KNOWLEDGE-BASED EXPERT SYSTEMS IN STRUCTURAL DESIGN

fay

D. Sriram, M.L. Maher and S.J. Fenves

DRC-12-24-84

December, 1984

KNOWLEDGE-BASED EXPERT SYSTEMS IN STRUCTURAL DESIGN

D. Sriram
Graduate Research Assistant, Department of Civil Engineering,
Carnegie-Mellon University, Pittsburgh, PA 15213

M. L. Maher
Graduate Research Assistant. Department of Civil Engineering,
Carnegie-Mellon University, Pittsburgh, PA 15213

S. J. Fenves
University Professor of Civil Engineering, Department of Civil Engineering,
Carnegie-Mellon University, Pittsburgh, PA 15213

ABSTRACT

A targe number of problems encountered by structural designers are ill-structured and are either not amenable to an algorithmic solution or the algorithms are too cumbersome and restrictive. Knowledge-based expert systems (KBES) provide a means to assist in the solution of these ill-structured problems that demand considerable expertise. KBES also provide a flexible methodology for developing engineering software, as welt as a mechanism for formalizing thought processes about design. In this paper applications of KBES to the preliminary design, analysts, and detailed design phases of structural design are discussed. A conceptual view of an integrated structural design system in the tight of recent advances in KBES development is also presented.

To be presented at the NASA Symposium on Advances and Trends in Structures and Dynamics, October 1984, at Washington, 0. C.

KNOWLEDGE-BASED EXPERT SYSTEMS IN STRUCTURAL DESIGN

D. Sriram[1], M. L. Maher[1] and S. J. Fenves[2]
Department of Civil Engineering
Carnegie-Mellon University
Pittsburgh, PA 15213

INTRODUCTION

The use of computers in structural design to date has been extensive, but limited to algorithmic analysis, proportioning and graphical presentation of results (see [8] for a survey of current CAD trends and applications in structural engineering). However, a number of problems encountered in structural design are not amenable to purely algorithmic solutions. These problems are often ill-structured, and an experienced engineer deals with them using his judgment and experience. Knowledge-based expert systems (KBES) provide a means for using the computer to assist in the solution of these ill-structured problem **areas.**

In this paper we discuss some applications of expert systems in structural design. A brief overview of KBES is provided in the next section. The design process is then briefly presented, followed by three application examples. Finally, a conceptual view of an integrated structural design and analysis system is presented incorporating recent developments in artificial intelligence and expert systems methodology.

OVERVIEW OF KBES

Artificial Intelligence (Al) is the area of computer science concerned with the emulation of human thought processes. Efforts in the application of Al methods to intelligent problem solving led to the development of *expert systems,* systems which perform tasks that require a great deal of specialized

---

[1] **Graduate Research Assistant**

University Professor of Civil Engineering

knowledge that experts in a particular field acquire from long experience with such tasks. Since these systems use domain-specific knowledge they are referred to as *knowledge-based expert systems.* Gaschnig et at. [12] define KBES as "Knowledge-based expert systems are interactive computer programs incorporating judgment experience, rules of thumb, intuition, and other expertise to provide knowledgeable advice about a variety of tasks".

The spectrum of problems encountered in the development of KBES is bounded by *derivation* and *formation* tasks [2]. In derivation problems, the problem conditions are presented as a part of the solution. These problems are normally solved by completing the solution path using available knowledge and data. Some problems encountered at this end of the spectrum are: *interpretation, diagnosis, monitoring,* and *prediction.* Formation problems are posed by the conditions or constraints that the solution as a whole must satisfy. One common way of solving a formation problem is to generate candidate solutions (or partial solutions) and test them against the given conditions or constraints. Examples of such tasks are *planning* and *design.* Many of the problems encountered by engineers fall between these two ends of the spectrum. A detailed discussion of the tasks mentioned above is beyond the scope of this work. The reader is referred to [29] for such a review.

An expert system typically consists of the following components, illustrated in Figure 1 [15]:

- Knowledge-Base. The knowledge-base consists of general facts and heuristic (rules of thumb) knowledge. A number of formalisms, such as production rules, frames (concepts) and semantic nets, are available for representing knowledge. The production rule representation has been extensively used in current KBES. In this approach, knowledge is represented as "IF-THEN" rules or "premise-action" pairs; the "action" is taken if the "premise" evaluates to be true. The knowledge-base may also be partitioned into knowledge levels in order to help organize the problem solving activities.

- Context. The context is a collection of symbols or facts that reflects the current state of the problem at hand. It consists of all the information generated during a particular program execution.

- Inference machine. The inference machine (Inference engine and inference mechanism are other terms commonly used instead of inference machine) monitors the execution of the program by using the knowledge-base to modify the context. Important problem solving strategies used in some existing systems are: means ends analysis, problem reduction, backtracking, least-commitment principle, control using agendas, and hierarchical planning. A detailed description of these strategies can be found in [22, 28].

