipment. A significant component of the planning will be the selection of the robotic equipment itself. Construction tasks to which robotics can be applied exhibit a large variation in the types of sensors, actuators, locomotion and control strategy that may be incorporated [Warszawski 84]. Therefore, both the construction task and the construction robots have to be configured from a variety of components for specific tasks. Since there are very few experts in this area, an extremely useful and practical expert system would be one which could recommend appropriate combinations of robotic components based on a description of the environment, tasks and constraints.

5.5. Monitoring

- *Equipment Monitoring.* Equipment monitors are similar to equipment diagnostic systems described above. However, in this application, the goal is not to determine and rectify the fault in the equipment, but to continually observe the operation of the equipment and determine if it conforms to the planned operation. Such systems can simulate the monitoring functions of an expert operator, who often takes automatic action based on the *sound* or the *feel* of the controls.

- *Construction Process Monitoring.* Construction management continually involves handling contingencies, reallocating resources and forecasting consequences during the construction process. This requires continual monitoring of the state of the construction process and comparison to the planned schedules. Expert systems could be built to model the processes used by experienced construction managers to track and monitor projects.

5.6. Control

A construction robot has to perform three functions:

1. *Sense* both the external environment and its internal state;
2. *Plan* an action based on the sensed state; and
3. *Act* according to the plan.

These three functions are supervised, coordinated and scheduled by an overall *control* functions. Most of the processing or "reasoning" takes place in the functions of sensing, planning and control.

In robotic construction equipment, the complete control system will require the integration of several types of expert systems outlined above. As an example, an automated tunnel boring machine requires an expert system to interpret the sensor readings to infer the tunnel geologic conditions. Diagnosis of machine faults is also made from machine sensor data. An additional expert system is required to monitor the machine performance. This data is then used to drive a planning task to select a mining strategy and detailed machine operation tactics. Such a combined control system is illustrated in Figure 2 [Sriram 83a].

6. Research In Progress

A number of research projects are underway, all aimed at developing prototype expert systems and exploring the application of expert system technology to civil engineering and construction computer applications. Major projects are described below.

6.1. MOVER — Automated Mass Transit System Diagnosis

A prototype equipment diagnostician is *MOVER* [Fenves 83, Fenves 84c], designed to assist in the diagnosis and repair of operational failures of a fully automated mass transit system. *MOVER* takes information on system failures reported to the central control computer and combines these with requested status information to determine the fault and to suggest repair procedures. *MOVER* also is an example of a prototype operational planning system. Once a fault has been determined, it makes suggestions to the operational supervisor to facilitate the timely restoration of service.

The transit system to be diagnosed contains a number of interacting subsystems, both electronic and mechanical. *MOVER* has models for a number of those subsystems. Each subsystem model performs two tasks:

