Journal of Wind Engineering and Industrial Aerodynamics, 36 (1990) 1245-1258
Elsevier Science Publishers B.V., Amsterdam — Printed in The Netherlands

DEVELOPMENT OF KNOWLEDGE-BASED SYSTEMS IN WIND ENGINEERING

Ahsan Kareem[1] and Robert H. Allen[2]

ABSTRACT

An overview of basic concepts and techniques necessary to develop expert systems is presented. The knowledge elicitation process and knowledge representation schemes are discussed. We demonstrate that knowledge-based systems can offer intelligent assistance to designers and planners in accomplishing a wide spectrum of tasks requiring wind engineering expertise. We present an example of an expert system under development. Specifically, object-oriented and rule-based representations are used to encode knowledge about the serviceability of high-rise buildings. The resulting prototype expert system assists structural engineers in the preliminary design of building by eliminating those designs that would be unsuitable based on human comfort criteria. We conclude that expert systems can be used to effectively advance the state-of-the-art in wind engineering.

KNOWLEDGE-BASED SYSTEMS

Knowledge-based systems are a part of a new technology stemming from Artificial Intelligence (AI) that have received considerable attention for computer-aided problem-solving (1). AI, as an engineering sub-discipline, seeks to extend the frontiers of sophisticated computer applications to solve critical problems of practical significance. As a scientific field, AI strives to uncover the fundamental principles of intelligent action, whether they be exhibited in humans or machines. The fundamental AI paradigms for problem solving and knowledge representation have contributed to the development of specialized areas such as natural language processing, computer vision and robotics and expert systems.

One definition of a knowledge-based expert system is a computer program designed to mimic human thought processes by utilizing a specific domain of knowledge, facts and procedures to solve complex problems at expert levels of performance. In science and engineering alone, expert systems have been used for such generic tasks as design, diagnosis, interpretation, monitoring, and planning in well over 500 applications (2). In addition to this, the development market of expert systems has burgeoned with both specialized hardware (e.g., Symbolics and Lambda computers) and software (e.g., OPS5, KEE™, and ART™). Recent exploitation of expert systems has been facilitated by the development of software

[1] Director, Structural Aerodynamics & Ocean Systems Modeling Laboratory, Cullen College of Engineering, University of Houston, Texas 77204-4791.

[2] Asst. Prof., Department of Mechanical Engineering, University of Delaware, Newark, DE. 19716.

0167-6105/90/$03.50

"shells" that enable engineers to develop their own expert system without extensive prior computer programming experience. Attempts have also been made in developing expert system shells and production systems for the personal computer (e.g., CLIPS, Personal Consultant™). These developments have facilitated the availability of numerous methods for achieving expert system design and implementation at various levels of sophistication. In reference 3, Allen, et al. attempt to synthesize the theoretical and practical features of the state-of-the-art expert system development.

One main feature distinguishing an expert system from a conventional algorithmic, and typically numerical, program is that the problem solving knowledge pertaining to a specific problem is explicitly encapsulated in a knowledge-base rather than being a part of sequentially executable statements. Simply stated, a conventional program uses problem-solving knowledge in terms of the data and procedures. A procedure is a predetermined sequence of steps (called an algorithm) employing input data to solve the problem at hand. These programs are necessarily restricted to well understood problems that permit repeated use of the same algorithm. A class of relatively less understood and ill-structured problems, e.g., preliminary design, has stimulated the development of expert systems in engineering to facilitate computer-aided solutions of these problems (2, 3, 4, 5).

The principle objective of an expert system is to have knowledge and reasoning encoded, capturing the mechanisms and characteristics of problem solving within a narrow domain of expertise. An expert system draws conclusions from domain specific knowledge that is central to the success of a system. The effectiveness of an expert system lies, to a large extent, in the quality and completeness of the knowledge-base which is encoded in symbolic form. The conclusions are drawn principally through logical or plausible inference rather than mathematical operations. An expert system exhibits cognitive abilities through interaction with the user during a problem solving session, e.g., the ability to provide explanation of line of reasoning in a common-language form, to acquire new knowledge interactively, and search for the optimal solution. This feature is called "transparency" as opposed to the opaque quality of a conventional algorithmic program.