The components discussed above form the kernel of most existing expert systems. In addition, there are three modules which are desired in any expert system: the user-interface, knowledge-acquisition and explanation modules. A more complete description of knowledge-based expert systems for civil

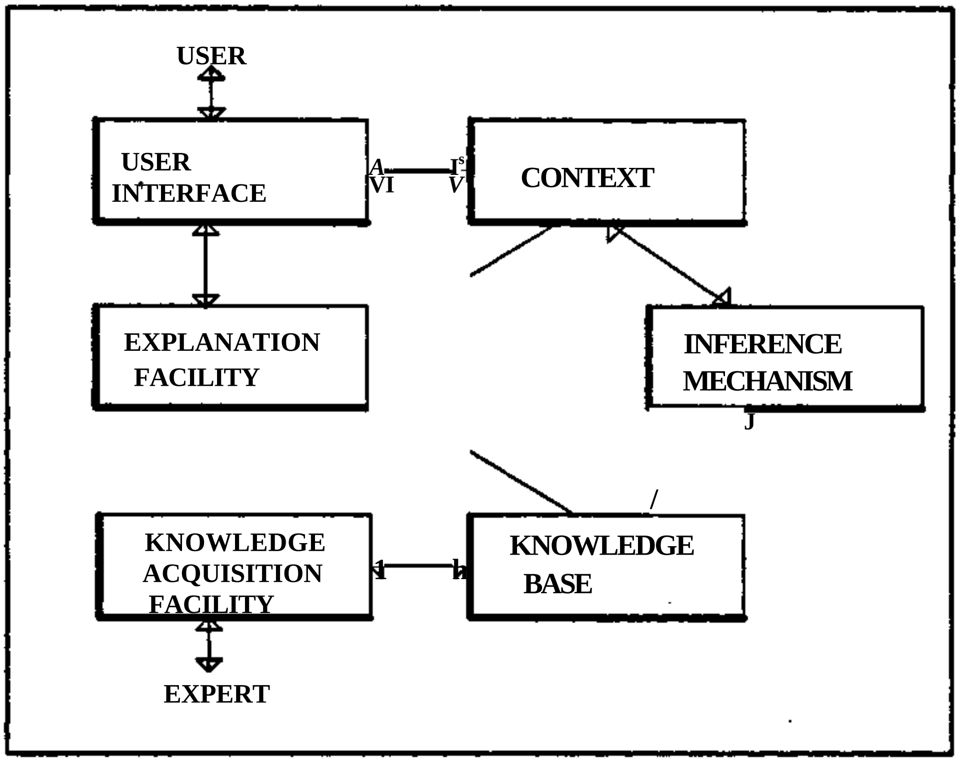

Figure 1: Schematic View of a KBES

engineering is **treated in [26].**

OVERVIEW OF STRUCTURAL DESIGN PROCESS

The structural design process.starts with the definition of a need to transmit loads in space to a support or foundation, subject to constraints on cost, geometry and other criteria. The final product of the design process is the detailed specification of a structural configuration capable of transmitting these loads with the appropriate levels of safety and serviceability. The design process may be viewed as a sequence of three stages:

1. Preliminary design (conceptual design) involves the synthesis of a potential structural configuration satisfying a few key constraints. This stage of design may be further reduced to a series of subproblems. The first subproblem consists of defining the domain of feasible structural configurations applicable to the particular design at hand and choosing one or at most a few of these configurations. The second subproblem involves the formulation of specific constraints applicable to the configuration(s) chosen. This step may require an approximate analysis to estimate the response of the structure. The third subproblem requires a preliminary choice of parameters that define the

configuration(s).

2. Analysis is the process of modeling the selected structural configuration and determining its response to external effects. Analysis may be viewed as the satisfaction of the functional constraints of equilibrium, compatibility and material stress-strain relationships. There are three subproblems in this stage. The structural configuration must first be transformed from a real structure to a mathematical model. This mathematical model is then analyzed. The final part is the interpretation of the results of the analysis in terms of the actual physical structure.

3. Detailed design is the selection and proportioning of the structural components such that all applicable constraints are satisfied. This stage again subdivides into a series of essentially hierarchical subproblems, such as detailing the main structural components (beams, columns, etc.) followed by detailing of their subcomponents (connections, reinforcement, etc.). Within each subproblem, there is a further subdivision into selection based on some controlling constraints (e.g., toad-earring capacity or buckling) followed by the evaluation of secondary constraints (e.g., local buckling or crippling).

There may be significant deviations between the properties of components assumed at the analysis stage and those determined at the detailed design stage, which would necessitate a reanalysis. Other major and minor cycles of redesign may also occur. The process continues until a satisfactory (or optimal) design is obtained. The *conceptualize-ana/yze-detail* cycle is typical of many design paradigms.

The following sections present example applications of expert systems to each of the three stages of structural design.

PRELIMINARY DESIGN

This stage of design is illustrated by a KBES that aids in the preliminary structural design of high rise buildings (HI-RISE) [18]. HI-RISE is implemented in PSRL [24], a frame based production system language being developed at Carnegie-Mellon University. PSRL is a general purpose programming language that provides the constructs available in a production system language and a schema representation language.

The knowledge-base of HI-RISE contains declarative and procedural knowledge. The declarative knowledge is stored in lists and represents a physical hierarchy of known structural types. For example, one physical hierarchy, used for configuring a lateral load resisting system, is given below.

- 3D-subsystems: core, tube

- 2D-subsystems: solid shear wall, braced frame, rigidly connected frame

- material: steel, reinforced concrete, prestressed concrete

The procedural knowledge is represented by production rules and LISP functions. The procedural knowledge is organized into sections or *knowledge modules*. The top level division of the knowledge modules concerns the design of two functional systems: design of the lateral load resisting system, and design of the gravity load resisting system. The design of each functional system is further decomposed into the following knowledge modules.