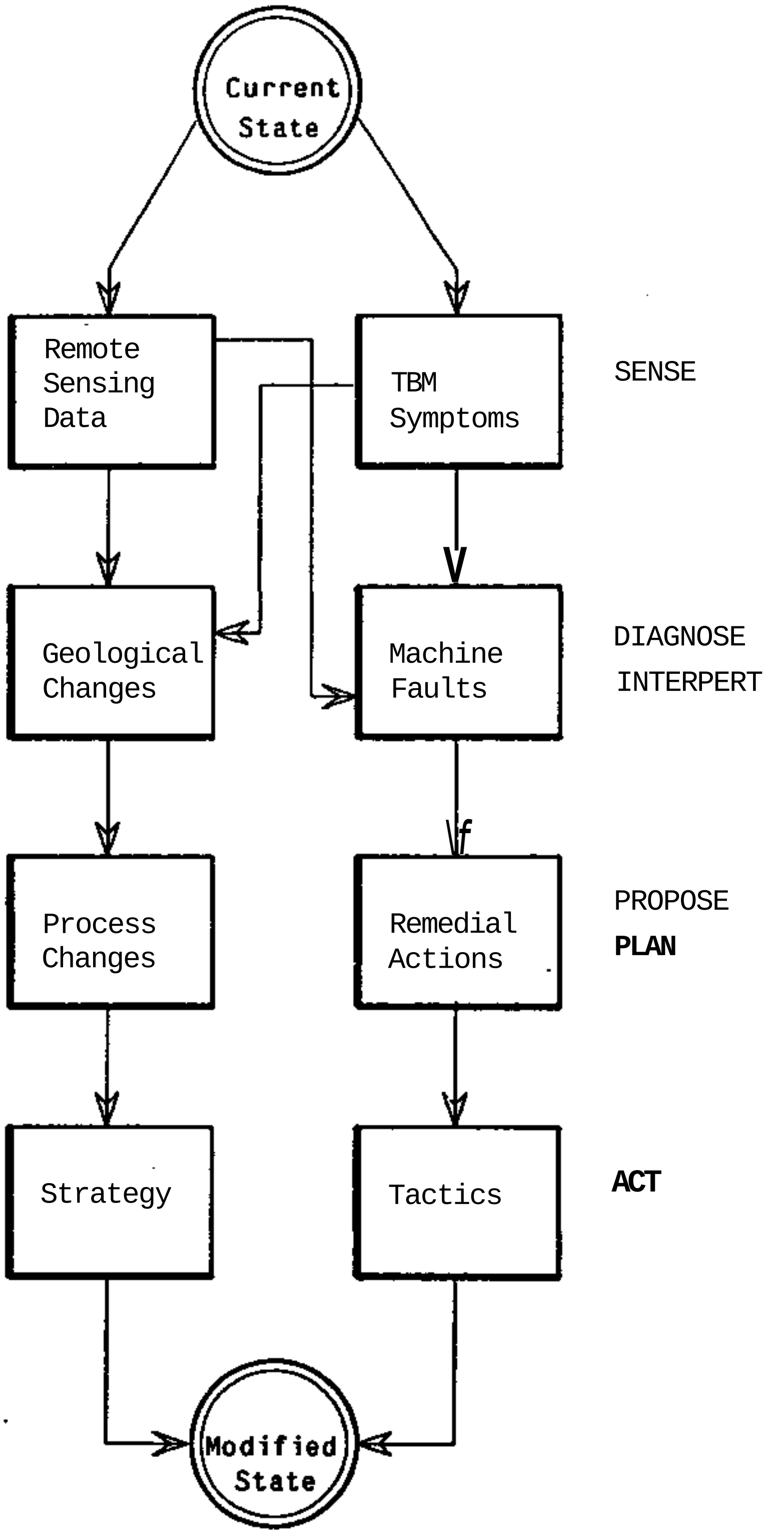


Figure 2: Automated TBM Control Process

- *Fault Diagnosis* — Based on external data regarding the failure and additional requested information, determine the likely causes of the failure

- *Service Restoration Actions Recommendations* — From the fault diagnosis, produce a conditional set of recommendations to restore normal operation of the system.

Two prototype versions *MOVER* have been implemented, both demonstrating a range of capabilities and in-depth diagnosis for selected subsystems.

- The first version was developed in *KAS* [Reboh 81], and contains both deterministic rules and probabilistic inferences. *KAS* was found to have some serious deficiencies for this application. In particular, database techniques are needed to handle historical information on system behavior, and causal representations of intended system behavior (as opposed to purely empirical knowledge about performance and failures) are needed.

- The second version of *MOVER* was developed in *PDS* [Fox 83], an event-driven expert system framework. A conclusion from the development of the first prototype was that efficiency of operation would be enhanced by direct access to the diagnostic data and state information of the subsystems being diagnosed. The *PDS* version addresses this issue by providing direct access to the available sensor information, and making failure inferences based on both the current state and past history of signals.

6.2. CONE — Cone Penetromenter Interpretation

An example of a prototype for sensor interpretation in the geotechnical engineering area is an expert system for the interpretation of cone penetrometer data, *CONE* [Mullarkey 83]. As the cone penetrometer is pushed into the soil, electrical signals provide raw data about tip resistance, friction, and pore pressure. This data is used by experts to infer the soil stratigraphy and more detailed engineering parameters about each soil layer.

CONE performs four interrelated tasks:

- *Data Input and Validity Scans* — Screen errorful data while allowing interactive editing of suspect data, followed by several preprocessing steps based on both theoretically-derived and heuristic corrections.

- *Classification and Stratigraphy Delineation* — Using the corrected data, infer the soil stratigraphy and the soil type of each layer.

- *Design Parameter Inference* — Based on both the classification and data, derive engineering parameters, such as shear strength, for each soil layer.

- *Aggregation and Trending* — Using the engineering parameters for each layer, revise stratigraphy and establish trend lines.

Cone is implemented as a rule based system in *0PS5* [Forgy 81]. For tasks such as classification, several domain knowledge sources are used concurrently to increase the accuracy and flexibility of the system. Fuzzy logic is also used to quantify uncertainty with respect to the quality of the inferences being made, representing heuristic factors that an expert would use in a judgemental fashion. The initial system will provide post-facto analysis as an office-based consultant, but it has been proposed that such a system could be used to build an automated microprocessor-based field penetrometer expert.

6.3. SITECHAR — Geotechnical Site Characterization

Another expert system prototype performing an interpretation task in the geotechnical enginnering domain is begin designed [Norkin 84, Rehak 85], to aid an engineer in inferring subsurface stratigraphy from available bore log data, field and lab test data, and overall site information and characteristics. Due to time limitations, an engineer can explore only a small number of potential inferences. The expert system aids in developing alternative inferences yielding three-dimensional models of site stratigraphy.

Cooperation of several type of problem solving, such as geometric trend recognition, matching soil descriptions, determining proximity, etc., is required to complete the characterization. Each set is represented as a knowledge module in a blackboard model expert system. The system simultaneously operates at two levels of problem solving:

- *Strategic* — At the strategic level an overall approach to characterizing the site is selected. The system poses hypotheses on the overall site stratigraphy, i.e., faults are present, depositional patterns result from oxbow lakes, etc., and several different alternatives are pursued in parallel. Strategic processors also monitor the problem solving process and redirect the system to focus its attention on those alternative which appear most promising.