ARCHITECTURE OF KNOWLEDGE-BASED SYSTEMS

In simplistic form, an expert system is comprised of a user interface (for input-output), an inference engine, a knowledge-base and perhaps a knowledge acquisition facility (Fig. 1). A knowledge-based system can draw on multiple sources of knowledge through an inference engine that facilitates reasoning methods and controls the activities in the system. The knowledge-base contains domain specific knowledge that provides the context for solving a specific problem. In the event sizeable algorithmic knowledge is included, the expert system may be classified as a hybrid knowledge-based expert system. The knowledge-base may be categorized into the following modules:

a) control knowledge;
b) expert knowledge;
c) algorithmic knowledge; and
d) static knowledge.

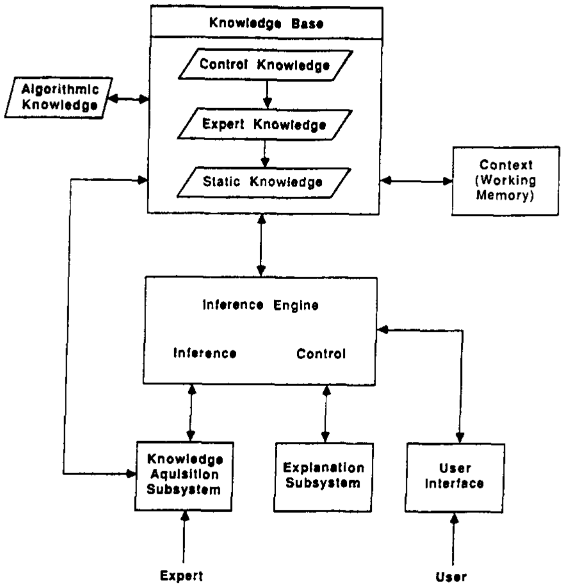

Fig. 1. Architecture of a Hybrid Knowledge Based Expert System

The control knowledge module is the (meta-level) knowledge that is utilized to control and coordinate the activities in the knowledge-base. The expert knowledge contains rules of thumb or heuristic knowledge based on knowledge of the expert in the area. The algorithmic knowledge module contains accepted facts such as those based on the laws of physics and mechanics, e.g., the analysis to estimate the response of a building to dynamic wind loads. The static knowledge is declarative in nature and is provided by handbooks, catalog, design codes and guidelines, e.g., ANSI, uniform and standard building codes (6).

The static knowledge and heuristic knowledge are generally organized in the form of production rules, semantic networks, frames and logic, or other knowledge structures. The concept of associative triple, object-attribute-value (O-A-V) is often used to represent knowledge of describing engineering systems, their components represent knowledge of describing engineering systems, their components and topological layout. A typical format of the O-A-V concept is "<object> <attribute> <value>", which represents the value of an attribute of an artifact, e.g., the "beam" "material" is "steel".

Production rules, or productions, are IF-THEN statements representing causal relations. A rule-based expert system mimics reasoning processes by inferencing from a system of rules. In forward chaining systems, a chain of reasoning is activated in which the system seeks to find a rule whose antecedent (left hand side) matches initial data or previous conclusions (right hand sides). The conclusions are then added to the list of known facts (in working memory) and the system reexamines the rules. This process is continued until a conclusion (goal) is reached. This system is also called bottom-up, antecedent driven, data driven and event driven. It is useful for problem described by an initial state and a goal state (e.g., design, planning). An inference mechanism backward chains when it starts with a goal and works backward to validate the antecedents that imply the goal. If antecedents are unknown, then they become subgoals. Backward chaining is also called hypothesis driven, consequent driven, goal directed or goal driven, and top-down. In some systems a combination of the aforementioned reasoning is included in a problem-solving procedure known as generate-and-test. This system produces possible solutions and then tests them for compliance with the requirements, eliminating those that fail given criteria. Other problem solving paradigms, such as means end analysis and goal reduction, are also described in the literature (1, 3, 5).

Semantic networks are another choice of knowledge representation schemes used in expert systems. A semantic network may be graphically represented by a set of nodes that represent concepts or objects; links that symbolize interrelationships such as a frame being part of a structural system. In computer memory, semantic networks consist of records linked by pointers. Records represent nodes and contain slots for pointers. The pyramid-like structure of semantic networks conserves memory by inheritance of information about objects farther up in the hierarchy.