- Synthesis. A set of alternative configurations are synthesized for the functional system under consideration. The synthesis is performed as a depth first search through the appropriate physical hierarchy in the knowledge-base. The search space is pruned by constraint elimination using heuristic elimination rules.

- Analysis. The major purpose of the analysis module is to evaluate system feasibility. This is done by the formulation and evaluation of one or more feasibility constraints. Analysis provides one ingredient of the constraint, namely the required load capacity of the system components. Parameter selection provides the remaining ingredients required for constraint evaluation.

- Parameter selection. The parameters of the system are initially selected using heuristics. These initial parameters are used to evaluate the feasibility constraints. If a constraint is violated, some heuristic recovery rules are applied to revise the parameters. Once satisfactory parameters are selected, i.e., all applicable constraints are satisfied, the alternative is evaluated.

- Evaluation. Alternative configurations are evaluated with a linear evaluation function. There is a distinct evaluation function for each of the functional systems. The variables in the function are features of the system that may be quantified.

- System selection. All structurally feasible configurations are presented to the user indicating which configuration has been determined to be the "best", selected as the configuration with the minimum value assigned by the evaluation function. The user may either accept the recommended design or override the decision of HI-RISE and choose one of the other structurally feasible systems.

The context tree of HI-RISE represents the solution to the design problem. The tree is decomposed into a functional plane and a physical plane. The functional plane contains two subtrees: a representation of the lateral load resisting system and a representation of the gravity load resisting system. The physical plane contains a hierarchy of the structural subsystems that define the functional systems. The physical plane of the context tree is built up from the physical hierarchy contained in the knowledge-base.

The problem solving strategy in HI-RISE is developed using the inference mechanism provided by

PSRL. PSRL provides several conflict resolution strategies; one strategy is selected for each rule set. The following strategies are used by HI-RISE:

- Fire-cycle. The first rule whose left hand side (LHS) is satisfied is fired, i.e, its right hand side (RHS) is evaluated. The next cycle starts at the beginning of the rule set and fires the first rule whose LHS is satisfied. Cycles continue until there are no rules in the rule-set whose LHS is satisfied.

- Fire-sequence. As in fire-cycle, the first rule with a satisfied LHS is fired. Subsequent cycles begin with the rule immediately following the previously fired rule. Cycles continue until the LHS of the last rule in the rule set is matched with working memory and it is not satisfied.

- Fire-one. Again the first rule that is satisfied is fired. There is only one cycle in this variation of conflict resolution.

Rule sets are used to develop a hierarchical planning strategy, implemented by decomposing the design problem into subprobiems. The subprobiems are solved in a fixed order; first the lateral load resisting system is designed followed by the design of the gravity load resisting system. The design of the functional systems are decomposed into subprobiems corresponding to the knowledge modules and are solved in a fixed **order.**

ANALYSIS

Analysis is the process of determining the response of a fully specified structure to its environment. The response quantities of interest are the stresses and stress resultants (forces and moments) in various regions or components and the deflections of various points. The environment describes the loads acting on the structure, including equivalent loads produced by thermal effects, material changes such as shrinkage and creep, or differential displacements. A fully specified structure is one whose topology, material properties and boundary conditions have all been previously described.

There are literally hundreds of general-purpose structural analysis programs, most of them based on the finite element method, available to the structural designer. Because of the comprehensiveness of the programs, it may take a user a year or more to learn to effectively and efficiently use the common options and capabilities of a large structural analysis program. The total set of capabilities of a large structural analysis program is not likely to be exercised by any single user during his entire professional career.

Knowledge-based expert systems can provide substantial assistance in translating the engineer's knowledge of *what* he wishes to learn from the analysis to *how* to specify this information to an analysis program. Such assistance can substantially elevate the role of an engineer from that of an analyst to that of a modeller, concerned with the choice of the appropriate analytical model and with the interpretation of the analysis results. Furthermore, an expert system knowledge-base can serve as a pool of the accumulated knowledge and modeling expertise of individuals within design organization or even larger professional groupings.

SACON - Structural Analysis CONsultant

An early expert system addressing some aspects of an *automated consultant* to advise non-expert engineers in the use of a general structural analysis program was SACON [4]. SACON was implemented in the EMYCIN system, which is essentially the MYCIN program with the medical diagnosis knowledge removed. This section provides a brief summary of [4].

The structural mechanics knowledge-base of SACON consists of:

- rules for inferring analysis strategies, consisting of the identification of the most appropriate analysis class to be performed and associated analysis recommendations;
- rules for inferring controlling stress, deflection and nonlinear behavior of substructures; and
- mathematical models, in the form of *procedural attachments* for estimating non-dimensional stress and deflection bounds for each substructure, based on its boundary conditions and loading.