- *Tactical* — At the tactical level a set of problem solvers attempt to prove hypotheses and fill in the details of the characterization.

All the problem solvers communicate through global data structures on the blackboard which represent the different hypotheses under consideration and the inferred model of the site. The results of the characterization are presented to the engineer through a set of two- and three-dimensional subsurface diagrams.

6.4. HI-RISE— Preliminary Design of Structures

HI-RISE [Maher 84b, Maher 84a] is an expert system which acts as an engineer's assistant for the preliminary structural design of high rise buildings. *HI-RISE* synthesizes feasible structural configurations for a building, performs the preliminary sizing and costing of key structural components, and then presents to the designer the "best" alternative.

Starting with a specified spatial layout, the preliminary structural design process is divided into two ordered subtasks:

- Design of the lateral load resisting system
- Design of the gravity load resisting system

For each subtask, the space of design alternatives is represented by a hierarchy of candidate three dimensional subsystems, two dimensional subsystems, materials and components. The design process for each subtask consists of the following steps:

- *Synthesis* — Search for a set of feasible structural systems.
- *Analysis* — Perform an approximate analysis to insure that the alternative under consideration provides the necessary load-carrying capacity.
- *Parameter Selection* — Select structural components by heuristics.
- *Evaluation* — Compute for each alternative a ranking based on a linear combination of factors such as drift, deflection, number of frames, member size, approximate costs and compatibility of subsystems.
- *Selection* — Select the *best* alternative based on a minimum value of the evaluation function.

HI-RISE is implemented as a rule based system in *PSRL* (Production Schema Representation Language) [Rychener 84], a frame based production system built on *SRL* (Schema Representation Language) [Wright 83].

HI-RISE is the most comprehensive prototype completed to date. Work is ongoing to extend its range of capabilities and accuracy. In addition, it has been used as a testbed for other work, including access to engineering databases and graphical interfaces.

Associated with *HI-RISE* is a smaller expert system-database interface, *HICOST* [Howard 83], used to develop preliminary cost estimates. It is used by *HI-RISE* to evaluate competing preliminary designs based on structure cost. Given the topology and geometry of a building and *HI-RISE's* preliminary choice for the structural system and elements of the superstructure, *HICOST* produces an estimate of the building's cost.

HICOST is implemented as a rule-based expert system, with algorithmic computational functions, which queries a database for specific data on component costs. It is a tightly coupled integration of several expert system tools and a relational database. The *cost estimator* is a hierarchically-organized production system with rules written in *PSRL.* Data objects are defined in frames in *SRL,* and cost data is stored in a relational database supported by *INGRES* [Stonebraker 76]. The production system computes the estimated cost from aggregate subsystem costs. Access to the database is through demons attached to *HI-RISE* cost schemas. Each subsystem data element has an associated cost value. Accessing the cost value of a basic component (beam, column, etc.) via a rule in the cost estimator triggers a demon which invokes a *FRANZ LISP* function [Foderaro 82] which in-turn calls a C procedure [Kernighan 78]. The C procedure uses *EQUEL,* the *INGRES* procedural query language, to access the database and return the item cost to the estimator.

DICE [Barnes 84], a graphical output interface to *HI-RISE* provide two user interface functions. First, it uses a geometric model of the building to accept the user's input and to display the alternative configurations generated by *HI-RISE.* Second, during execution, it displays the context tree of feasible alternatives as these are generated by *HI-RISE.* The completed tree is used for designating alternatives to be displayed and for selecting the "best" alternative.

6.5. DESTINY — Integrated Structural Design

DESTINY (Integrated Structural Design) [Sriram 83b] is an extension of *HI-RISE* to encompass more building configurations, more design tasks and to provide a more flexible design process. The system consists of a number of *knowledge modules* (KMs) (which are comprised of smaller rule sets) which communicate through a blackboard. Its structure is modeled after expert system frameworks such as *HEARSAY-II* [Erman 80] and *HASP/SIAP* [Nii 82]. In *DESTINY,* knowledge is divided into four levels:

- *Strategy KMs* analyze the current design state and determine the next action.
- *Activation KMs* invoke the appropriate specialist KMs to carry out the current design strategy.
- *Specialist KMs* perform one knowledge-based processing task, and are similar to traditional expert systems. Specialist include:
    - o *Preliminary Design System* used to generate potential structural configurations *(ALL-RISE* — an extension of *HI-RISE),*
    - o *Analysis and Modeling Consultant* used as a modeling and result interpretation interface for a finite element analysis program, (similar to *SACON* [Bennett 79]).
    - o *Specification Consultant* used to interface to a set of design standards.
    - o *Estimating Consultant* used to determine cost estimates for alternative designs (similar to *HICOST).*

o *Design Critic* used to evaluate alternative designs based on functional and technical criteria.

- *Resource KMs* provide the knowledge-based and algorithmic processors required for design and analysis such as: finite element analysis, database management systems with associated design and component databases, and standards processors and associated standards.