A frame system represents data structures that consist of procedural and declarative information about a class of objects which is stored in slots. The slots may contain a set of procedures, default values. Like semantic networks, the frame system allows the inheritance of information contained in objects farther up in the hierarchy. One example of a frame representation for a building is given below.

```
{{ Building-1
   Instance: Building
   H:  600 ft.
   W: 100 ft.
   D:  100 ft.
   ρ:  10 lb./cu.ft.
   m:  demon; "mass-calc":
   f:  0.2 hz
   stiffness:  demon: "stiffness:calc":
   Procedure Mass-Calc (H,W,D: Real; var m: real);
   Begin
   m: = H x D x W x ρ/32.2/3;
   End;
   Procedure stiffness-calc (f, m: real; var stiffness: real)
   Begin
   Stiffness:  = (2 x π x f) ** 2 x m
   End;
```

The development of an expert system for a task entails several stages (3, 7). These development stages are summarized below:

i) Identification. This stage involves identification of the task domain and the establishment of goals for the project.

ii) Conceptualization. Following the identification, the next stage entails conceptualization and formalization of the task domain and knowledge representation.
iii) Prototyping. The prototype is made to work on some of the case studies.
iv) User Interface. The user interface constitutes an important component of an interactive expert system. The user interface development is most time consuming.
v) Testing and Refinement. The prototype system in its final stages is exercised against test problems in their entirety. This step helps to identify new problems that may lead to subsequent versions of the prototype.
vi) Knowledge-Base Maintenance. This stage involves testing, development, transfer, extension and maintenance of the knowledge-base.

CHARACTERIZATION OF EXPERT TASKS

Potential knowledge-based systems in wind engineering may be classified as:

1) analysis assistant,
2) design and synthesis,
3) damage assessment and interpretation, and
4) monitoring and control.

Some of the specific applications are identified below.

Wind Loading Assistant

A wind loading assistant may be used to offer advice to users of the provisions of building codes and specifications. In this area, an expert system, WINDLOADER, is being developed utilizing a new version of the Australian Wind Loading code (8). This system will help designers to follow a correct path through the code provisions. The system is designed to take the user through the clauses of the code to select appropriate design wind speed, pressure or force coefficients. The system offers a high percentage of graphical output and user-interaction. The prototype version is developed on the IBM-AT system utilizing ESI Prolog-2 (8).

Risk and Damage Assessment

Systems for the assessment of risk and damage caused by extreme wind would facilitate complex tasks of risk analysis and damage pattern recognition. Generally, the assessment of structural damage is of the heuristic nature involving past experience of investigators together with qualitative information which makes this task an ideal candidate for a knowledge-based diagnosis and interpretation system. Considerable effort in the development of expert systems has been made in the damage assessment and seismic risk assessment of structures, e.g., DAPS, SPERIL and SRAS (9, 10). The systems addressing the wind induced damage and risk assessment are being initiated (11-12). The knowledge base in these systems contains imprecise or uncertain information that requires implementation of inexact or uncertain inferencing schemes (13, 14).

The determination of hazard, vulnerability and significance of the facilities, and analysis of damage potential constitute the essential prerequisites for risk assessment. A synthesis of these attributes leads to the development of a hazard-vulnerability-damage model for risk assessment. The assessment of vulnerability and damage potential of a built environment involves the use of qualitative knowledge which is often vague, uncertain, and imprecise. Reliability-based mathematical modeling of load effects and resistance, and expert knowledge gleaned from the exposure of similar structures to extreme wind events are emerging algorithmic and inferential means of ascertaining damage states of a structure in the form of a damage probability matrix (DPM).

The integration of wind hazard with the facilities at risk is an assessment, which is based heavily on experience. Based on the observations from the past storms, inductive and inexact inference mechanisms can be utilized to encapsulate the body of knowledge concerning damage patterns, often expressed in the form of causal relations or rules, in a knowledge-base. This inference scheme permits assessing the damage potential of a structure at a given level of load by searching through an event tree for associated damage state. The probability of occurrence of any damage state in the DPM can be determined from the probability of occurrence of the basic events. The integration of the statistics of the wind field, as achieved in the first task with the exposed spatial distribution of facilities at risk permits realization of risk in terms of expected damage and attendant loss of lives. An overall hierarchy of events in the simulation of risk is illustrated in Fig. 2.