The rule base of SACON is encoded as inference rules indicating an action to be performed if the conditions are fulfilled. Premises are conjunctions of clauses, each clause being of the form

```
<predicate function> <object> <attribute> <value>.
```

The effect of the actions is to add or modify a fact in the context, where each fact is represented as a 4-tuple in the form

```
<object> <attribute> <value> <certainity factor>.
```

An example rule for stress behavior is shown below:

```
IF      The material composing the sub-structure is one of:
        the metals, AND
        The analysis error, in percent, that is tolerable
        is between 5 and 30, AND
        The non-dimensional stress of the sub-structure is
```

greater than 0.9. ANO
The number of cycles the loading is to be applied is
between 1000 and 10000
THEN It is definite (1.0) that fatigue is one of the stress
behavior phenomena in the substructure

SACON dynamically constructs a context tree from a fixed hierarchy to represent the relationships between the objects. The variable bindings in a rule are determined by the context in which the rule is fired.

The inference mechanism uses backward chaining that produces a depth-first search of a goal tree. In order to reach a goal, i. e., establish a fact, SACON retrieves all rules that mention that fact in their action and invokes each one in turn, evaluating its premise to see if the conditions specified have been met If any fact needed to establish the premise is not in the context, a subgoa! is set up to establish it and the process recurs. If there are no relevant rules to establish a fact, the user is asked for the value. The generalized subgoai is "determine the value of <attribute>H, rather than "determine whether <attribute> is equal to a specific <value>", thus tending to group related questions together.

Another KBES developed on the same lines as SACON is SesCon [10], a prototype system developed as a front end for the use of the Seasam-69 structural analysis package.

EXTENSIONS AND POSSIBLE IMPLEMENTATION

An extension of a *passive* analyst consultant proposed in [4] would be to integrate the expert system with one or more analysis programs to produce a *dosed loop* system as illustrated in Figure 2. In such a system, the expert system's recommendations (subject to user intervention) would be translated into specific input data, the analysts program executed, and its results returned to the expert system for comparison with its initial predictions and recommendations. If the expert system discovered modeling inconsistencies, it could recommend alternate analysis strategies or model refinements.

An earlier attempt was made by Rivlin et al. [23] to develop such a system. The system, written in FORTRAN, has a finite element KB and has an interface with the analysis program. The system does not have a Muser friendly" natural interface and is very difficult to use. However, recent developments in user-system interfaces [16] and expert system tools, such as EXPERT [31], [32], provide the necessary tools to build the closed loop sysem shown in Figure 2.

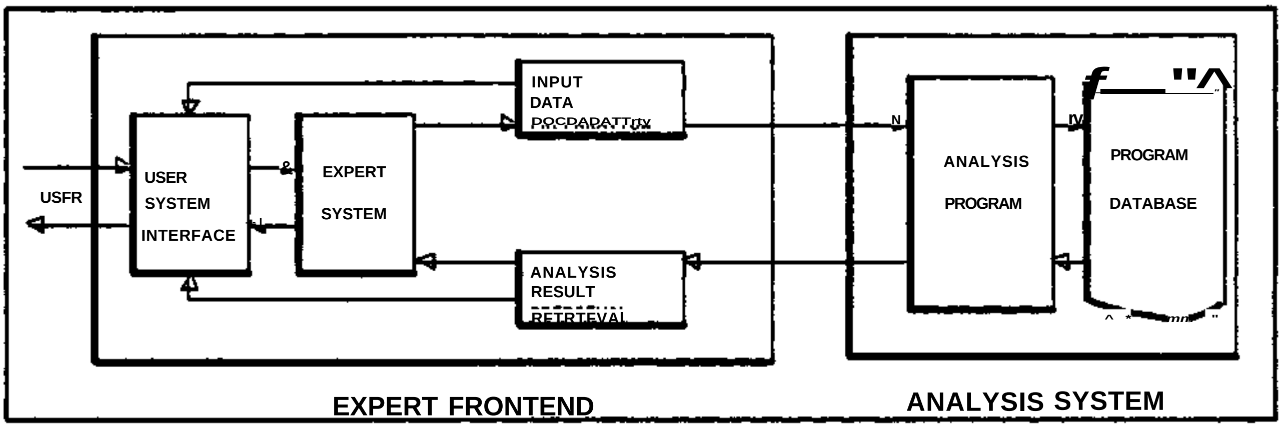

Figure 2: A Conceptual View of an Integrated Analysis System

A complete structural analysis assistant may eventually contain several levels of modeling and model interpretation knowledge. A hierarchy of knowledge sources may be organized as follows:

- <u>modeling strategies</u>: significant modes of behavior (static vs. dynamic, linear vs. nonlinear, etc.) to consider; likely failure modes or limit states (yielding, buckling, etc.) applicable to the structure;
- <u>global idealization</u>: appropriate "granularity" or level of modeling; appropriate subdivisions into substructures;
- <u>local idealization</u>: applicable or appropriate mesh size, element types, loading and boundary condition representations;
- <u>result interpretation</u>: appropriateness of models and idealizations chosen; meaningful aggregations of data to discern patterns and trends; ways of evaluating sensitivity of the model;
- <u>evolutionary strategies</u>: appropriate initial models and idealizations; ways of interpreting results and progressively refining the model;
- <u>efficiency issues</u>: types of models that are efficiently handled by a given program; ways of overcoming program inefficiencies; and
- <u>program details</u>: rules for translating the model into the input language of a particular program.

The hierarchical structuring would result in a more efficient use of the knowledge-base.