The overall structure of *DESTINY* is depicted in Figure 3. *DESTINY* is currently under development and is being built from several tools including *PSRL, INGRES,* and *FINITE* [Lopez 77], and is being written in several programming languages including *FRANZ LISP, C* and *FORTRAN.*

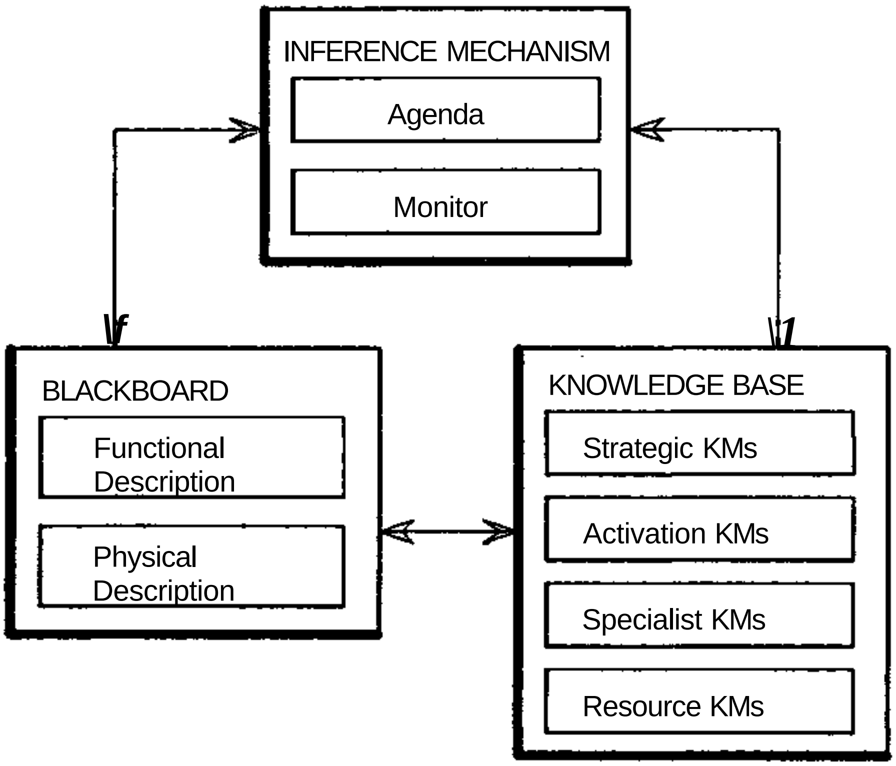


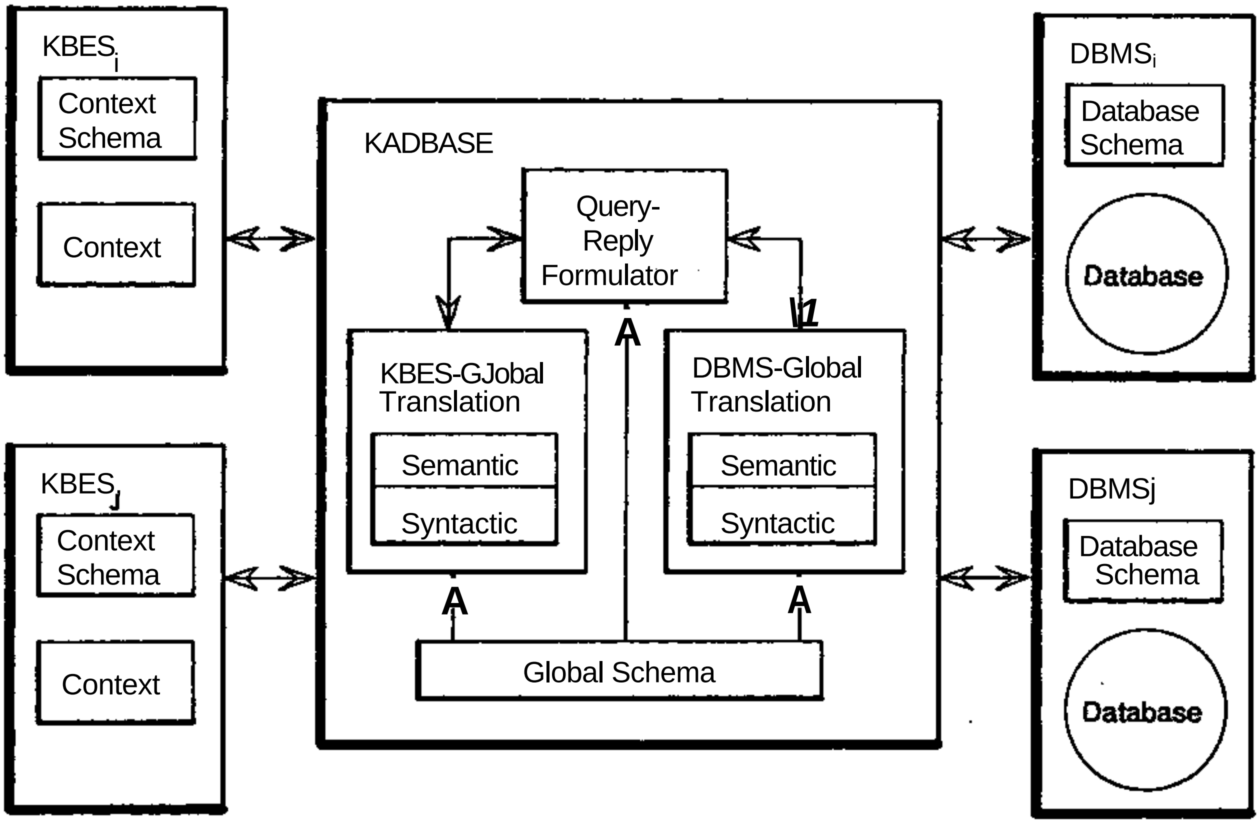
Figure 3: *DESTINY* System Structure

6.6. KADBASE —Interfacing Expert Systems to Design Databases

KADBASE (Knowledge Aided Database Management System) [Howard 84] is a multiple database manager for knowledge-based structural engineering applications. It provides the capabilities for an integrated system to access a collection of databases, and it addresses the issue of interfacing databases with expert systems. Its overall architecture resembles a networked heterogeneous database management system [Cardenas 80]. The individual databases may be based on different data models (relational, network, hierarchical) and may reside on different host computers. However *KADBASE* is not concerned with the issues involved in the physical networking of the databases.

In current applications which combine databases and expert systems, the expert system is tightly coupled to the database in one of two ways: either the expert system has detailed knowledge of the syntax and semantics of the database [Blum 82, Howard 83], or the expert system and database form a combined system [Kellogg 83]. *KADBASE* is an attempt to provide a flexible interface between expert systems and networks of databases by freeing the expert systems fronr, the requirement that they conxain detailed syntactic and semantic database knowledge.

KADBASE provides an interface which allows an expert system to issue queries and updates and receive replies, all in terms of its own information model. In addition, each individual database communicates with the interface in the context of its own database schema. The interface uses mapping information provided by the individual expert systems and database schemas to construct a *global* schema. Queries from an expert system are first translated into the context of the global schema. *KADBASE* determines a strategy for answering the query within the context of the global data model and decides which databases contain the necessary information to answer all or part of the query. Subqueries to the individual databases are translated from the global data model to the syntax and semantics of the design databases. The replies are handled in the reverse fashion. In each translation, *KADBASE* performs a two level, two step language transformation process to communicate between the expert and database management systems. The basic system architecture is shown in Figure 4.



Figu re 4: *KADBASE* System Structure

Two small-scale expert systems prototypes are being developed in association with *KADBASE,* one aimed at showing a breadth of capabilities and the other demonstrating in-depth techniques for database-expert system interaction:

- *HICOST-II* — A revised version of the *HICOST* cost estimator, interfacing with a building design database and an interrelated set of databases containing cost and estimating information.

- *JADE* (Joint Analysis and Design Expert) — A system for designing and detailing structural connections in a building, interfacing with a building design database.