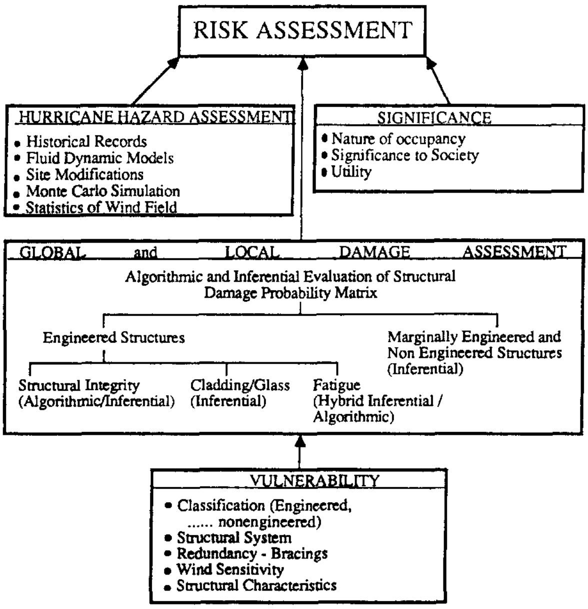
RISK ASSESSMENT
HURRICANE HAZARD ASSESSMENT
• Historical Records
• Fluid Dynamic Models
• Site Modifications
• Monte Carlo Simulation
• Statistics of Wind Field
SIGNIFICANCE
• Nature of occupancy
• Significance to Society
• Utility
GLOBAL and LOCAL DAMAGE ASSESSMENT
Algorithmic and Inferential Evaluation of Structural Damage Probability Matrix
Engineered Structures
Marginally Engineered and Non Engineered Structures (Inferential)
Structural Integrity (Algorithmic/Inferential)
Cladding/Glass (Inferential)
Fatigue (Hybrid Inferential / Algorithmic)
VULNERABILITY
• Classification (Engineered, ... nonengineered)
• Structural System
• Redundancy - Bracings
• Wind Sensitivity
• Structural Characteristics

Fig. 2. Risk Assessment

Design

AI tools can significantly help to improve the design process by better and explicit knowledge representation and definition of design goals and constraints. An expert system may help to configure a structural system and its components to resist wind loads for both the serviceability and survivability limit states on the basis of a set of alternative possibilities. This objective can be accomplished by combining a system like a wind load assistant with a system tailored for the design of structures, e.g., Hi-Rise (5). Efforts in this direction are currently in progress for the development of a hybrid knowledge-based system, WISER, for the preliminary design of high-rise buildings (12, 15). Further details on WISER are given in the ensuing sections.

Monitoring and Control

Wind engineering involves both laboratory and full-scale experiments to examine the effects of wind flow field on the surface pressure and related load effects. In the laboratories, boundary layer wind tunnels with rigid floors or wave basins are used to study the wind structure interactions utilizing scale models of land based and offshore structures. In the full-scale, structures are monitored to collect surface pressures, accelerations and strains at various locations. The primary objective of these field measurement programs is to validate or compare laboratory, computational or analytical predictions of structural behavior in winds. Data monitoring and control expert systems can help to automate wind tunnel and full-scale experiments further by the intelligent process of continuous or intermittent interpretation of signals. The system will observe the experimental progress and alert the user if there is a departure from the expected or usual. At the same time, the system may help to take appropriate actions in response to the monitoring. For example, unusual readings from a pressure transducer may be detected and remedial action may be suggested for the continuation of the experiments. These systems would minimize down-time and repeating experiments in which faulty data is discovered during the analysis. For real-time monitoring problems, it is imperative that the system execute fast enough to perform the assigned task. Further discussion on the performance evaluation of systems may be found in Ref. (3, 16).

Predictive Systems

Predictive systems may be used in wind engineering to forecast expected extreme events on the basis of current information. This forecasting system may rely on a combination of experience, models and procedures. Simulation techniques may be invoked to aid forecasting. For example, the prediction of design winds for a hurricane-prone site requires a hybrid system involving a combination of a knowledge-based and procedural code. During a thunderstorm, prediction of a downburst or wind shear condition derived from doppler radar image may help the weather service to alert airports and local communities of the imminent hazard. This will permit institution of mitigative measures, e.g., diverting flights and relocation of people to secure compounds. Such a system would involve analysis of doppler radar composites to detect incipient conditions for a wind shear.