DETAILED DESIGN

The detailed design of structural components is governed by many constraints, some mandated by design specifications and building codes, others representing limitations and "design styles" of individual design firms. For example, the two production rules

```
IF      Actual stress f <= allowable stress F
THEN    Stress constraint satisfied
```

and

```
IF      Section is compact, AND
        Not hybrid girder or made of A514 steel
THEN    Allowable stress in bending Fb * 0.66 Fy
```

represent code constraints based in sections 1.5 and 1.5.1.4.1, respectively, of the AISC Steel Design Specification [1]. On the other hand, the decision on whether to use a compact section or not may be a heuristic based in the firm's "design style" of handling specific situations.

The incorporation of specification constraints mandated by codes into computer aided design programs is a two-step process [8]:

1. representing the constraints in a form that can be manipulated by computer programs; and

2. converting passive conformance checking constraints into an active design sequence.

The first problem was addressed by Fenves and others [14,20], by representing provisions in the form of networks of decision tables; the processing of these tables was implemented in a computer program by Stirk [30]. The second issue was addressed in [17]. An alternative or extension of the above is to represent these constraints in the knowledge-base of a KBES (Research in this area is also actively being pursued in U. K. and Australia, see [13] for further details).

SPECON -SEEcification CONsultant

Specifications formulated as decision tables can be recast into expert system representational schemes, such as frames and production rules. The application of expert systems to provision specification is illustrated by using the production rule approach.

SPECON is a small but representative knowledge-based system intended to aid the engineer in checking of structural steel elements for conformance with the AISC Steel Design Specification [1]. It was implemented in LISP [33], a list processing language, and OPS5[11], a production system language.

The organization of the system is similar to that of MYCIN [25] and SACON; it consists of three main modules: the knowledge-base, the context, and the inference machine. The essential difference between SPECON and other expert systems is the flexibility provided to the user to alter numerical values of design parameters until the hypothesis is satisfied, i.e., a component satisfying the constraints is selected.

The knowledge-base is divided into two levels. The first level consists of production rules which identify the applicable constraints. The second level is made up of the specific design constraints, in the same form as discussed in the section on preliminary design. The second level constraints are called from first level rules whenever applicable preconditions are satisfied.

Illustrative rules representing Section 1.5.1.4 of [1] are shown below:

```
First level * set up constraints:

RULE 001
  IF    Section contains unstiffened elements, AND
        Constraint for unstiffened elements satisfied
  THEN  Unstiffened section requirement satisfied


RULE 013
  IF    Constraint for doubly symmetric section satisfied, AND
        Unstiffened section requirement satisfied
  THEN  Allowable bending stress (Fb) is 0.66 times
        the yield stress (Fy)

RULE 029
  IF     Actual bending stress (fb) < allowable bending stress (Fb)
  THEN   Bending stress requirement  satisfied

Second level-constraint evaluation:

RULE 032
  IF     bf/(2*tf) < 65/Sqrt(Fy)
  THEN  Constraint for unstiffened element satisfied

 {bf and tf are the width and thickness of the flange}
```

The context in the LISP implementation consists of four list structures that keep track of the various facts generated in a particular consultation. In the OPS5 implementation, this is reflected in the working memory.

The control strategy implemented in the inference machine is similar to MYCIN's backward chaining strategy. Essentially, it involves starting from the top hypothesis, *Bending stress requirement satisfied,* and checking the facts in the context to see if it can be supported from these facts. If the facts are not available, then the preconditions that are needed for the hypothesis are set up as sub-goals. For example, the condition part of Rule 029 is set up as a sub-goal if it is not found in the context

The explanation module of the system can answer questions, such as

- how a certain hypothesis is deduced, and
- why a particular question was asked.

The user can ask the system any time during the consultation to explain its reason for asking a certain question by typing "why". If the user types "how-did-you-deduce" after a consultation session is over, then the system would explain its reasoning process for any desired **fact.**

EXTENSIONS AND POSSIBLE IMPLEMENTATIONS

A complete detailed design consultant may incorporate the following additional features:

- laroer knowledge-base, incorporating all the mandated constraints of codes and specifications such as [1], possibly including the reason for each constraint;
- soft constraints, such as those representing office practices and "design styles" of individual organizations;
- external constraints, such as those representing good field practices, and relative costs of details;
- meta-rules, such as heuristics for choosing which constraints to apply at which stage of design; and
- design rules f17] to provide bounds on variables such that controlling constraints are satisfied.

The practical implementation of such an expert system must be closely integrated with a database management system which maps the context onto a project file containing the description of the design. For example, the bending stress constraint illustrated in the previous section must eventually be applied to every element in the structure; in order to evaluate the condition "section contains unstiffened elements", the database must be accessed to obtain the composition of each particular section, etc. A mechanism for treating such constraints is presented in [9]. Currently, this topic is an active area of research at Carnegie-Mellon University.

INTEGRATED SYSTEM

In the previous sections, applications of expert systems in the three stages of structural design were presented. In actual design of complex structures these three stages are performed by different experts. For example, architects, space planners, and structural engineers are normally jointly responsible for the preliminary design, while the detailed design is accomplished by teams of structural engineers, detailers, and draftsmen. Furthermore, each of these tasks are divided into a number of subtasks; dividing the design task itself demands considerable experience. Hence the design task requires a good deal of coordination among different experts. Based on these considerations, a *conceptual* view of an integrated design system that utilizes the concept of cooperation of different experts is proposed. The system is intended for a single user, although it provides an excellent paradigm for the development of a multiuser system. The input to the system consists of information provided by space planning, including the number of stories, the plan of the building, and other pertinent spatial constraints.