7. Future Directions

7.1. Expert System Development

Work in producing expert systems for civil engineering and construction applications is progressing in parallel in three areas.

- *Frameworks and Basic Capabilities.* Research is underway in developing better expert system frameworks and functional capabilities. Such frameworks provide better knowledge representation schemes, alternative inference techniques, and alternative mechanisms for dealing with uncertain or partial data. Extensions of basic functional capabilities to provide interfaces to algorithmic programs, database management systems, graphical displays, and connections to sensors are also being investigated. Better user interfaces and knowledge acquisition modules will improve the usability of expert systems and will reduce application development cost and time.

- *Applications in New Domains.* In addition to the work in progress described in Section 6, numerous prototype expert system applications are under development or consideration in many civil engineering domains. This work leads to demands for better expert system frameworks and capabilities and results in better techniques for applying expert system technology. Equally important, this work also leads to a more fundamental understanding of the decision making process in the application domains.

- *Knowledge and Process Engineering.* In many application areas, the actual processes used by experts do not appear in the literature, and has not been critically reviewed. Through the development of an expert system, the process and domain knowledge must be formally gathered and represented. The knowledge used is then available for critical examination, which may lead to refinements or improvements in the processes. Equally as important, once captured the knowledge is preserved and can be disseminated.

Continued work will improve the technology and applications of knowledge based expert systems in the civil engineering, construction and construction robotics domains.

7.2. Long-Range Implications

The emergence of expert systems as an additional component of the civil 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.

- *Practice.* The most direct implication in civil engineering practice 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.

- *Education.* 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 [Starfield 83].

- *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.

8. Summary

Knowledge-based expert systems were introduced and presented as an alternative to traditional algorithmic programs in extending computer applications in civil engineering and construction. The overall structure, use, and operation of expert systems was presented. Potential applications in many civil engineering domains have been outlined and research in progress aimed at developing prototypes was described.

By capturing the knowledge used by experts in problem solving, expert systems are capable of addressing problems which, to date, have avoided computerization. There is a wide variety of potential applications in civil engineering and construction. The emergence of construction robots will create the need for many further applications. Additional research and development of expert systems will demonstrate how this powerful new computer based technology can aid in the civil engineering, construction and construction robotics domains.