Urban Planning

The planning of a new urban development can be facilitated and improved by means of an expert system. For example, knowledge regarding the flow around buildings at plaza level, needed to ensure human comfort is generally not available to urban planners. This knowledge, once coded into an expert system, can be more easily reached by architects and urban planers in the early stages of development.

KNOWLEDGE ACQUISITION AND IMPLEMENTATION

The process of knowledge acquisition is a complex one and typicall requires interactive exchange of information between the domain expert(s) and knowledge engineer(s).* The developer(s) lays(lay) out the flowchart of events and reasoning involved to get the task accomplished. Rather than just describe what they do, it is often desirable to ask experts to perform their task. This step is necessary since experts have difficulty in articulating their problem-solving capabilities (7).

Many of the current tools provide assistance in the development of expert systems. Specific tools include intelligent editors and graphical interfaces. These tools can detect syntax or typographical errors prior to compilation, and provide aid in tracing logic and bugs through the use of multiple windows and pointing devices. The effectiveness of the system developers is significantly enhanced when these tools are employed.

A desirable feature of an expert system is hardware independence. The development of a system may be undertaken utilizing a hybrid tool resident on a LISP machine. Subsequently, this system may be ported to a common computer environment for delivery. This will eliminate much recoding of a development system for delivery.

Another important desirable feature is effective linkage of the expert system with the procedural codes. For example, in wind effects applications, a heuristic program will frequently contain algorithmic portions that are best suited for the procedural code. The procedural code may require access to data contained in the rule system's working memory. This linkage is often inefficient and awkward. Some software tools offer a solution to this problem by structuring the working memory in a manner that permits convenient outside access. As a result, the same data base that serves as the rule system's working memory may be used as a data base for procedural codes. Theoretically, this solves the integration difficulty, but not without apparent performance penalty (7). The need may arise to develop intelligent pre- and post-processes for an analysis package based on the finite element method. The interface between the two packages may be obtained through file transfers; although this is not an efficient coupling approach. Alternative options exist in which the FEM codes may be recoded in C. The expert system portion, if coded in LISP, may be compiled into C by utilizing cross-compilers, or it may be initially coded in C (7).

A more detailed overview of the expert system implementation tools and their associated characteristics may be found elsewhere (3, 16).

* A knowledge engineer is a person with training in AI, and is capable of developing an expert3 system.

SOFTWARE FOR KNOWLEDGE-BASED SYSTEMS DEVELOPMENT

One key for the development of a knowledge-based expert system is representation and control of knowledge. There exists an increasing number of tools with which to accomplish this goal (3, 16, 17). Major programming categories include:

- programming languages (LISP, Prolog, C)
- expert system shells and general purpose languages
- programming environment (LOOPS, ART, KEE)
- rule induction tools

LISP is one of the oldest programming languages, and it is most useful for symbolic representation. However, using LISP to develop an expert system requires the most development time because desirable characteristics such as the user interface, inheritance and method of reasoning need to be specifically coded. Prolog is a logic-based language based on first-order predicate calculus. Recently, expert system development tools (e.g., shells) developed in LISP, C, or Prolog offer built-in functions that have significantly simplified the task of building expert systems. These shells offer convenient means for expedient prototyping.

In a production system or rule-based shell, domain knowledge is introduced through production rules, whereas, operational features of the inference engine remain independent of the domain. For different applications, one needs to ensure that the production rules are appropriate for the new domain knowledge representation and the inference engine is suitable for the specific problem.

There are certain limitations as to the application of shells for various tasks. The basic drawback stems from the fact that a single inference mechanism and knowledge representation scheme may not be ideally suited for developing expert systems based on a variety of knowledge domains and tasks (17).

A programming environment combines features of AI, interactive computer graphics and object-oriented programming. This environment assembles basic programming paradigms, such as procedure-oriented, object-oriented, access-oriented and rule-oriented programming. The programming environment offers the flexibility in the representation of knowledge, integration of features and advantage of graphic displays that facilitate debugging, and offers an explanation feature and user interfaces that promote user-friendly systems and rapid prototyping. Further details may be found elsewhere (3, 16-18).