The overall organization of the system is similar to that of HEARSAY-II [7] and HASP/SIAP [19] systems: it consists of a number of *knowledge-modules* that communicate through a *blackboard*, as shown in Figure 3, and controlled by an *inference mechanism*. These components are discussed below.

The <u>knowledge-base</u> consists of a number of knowledge modules (KMs). Some KMs incorporate both *textbook* and *heuristic* (surface) knowledge, while other KMs comprise *fairly deep* knowledge (Surface knowledge consists mainly of *production rules* encoding empirical associations based on experience, while fairly deep knowledge is comprised of algorithmic procedures). These KMs form a hierarchy of four levels: the *strategy, activation, specialist,* and *resource levels*. The knowledge levels are briefly described below.

- <u>Strategy level KMs</u> analyze the current solution state to determine the course of next action. This level supplies the knowledge required for *focusing attention* on a certain part of the solution. Knowledge from this KM is used by the blackboard monitor to fire the appropriate activator.

- <u>Activation level KMs</u> know when to use the Specialist KMs. The task of KMs at this level is to invoke the appropriate specialists for a given strategy. The system may incorporate both activators available in the HASP system - *the event driver* and *the goal driver*. The event driver invokes appropriate Specialists based on a certain event or group of events. The goal driver acts on those KMs which are responsible for putting a potential design alternative in the blackboard. For example, the event driver would invoke the KM for analyzing the structure if all the required input data is available, while the goal driver would set up the task of analysis as a goal and would try to obtain the input data from the user or other KMs. Essentially, the event and goal drivers are used to control the sequence of execution of the knowledge modules (KMs). This type of knowledge is called *process control knowledge* (see the section on the inference mechanism for a different type of control structure).

- Specialist KMs are used to assess the current state of the design. Some KMs are comprised of pure production rules, while other KMs invoke analytical models of knowledge from the Resource plane. The KMs at this level are comprised of smaller independent modules which are called knowledge sources (KSs). There are two types of KSs: KSs unique to a certain KM and KSs common to several KMs. These KSs are activated by appropriate KMs. Each KM at the Specialist level will be specified in four parts [6]:

  1. conditions under which it is to be activated;

  2. kinds of changes it makes to the blackboard;

  a conditions under which KMs at the Resource level are activated; and

  4. conditions under which KSs are activated.

  Further details on the individual KMs, shown in Figure 3, are presented in [27].

- Resource level KMs contains the knowledge required for analysis and design. Typical KMs at this level are: analyzers, geometric modellers, codes, catalogs of available components, data base management systems, costing procedures, etc.

The blackboard or context consists of two planes: the functional plane and the physical plane. The functional plane is subdivided into two functional subsystems: lateral load resisting system, and gravity load resisting system. The hierarchy of levels in the physical plane was discussed in the section on preliminary design.

The main units in the blackboard are hypotheses or guesses. For example, a possible 3-0 hypotheses such as a *Core* or *Tube-in-tube* may exist in the blackboard. The hypotheses are primary guesses about particular aspects of the problem and may be related through structural links. For example, the gravity load functional level can have many alternative configurations which are linked through an is-aJt (is an alternative) link. The hypothesis structure formed from the most up-to-date information is known as the *current best design* (CBO). Constraints play an important role in forming the CBO. The interaction between various levels through constraints is addressed elsewhere [18].

The inference mechanism consists of two main components: *the agenda,* and *the monitor.* The agenda keeps track of all the events in the blackboard and calculate the priority of execution of tasks that are generated as a result of activation of the KMs. The activation and interaction between various levels is accomplished by the blackboard monitor. In the proposed system, the agenda is implemented as a queue and the priority of execution may be set by the user; the user will have considerable control over the agenda. The monitor performs the following tasks:

- check the blackboard for appropriate KMs to fire; and
- take the activity with the highest priority from the agenda and execute it.

The activation of a certain KM depends on the result of actions of other KMs. For example, the KM for detailed design stage would be activated after the analysis is complete. This can be cast into the following production rule:

```
IF    The analysis of structure is complete, AND
      Detailed design is  required
THEN  Activate the knowledge-module for detailed design
```

Similar production rules can be used to provide an alternate solution to the problem of interaction of multidisciplinary design databases, described by Eastman [5].

Both the data-driven and goal-driven strategies are to be incorporated in the system. For example, if a KM generates a surface element, such as a grid, from linear elements (beams), then the hypothesis about the surface element is data-driven. On the other hand, if the type of surface element, such as the grid, is hypothesized in the light of domain knowledge or suggested by use, then the hypothesis formation is goal driven (Note that in the goal-driven case the hypothesis, i.e., grid, is checked for correctness and consistency from knowledge about linear beam elements which further depend on the material). The data-driven and goal-driven strategies act on the hypothesis structure and are termed *hypothesis-control* strategies.