Acknowledgements: The research reported herein is the product of a number of graduate research assistants, including M. L. Maher, D. Sriram, H. C. Howard, P. Mullarkey and D. D. Norkin. Other Civil Engineering Faculty participating in the work include J. Bielak, C. T. Hendrickson, P. P. Christiano and D. A. Sangrey. M. D. Rychener, M. Fox and other members of the Robotics Institute have participated in various projects.

9. References

[Amarel 78] Amarel, S., "Basic Themes and Problems in Current AI Research,[11] *Proceedings, Fourth Annual AIM Workshop,* Ceilsielske, V.B., Ed., Rutgers University, pp. 28-46, June 1978.

[Balzer80] Balzer, R, Erman, LD. London, P., and Williams, C, "Hearsay-Ill: A Domain Independent Framework for Expert Systems," *Proceedings,.First Annual National Conference on Artificial Intelligence,* pp. 108-110,1980.

[Barnes 84] Barnes, S.J., *DICE: A Design Interface for Civil Engineering,* unpublished Master's Thesis, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, September 1984.

[Bennett 79] Bennett J.S., and Englemore R.S., "SACON: A Knowledge-Based Consultant for Structural Analysis,[11] *Proceedings, Sixth International Joint Conference on Artificial Intelligence,* pp. 47-49, August 1979.

[Blum 82] Blum, R.L., "Induction of Causal Relationships from a Time-Oriented Clinical Database," *Proceedings, Second National Conference on Artificial Intelligence,* pp. 355-357,1982.

[Bobrow 83] Bobrow, D., and Stefik, M., *The Loops Manual,* Xerox Palo Alto Research Center (PARC), Palo Alto, CA, 1983.

[Buchanan 69] Buchanan, B., Sutherland, G., and Feigenbaum, E.A., "Heuristic DENDRAL: A Program for Generating Explanatory Hypothesis in Organic Chemistry," in *Machine Intelligence,* American Elsevier, New York, 1969.

[Cardenas 80] Cardenas. A.F., and Pirahesh, M.H., "Data Base Communications in a Heterogeneous Data Base Management System Network," *Information Systems,* Vol.5, No. 1, pp. 55-79, 1980.

[Clocksin81] Clocksin, W.F., and Mellish, C.S., *Programming in Prolog,* Springer-Verlag, New York, 1981.

[Davis 81] Davis, R., et al., "The Dipmeter Advisor: Interpretation of Geologic Signals," *Proceedings, Seventh International Joint Conference on Artificial Intelligence,* pp. 846-849,1981.

[Doyle 78] Doyle, J., *Truth Maintenance Systems for Problem Solving,* unpublished Master's Thesis, MIT, Cambridge, MA, 1978.

[Duda 79] Duda, R.O., et al., *A Computer-Based Consultant for Mineral Exploration,* Final Report SRI Project 6415, SRI International, 1979.

[Duda83] Duda, R.O., and Shortliffe, E.H., "Expert Systems Research," *Science,* Vol. 220, pp. 261-268, April 1983.

[Dym84] Dym, C.L., *New Approaches to Computer-Aided Engineering,* Technical Report ISL-84-2, Xerox Palo Alto Research Center (PARC), Palo Alto, CA, April 1984.

[Erman 80] Erman, L.D., Hayes-Roth, F., Lesser, V.R., and Reddy, D.R., "The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty," *Computing Surveys,* Association for Computing Machinery (ACM), Vol. 12, No. 2, pp. 213-253, February 1980.

[Fagan 79] Fagan, L.M., et al., "Representation of Dynamic Clinical Knowledge: Measurement Interpretation In the Intensive Care Unit," *Proceedings, Sixth International Joint Conference on Artificial Intelligence,* Tokyo, pp. 260-262,1979.

[Fain 81] Fain, J., et al., *The ROSIE Reference Manual,* Technical Report N1-1647-ARPA, Rand Corporation, Santa Monica, **CA, 1981.**

[Fenves 83] Fenves, S.J., Bielak, J., Rehak, D.R., Rychener, M., Sriram, D., and Maher, M.L., *Feasibility Study of an Expert System for Peoplemover Operation & Maintenance,* Technical Report, Civil Engineering and Construction Robotics Laboratory, Department of Civil Engineering and Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, January 1983.

[Fenves 84a] Fenves, S.J., *Who Will Write Expert Systems: A Historical Perspective,* unpublished working paper, **1984.**

[Fenves84b] Fenves, S.J., and Rehak, D.R., "Role of Expert Systems in Construction Robotics," *Proceedings, Workshop Conference on Robotics in Construction,* Civil Engineering and Construction Robotics Laboratory, Department of Civil Engineering and Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, June 1984.

[Fenves 84c] Fenves, S.J., Maher, M.L., Rehak, D.R., Rychener, M.D., and Sriram, D., *PDS MOVER: Development of a PDS Expert System Prototype for Peoplemover Operation & Maintenance,* Technical Report, Civil Engineering and Construction Robotics Laboratory, Department of Civil Engineering and Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, May 1984.