WISER: A KNOWLEDGE-BASED SYSTEM FOR THE DESIGN MODIFICATION OF HIGH-RISE BUILDINGS FOR SERVICEABILITY

Under the action of wind, modern high-rise buildings are measurably more sensitive to motion than heavy weight high-rises of the past. Serviceability becomes an important design criterion as a result of human biodynamic sensitivity to motion as well as awareness of building motion provided by visual cues of moving surroundings (19). The torsional motion may often be perceived by a visual-vestibular mechanism at motion thresholds that are an order of magnitude smaller than those for lateral translatory motion.

There have been numerous full-scale and laboratory experiments conducted to evaluate the human response to motion and thresholds of acceptable motion for several applications. Most of these studies suggest acceptable levels of acceleration in tall buildings. However, the studies differ in details regarding the guidelines for acceptability. These studies also do not agree on whether peak acceleration or rms acceleration should be the governing criterion. For human comfort, the duration and expected number of cycles of motion above a threshold value of acceleration in a steady state motion are more significant than an occasional high peak of transient nature.

The following general criterion for limiting tall building motion is being examined for design application.

> The mean recurrence interval for storms causing an rms acceleration at the building top, which exceeds a pre-established limit, shall not be less than N number of years. The acceleration limit is proposed to be between 8 and 10 mg. and N equal to 10 years.

This follows the format of a similar criterion proposed in reference 20. Currently used design values typically are peak accelerations of 20-25 mg. for a ten year return period. The serviceability checking procedure may be expressed in terms of structural motion and acceptable threshold of motion (21). The structural motion may be obtained using random vibration methods. One of the key components in the checking procedure, the information of the description of wind excited loads in the mechanism that relates an atmospheric turbulent flow field to various wind-induced effects on structures, has not developed sufficiently for an analytical relationship to be formulated. Physical modeling of wind-structure interaction in a wind tunnel continues to serve as the most practical means of relating aerodynamic loading characteristics of the structure to the properties of the local wind climate.

In Fig. 3, a typical structural design cycle for the serviceability limit state is presented. It begins with the preliminary design which is based upon static strength requirements. As a next step, the designer needs to establish the building system's mass, damping and stiffness matrices. This is followed by determining the dynamic characteristics of the building, e.g., natural frequencies and mode shapes. Both these tasks may be accomplished by using algorithmic analysis packages such as ETABS or STRUDL. This constraint makes it essential for an expert system to be interfaced with such a package or have a communication capability through file transfer. Information on the aerodynamic loading may be obtained through a wind tunnel test or a data base for generic building shapes in terms of the building geometry and aspect ratio. The predicted acceleration response is compared to the criterion established for serviceability. In the event a building satisfies this checking criterion, details for the final design are undertaken. Otherwise, the building design is modified and the design evaluation cycle is repeated until satisfactory performance is obtained.

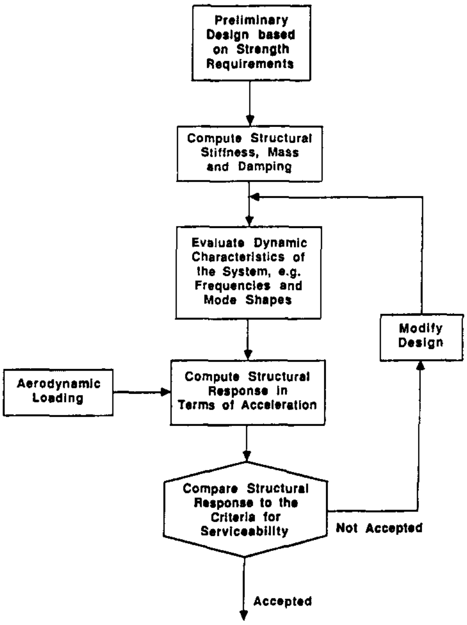
Preliminary Design based on Strength Requirements
Compute Structural Stiffness, Mass and Damping
Evaluate Dynamic Characteristics of the System, e.g. Frequencies and Mode Shapes
Modify Design
Aerodynamic Loading
Compute Structural Response in Terms of Acceleration
Compare Structural Response to the Criteria for Serviceability
Not Accepted
Accepted

Fig. 3. Example of a Design Cycle

Global design modifications presented in Fig. 4 range from considering alternative structural systems to aerodynamic modifications. Further details on the impact of each alternative are discussed in Reference 19. As indicated in Fig. 7, lateral links among various global methods renders the design modification exercise algorithmically unmanageable, consequently in design practice only a limited number of options are explored. An expert system approach holds the promise of offering means of configuring a structural system based on a set of possible alternatives by incorporating most, if not all, constraints that the system must meet.