CONCLUSIONS

A number of applications of expert systems to structural design have been described. A conceptual view of an integrated structural design system using a knowledge-based approach was presented. The objective of these systems is to complement the extensive algorithmic use of computers in structural engineering by augmenting the engineer's problem solving capabilities in areas characterized by heuristic expertise. Prototype implementations of all the applications described above can be developed using existing expert system frameworks such as OPS5, PSRL, KAS [21],

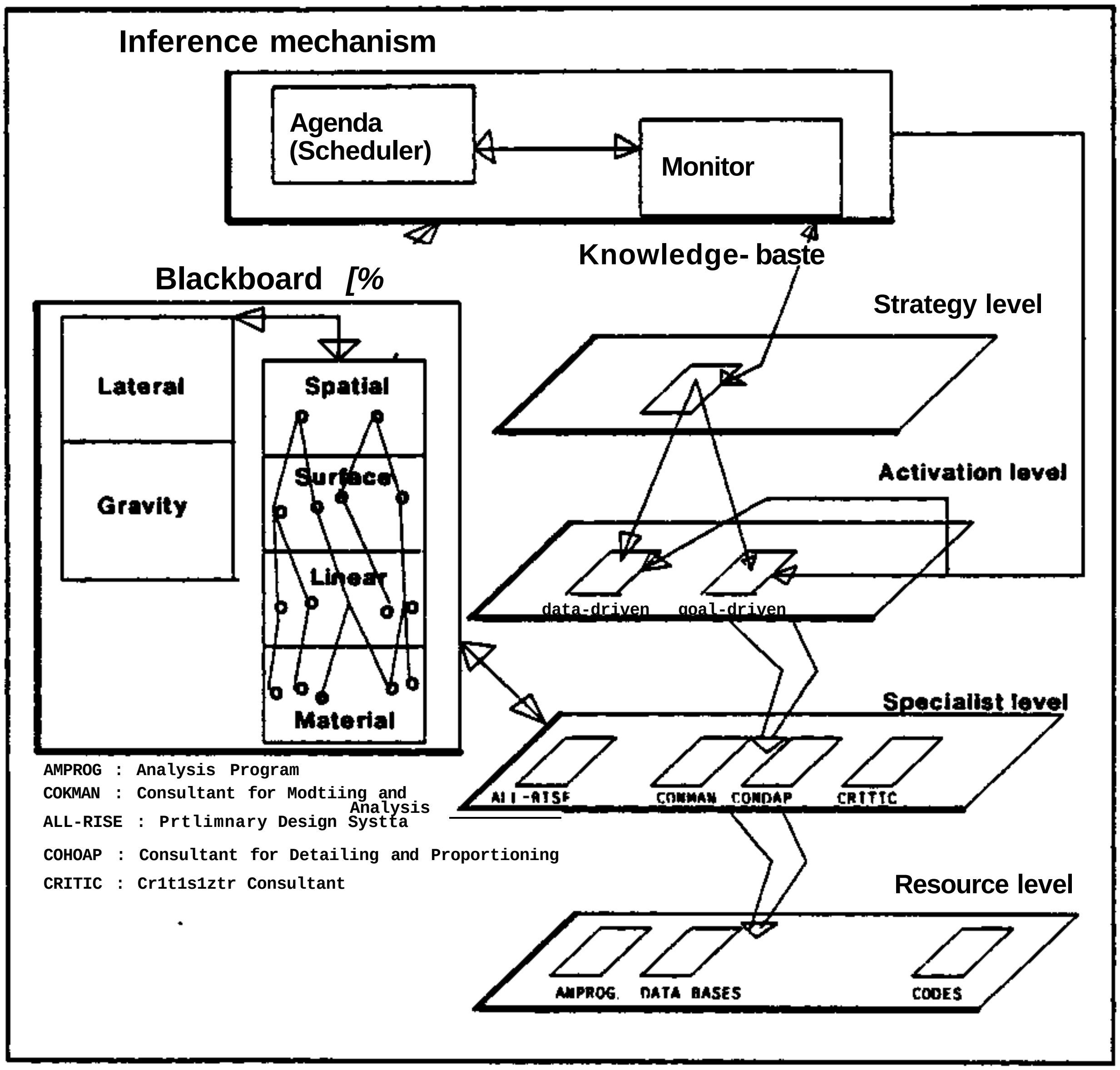

Figure 3: A Conceptual View of an Integrated Structural Analysis and Design System

HEARSAY-HI [3] or EMYCIN. However, their full-scale practical implementation will require higher-level expert system development tools, notably in the areas of user interfaces, explanation capabilities and knowledge-acquisition capabilities. Furthermore, practical implementations will also require close coupling to databases and algorithmic application programs.

ACKNOWLEDGMENTS

The first author is supported by the Ben Franklin Fellowship and the second author is supported by a grant from the National Science Foundation.

BIBLIOGRAPHY

1. ***Specification for the Design*** *Fabrication and Erection of Structural Steel for Buildings,* **American Institute of Steel Construction (AISC), Chicago, 1978.**

2. **Amarel, S., "Basic Themes and Problems in Current AI Research,"** *Proceedings* ***of the Fourth Annual AIM Workshop,*** **Ceilsietske, V. B., ed., Rutgers University, June 1978, pp. 28-46.**

3. **Balzer, R, Erman, L. 0. London, P. and Williams, C, "Hearsay-III: A Domain Independent Framework for Expert Systems,"** *Proceedings of the First Annual National Conference* ***on*** *Artificial Intelligence.,* **1980, pp. 108-110.**

4. **Bennett, J., Creary, L., Englemore, R. and Melosh, R.,** ***SACON: A Knowledge-Based Consultant for Structural Analysis,*** **Technical Report STAN-CS-78-699, Stanford University, September 1978.**