[Foderaro 82] Foderaro, J.K., and Sklower, K.L., *The FRANZ LISP Manual,* University of California at Berkeley, 1982.

[Forgy 81] Forgy, C.L., *0PS5 User's Manual,* Technical Report CMU-CS-81-135, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, July 1981.

[Fox 83] Fox, M., Lowenfield, S., and Kleinosky, P., "Techniques for Sensor-Based Diagnosis," *Proceedings, Eighth International Joint Conference on Artificial Intelligence,* pp. 158-163, August 1983.

[Greiner 80] Greiner, R., and Lenat, D.B., "A Representation Language Language," *Proceedings, First Annual National Conference on Artificial Intelligence,* pp. 165-168,1980.

[Hayes-Roth 83] Hayes-Roth, F, Waterman, D., and Lenat, D., *Building Expert Systems,* Addison-Wesley, Reading, MA, 1983.

[Howard 83] Howard, H.C., *HICOST,* unpublished Project Report, 12-743 Expert Systems in Civil Engineering, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, December 1983.

[Howard 84] Howard, H.C., *Integrating Knowledge-Based Systems with Database Management Systems for Structural Engineering Applications,* unpublished Ph.D. Thesis Proposal, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, April 1984.

[Intelligences 84] ——————, *KEE: Knowledge Engineering Environment,* IntelliGentics, Inc., 1984.

[Ishizuka81] Ishizuka M., Fu K.S., and Yao J.T.P., *SPERIL I-Computer Based Structural Damage Assessment System,* Technical Report CE-STR-81-36, Department of Civil Engineering, Purdue University, November 1981.

[Kellogg 83] Kellogg, C. H., "Intelligent Assistants for Knowledge and Information Resources Management," *Proceedings, Eighth International Joint Conference on Artificial Intelligence,* 1983.

[Kernighan 78] Kernighan, B.W., and Ritchie, D.M., *The C Programming Language,* Prentice-Hall Inc., Englewood Cliffs, NJ, 1978.

[Lopez 77] Lopez, L.A., "FINITE: An Approach to Structural Mechanics Systems," *International Journal for Numerical Methods in Engineering,* Vol. 11, pp. 851-866, 1977.

[Mackworth 77] Mackworth, A.K., "Consistency in Networks of Relations," *Artificial Intelligence,* Vol.8, pp. 99-118,1977.

[Maher84a] Maher, M.L., *HI-RISE: A Knowledge-Based Expert Systems for the Preliminary Structural Design of High Rise Buildings,* unpublished Ph.D. Dissertation, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, November 1984.

[Maher84b] Maher, M.L., and Fenves, S.J., "HI-RISE: An Expert System for the Preliminary Structural Design of High Rise Buildings," *Knowledge Engineering in Computer-Aided,Design,* IFIP Working Group 5.2 Working Conference, Budapest, Hungary, International Federation for Information Processing (IFIP), September 1984, in press.

[McDermott 80] McDermott, J., *R1: A Rule-Based Configurer of Computer Systems,* Technical Report CMU-CS-80-119, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1980.

[McGartland 83] McGartland, M.R., *Application of Knowledge-Based Expert Systems to Construction Project Monitoring,* unpublished Master's Thesis, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, August 1983.

[Moses 71] Moses, J., "Symbolic Integration: The Stormy Decade," *Communications of the Association for Computing Machinery (CACM),* Association for Computing Machinery (ACM), Vol. 14, No. 8, pp. 548-560, August 1971.

[Motazed 84] Motazed, B., *Expert Imaging of Magnetic Objects,* unpublished Ph.D. Thesis Proposal, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, January 1984.

[Mullarkey83] Mullarkey, P.W., *Development of an Expert System for Interpretation of Geotechnical Characterization Data from Cone Penetrometers,* unpublished Ph.D. Thesis Proposal, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, November 1983.

[Nau 83] Nau, D.S., "Expert Computer Systems," *Computer,* IEEE Computer Society, Vol. 16, pp. 63-85, February 1983.

[Nii 79] Nii, H.P., and Aiello, N., "AGE (Attempt to Generalize): A Knowledge-Based Program for Building Knowledge-Based Programs," *Proceedings, Sixth International Joint Conference on Artificial Intelligence,* pp. 645-655,1979.

[Nil 82] Nii, P., Feigenbaum, E., Anton, J., and Rockmore, A., "Signal-to-Symbol Transformation: HASP/SIAP Case Study," *AI Magazine,* Vol. 3, No. 2, pp. 23-35, Spring 1982.

[Nilsson 80] Nilsson, N.J., *Principles of Artificial Intelligence,* Tioga Publishing Company, Palo Alto, CA, 1980.

[Norkin 84] Norkin, D.D., *Expert System Framework for Geotechnical Site Characterization,* unpublished Master's Thesis, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, 1984, in progress.

[Ray 83] Ray, **M.,** and Swetz, S., *Ground Penetrating Sensor Systems: Robotic Excavator Project,* Technical Report, Civil Engineering and Construction Robotics Laboratory, Department of Civil Engineering and Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, December 1983.