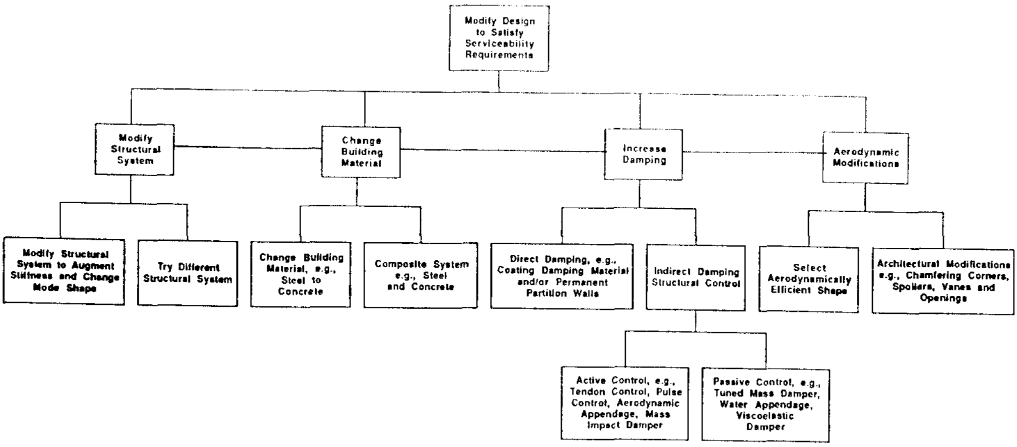
Modify Design to Satisfy Serviceability Requirements
Modify Structural System
Change Building Material
Increase Damping
Aerodynamic Modifications
Modify Structural System to Augment Stiffness and Change Mode Shape
Try Different Structural System
Change Building Material, e.g., Steel to Concrete
Composite System e.g., Steel and Concrete
Direct Damping, e.g., Coating Damping Material and/or Permanent Partition Walls
Indirect Damping Structural Control
Select Aerodynamically Efficient Shape
Architectural Modifications e.g., Chamfering Corners, Spoilers, Vanes and Openings
Active Control, e.g., Tendon Control, Pulse Control, Aerodynamic Appendage, Mass Impact Damper
Passive Control, e.g., Tuned Mass Damper, Water Appendage, Viscoelastic Damper

Fig. 4. Design Modification of High-Rise Buildings

THE WISER MODEL

The WISER model is currently being developed in the commercial hybrid environment KEE, marketed by INTELLICORP. The general purpose AI software tool is designed to allow rapid development of domain specific prototypes. The KEE system is implemented in LISP, and a present work is being implemented on a Symbolics 3640 computer (a specific type of AI workstation). KEE is termed a hybrid environment because it integrates several programming paradigms. The primary knowledge representation mechanism is derived from object-oriented (or frame-based) programming. These objects, or Units as described in KEE, are useful to describe both physical objects and abstract concepts within the domain such as building or stiffness. A sample representation of the Units of WISER, and their underlying relationships (through directed acyclic graphs), is shown in Fig. 6. In KEE, rules refer to the object-attribute-value (or, unit-slot-value in KEE jargon) triples discussed previously, such as "IF the RMS-Acceleration of Building is 8 mg, THEN the Preliminary-Design of the Building is acceptable."

Knowledge is represented in WISER through class-subclass relations. Inheritance, an AI formalism whereby attributes and values are passed from a higher-level object to a lower one, is used to transmit information and keep information consistent. In general, class units are used to describe abstract concepts. The ALTERNATIVE.RULES class and its members are an example of this. Member units typically describe distinct objects and concepts such as STEEL and INCREASE.FREQUENCY.RULES.

The current capabilities of WISER are limited to making design modifications based on a small number of rules and a limited data base of experimental measurements. Continuing efforts should produce a prototype system that can be demonstrated to researchers and practitioners alike in the near future.