5. **Eastman, C. M., "Database Facilities for Engineering Design,"** ***Proceedings of the IEEE,*** **Vol. 69, No. 10, October 1981, pp. 1249-1263.**

6. **Erman, L. D. and Lesser, V. R.f "A Multi-Level Organization for Problem Solving Using Many, Diverse, Cooperating Sources of Knowledge.,"** *Proceedings Fourth IJCAI,* **1975, pp. 483-490.**

7. **Erman, L. D., Hayes-Roth, F., Lesser, V. R., and Reddy, D. R., "The Hearsay-H Speach Understanding System: Integrating Knowledge to Resolve Uncertainty,"** *Computing* ***Surveys,*** **Vol. 12, No. 2,1980, pp. 213-253.**

8. **Fenves, S. J., "Computer-Aided Design in Civil Engineering,"** *Proceedings of the IEEE,* **Vol. 69, No. 10, October 1981, pp. 1240-1248.**

9. **Fenves, S. J. and Rasdorf, W. J.,** *Treatment of Engineering Design Constraints in a Relational* ***Database,*** **Technical Report DRC-12-14-82, Design Research Center, Carnegie-Mellon University, December 1982.**

10. **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.**

11. **Forgy, C. L.,** ***OPS5 User's Manual,*** **Technical Report CMU-CS-81-135, Carnegie-Mellon University, July 1981.**

12. **Gaschnig, J., Reboh, R. and Reiter, J.,** *Development of a Knowledge-Based System for* ***Water*** *Resources Problems,* **SRI Project 1619, SRI International, August 1981.**

13. **Gero,** J. S., *Knowledge Engineering - Future of Computers in Engineering,* **Presented at the Institution of Engineers - Computers and Engineering ConferenceAustralia, 1983.**

14. **Goel, S. K. and Fenves, S. J., "Computer-aided Processing of Design Specifications,"** ***Journal***

of Structural Division, ASCE, Vol. 97, No. ST1, 1971, pp. 463-479.

15. Hayes-Roth, F., Waterman, D. A., and Lenat, D. B., "Principles of Pattern-Directed Inference Systems," in *Pattern-Directed Inference Systems,* Waterman, D. A., and Hayes-Roth, F., eds., Academic Press, Inc., 1978, pp. 578-601.

16. Hayes, P. J., Ball, E. and Reddy, R., "Breaking the man-machine communication barrier,'[1] *Computer,* Vol. 14, No. 3, March 1981, pp. 19-30.

17. Holtz, N. M., *Symbolic Manipulation of Design Constraints,* Technical Report DRC-02-12-82, Design Research Center, Carnegie-Mellon University, February 1982.

18. Maher, M. L., *HI-RISE: An Expert System For The Preliminary Structural Design Of High Rise Buildings,* Forthcoming Ph. D. thesis, Department of Civil Engineering, Carnegie-Mellon University.

19. Nii, P., Fiegenbaum, E.f Anton, J. and Rockmore, A., "Signal-to-Symbol Transformation: HASP/SIAP Case Study," *The AI Magazine,* Vol. 3, No. 2, Spring 1982, pp. 23-35.

20. Rasdorf, W. and Fenves, S. J., "Design Specification Representation and Analysis," *Proceedings 2nd ASCE Conference on Computing in Civil Engineering,* 1980, pp. 102-111.

21. Reboh, Rene', *Knowledge Engineering Techniques and Tools in the Prospector Environment* Technical Report 8172, SRI International, June 1981.

22. Rich, E., *Artificial Intelligence,* McGraw Hill, 198a

23. Rivlin, J. M., Hsu, M. B., and Marcal, P. V., *Knowledge-based Consultation for Finite Element Analysis,* Technical Report AFWAL-TR-80-3069, Right Dynamics Laboratory(FIBRA), Wright-Patterson Airforce Base, May 1980.

24. Rychener, M. and Fox, M. *PSRL An SRL-Based Production System,* Robotics Institute, Carnegie-Mellon University (forthcoming), 1984.

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

26. 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, July 1982.

27. Sriram, D., *Knowledge-Based Approach to Integrated Structural Design,* Ph. D. thesis proposal, Department of Civil Engineering, Carnegie-Mellon UniversityAugust 1983.

28. 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, Computer Science Department, Stanford University, March 1977.

29. Stefik, M. et. al., "The Architecture of Expert Systems," in *Building Expert Systems,* Hayes-Roth, F., Waterman, D. A., and Lenat, D.. B., ed., Addison-Wesley Publishing Company, Inc., 1983, pp. 89-126.

30) Stirk, J., *Two Software Aids for Design Specification Use,* Master's Thesis, Department of Civil Engineering, Carnegie-Mellon University, October 1981.

31. Weiss, S. M. and Kulikowski, C. A., "EXPERT: A System for Developing Consultation Models," *Proceedings Sixth IJCAI,* 1979, pp. 942-947.

32. Weiss, S. M, Kulikowski, C. A., Apte, C. and UschoW, M, "Building Expert Programs for Controlling Complex Programs,[11] *Proceedings of the Second National Conference on AI,* 1982,

pp. 322-326.

33. Winston, P. H. and Horn, B. K. P., *LISP,* Addison-Wesley Publishing Company, Massachusetts, 1981.