[Reboh81] Reboh, R., *Knowledge Engineering Techniques and Tools in the Prospector Environment,* Technical Report 8172, SRI International, June 1981.

[Reggia 82] Reggia, J.A., and Perricone, B.T., *KMS Manual,* Department of Mathematics, University of Maryland, 1982.

[Rehak83] Rehak, D.R., "Expert Systems in Water Resource Management," *Proceedings, ASCE Speciality Conference on Emerging Computer Technicues in Flood Routing and Stormwater Management,* Niagara on the Lake, Ontario, American Society of Civil Engineers (ASCE), November 1983.

[Rehak84] Rehak, D.R., and Fenves, S.J., "Expert Systems in Construction,[11] *Proceedings, ASME Conference on Computers in Mechanical Engineering,* Vol. 1, Las Vegas, NV, American Society of Mechanical Engineers (ASME), pp. 228-235, August **1984.**

[Rehak85] Rehak, D.R., "An Electronic Workbench for Geotechnical Site Characterization," *Proceedings, Fourth International Symposium on Offshore Mechanics and Arctic Engineering,* Dallas, TX, American Society of Mechanical Engineers (ASME), February 1985, in press.

[Rich 83] Rich, E., *Artificial Intelligence,* McGraw Hill, 1983.

[Rychener **84]** **Rychener, M.D.,** *PSRL An SRL-Based Production System. Reference Manual for PSRL Version 1.2,* Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, **1984.**

[Shortliffe76] **Shortliffe E.H.,** *Computer-Based Medical Consultations: MYCIN,* **American Elsevier, New York, 1976.**

[Simon 81] Simon, H.A., *The Sciences of the Artificial,* Second Edition, MIT Press, Cambridge, **MA, 1981.**

[Smith 83] Smith, R.G., and Baker, J.D., "The Dipmeter Advisor: A Case Study in Commercial **Expert System Development,"** *Proceedings, Seventh International Joint Conference on Artificial Intelligence,* **pp. 122-129, August 1983.**

[Sriram 82] **Sriram, D., Maher, M., Bielak, J., and Fenves, S.,** *Expert Systems for Civil Engineering — A Survey,* **Technical Report R-82-137, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, July 1982.**

[Sriram 83a] **Sriram, D., Mullarkey, P., Bielak, J., and Fenves, S.J.,** *Potentials for Artificial Intelligence and Robotics in Tunnel Boring Machines,* **Technical Report, Civil Engineering** and Construction Robotics Laboratory, Department of Civil Engineering and Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, January 1983.

[Sriram 83b] **Sriram, D.,** *Knowledge-Based Approach to Integrated Structural Design,* **unpublished** Ph.D. Thesis **Proposal, Department of** Civil Engineering, Carnegie-**Mellon University, Pittsburgh, PA, October 1983.**

[Stallman 77] Stallman, R., and Sussman, G.J., "Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis," *Artificial Intelligence,* Vol. 9, pp. 135-196, 1977.

[Stanford 83] Stanford, G., *Potential Applications of Expert Systems in Geotechnical Engineering,* unpublished Master's Thesis, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh. PA, April 1983.

[Startie!d 83] Starfield, A.M., Butala, K.L., England, M.M., and Smith, K.A., "Mastering Engineering Concepts by Building an Expert System," *Engineering Education,* American Society of Engineering Education (ASEE), pp. 104-107, November 1983.

[Stefik77] Stefik, M., and Martin, N., *A Review of Knowledge Based Problem Solving as a Basis for a Genetics Experiment Designing System,* Technical Report STAN-CS-77-596, Department of Computer Science, Stanford University, March 1977.

[Stefik80] Stefik, M_M *Planning With Constraints,* Technical Report STAN-CS-80-784, Department of Computer Science, Stanford University, January 1980.

[Stefik81] Stefik, M., "Planning with Constraints (MOLGEN 1)," *Artificial Intelligence,* Vol. 16, pp.111-140,1981.

[Stonebraker 76] Stonebraker, M., Wong, E., and Kreps, P., "The Design and Implementation of INGRES," *Transactions on Database Systems (TODS),* Association for Computing Machinery (ACM), Vol. 1, No. 3, pp. 189-222, September 1976.

[vanMelle79] van Melle, W., "A Domain Independent Production-Rule System for Consultation Programs," *Proceedings, Sixth International Joint Conference on Artificial Intelligence,* pp. 923-925, August 1979.

[Warszawski 84] Warszawski, A., *Applications of Robotics to Building Construction,* Technical Report, Civil Engineering and Construction Robotics Laboratory, Department of Civil Engineering and Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, May 1984.

[Weiss 79] Weiss, S.M., and Kulikowski, C.A., "EXPERT: A System for Developing Consultation Models," *Proceedings, Sixth International Joint Conference on Artificial Intelligence,* pp. 942-947,1979.

[Winston 81] Winston, P.H., and Horn, B.K.P., *LISP,* Addison-Wesley, Reading, MA, 1981.

[Wright 83] Wright, M., and Fox, M., *SRL: Schema Representation Language,* Technical Report, Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, December **1983.**