CLOSURE

A brief description of expert systems, wind engineering problem domains, implementation tools, software for development of expert systems and practical implementation issues have been provided. We demonstrate that the knowledge-based system can offer intelligent assistance to designers and planners in accomplishing a wide range of tasks requiring wind engineering know-how with limited services of experts in the field. By way of an example of a system under development, we illustrate that preliminary design of high-rise buildings for serviceability may be enhanced by using a knowledge-based expert system. We hope that this paper conveys the rudimentary knowledge for designing and implementing expert systems and would result in motivating interest in this field and subsequent proliferation of KBS in wind engineering.

ACKNOWLEDGMENTS

Financial assistance provided by the National Science Foundation Grant ECE-8352223 and American Institute of Steel Construction is gratefully acknowledged.

REFERENCES

1. P. H. Winston, Artificial Intelligence, Addison-Wesley Publishing Co., Reading, MA, 1984.
2. D. A. Sriram, Bibliography on Knowledge-Based Expert Systems in Engineering, Tech. Report DRC-12-23-84, Carnegie-Mellon University, Design Research Center, Pittsburgh, PA, 1984.
3. R. H. Allen, et al., Using Hybrid Expert System Approach for Engineering Applications, Engineering with Computers, Vol. 2, 1987.
4. D. Sriram, M. L. Maher and S. J. Fenves, Knowledge-Based Expert System for Structural Design, Computer and Structures, January 1985.
5. M. L. Maher, Problem Solving Using Expert System Techniques, Expert Systems in Civil Engineering (C. N. Kostem and M. L. Maher, eds.), American Society of Civil Engineers, N. Y., 1986.
6. C. L. Dym, Expert Systems: New Approach to Computer-aided Engineering, Engineering with Computers, Vol. 1, No. 1, 1985.
7. D. Bobrow, S. Mittal and M. Stefik, Expert Systems: Perils and Promise, Communications of the ACM, September 1986.
8. J. V. Thomson, et al., A Knowledge-Representation Language for Engineering Design Codes, Environment and Planning B: Planning & Design, Vol. 14, 1987.
9. T. J. Ross, et al., DAPS: An Expert System for Damage Assessment of Protective Structures, Expert Systems in Civil Engineering (Ed. E.N. Kostem and M.L. Maher).
10. G.H. Miyasato, et al., Seismic Risk Analysis Systems, Expert Systems in Civil Engineering (Ed. E.N. Kostem and M.L. Maher), ASCE, New York, 1986.
11. D. A. Reed, Risk Assessment for Cladding Panels: An Expert System Approach, Dynamics of Structures (Ed. J. M. Roesset), Proceedings of the sessions at Structures Congress '87, ASCE, 1987.

12. A. Kareem and R. H. Allen, Risk Assessment for Constructed Facilities Under Wind Events: A Knowledge-Based Approach, Research in Preliminary Phases, University of Houston, 1988.
13. J. Fox, Knowledge Decision Making and Uncertainty, AI and Statistics, Addison-Wesley, Reading, Mass., 1986.
14. E. J. Horvitz, et al., A Framework for Comparing Alternative Formalisms for Plausible Reasoning, Uncertainty and Expert Systems: Automated Reasoning Science, 1986.
15. A. Kareem and R. H. Allen, WISER: A Knowledge-Based Expert System for the Design Modification of High-Rise Buildings for Serviceability, Preprints Vol. 2, Proceedings of the Seventh International Conference on Wind Engineering, Aachen, West Germany, July 1987.
16. W. B. Gevarter, The Nature and Evaluation of Commercial Expert System Building Tools, Computer, May 1987.
17. L. Ortolano and C. D. Perman, Software for Expert Systems Development, Journal of Computing in Civil Engineering, Vol. 1, No. 4, October 1987.
18. R. E. Levitt and J. C. Kunz, A Knowledge-Based System for Updating Engineering Project Schedules, Applications of Knowledge-Based Systems to Engineering Analysis and Design (ed. by C. L. Dym), ASME, 1985.
19. A. Kareem, Mitigation of Wind Induced Motion of Tall Buildings, Journal of Wind Engineering and Industrial Aerodynamics, Vol. 11, Nos. 1-3, May 1983.
20. R. J. Hansen, et al., Human Response to Wind-Induced Motion of Buildings, J. of the Struct. Div., ASCE, Vol. 99, No. ST7, July 1973.
21. A. Kareem, Wind Effects on Structures: A Probabilistic Viewpoint, Probabilistic Engineering Mechanics, Vol. 2, No. 4, December 1987.