Applications of Expert Systems in Engineering, Design, and Construction

Kenneth F. Reinschmidt

Gavin A. Finn

Stone & Webster Engineering Corporation
P. O. Box 2325
Boston, Massachusetts 02107 USA

KEYWORDS

Artificial Intelligence, Computers, Construction, Design, Engineering, Expert Systems

ABSTRACT

Expert systems are computer programs based on the experience of human experts, translated into sets of rules for computer evaluation. Expert systems can be used in engineering, design, and construction for diagnosis of problems, technical evaluation, code checking, planning, and management assistance. Expert systems have been developed at Stone & Webster Engineering Corporation (SWEC) to run on minicomputers (DEC VAX) and microcomputers (IBM PCs) commonly available in design offices and on construction sites. Some examples of the systems in use at SWEC include vibration analysis, mechanical equipment diagnosis, welding diagnosis, and system planning. Experience with these programs has shown that expert systems can provide a useful method for distributing specialized technical expertise to construction jobsites and other locations when and where human experts are not available. Such expert systems could also be useful in transfering technology to developing countries.

APPLICATIONS DES SYSTEMES EXPERTS DANS L'INGENIERIE LA CONCEPTION ET LA CONSTRUCTION

Kenneth F. Reinschmidt

Gavin A. Finn

Stone & Webster Engineering Corporation
P. O. Box 2325
Boston, Massachusetts 02107 USA

Mots Clef

Intelligence artificielle, ordinateurs, construction, conception, ingenierie, systemes experts.

Sommoire

Les systemes experts sont des systemes informatique utilisant de expertise humains formalisee dans une ensemble de regles, capable d'etré evalue par le systeme. Utilises dans l'ingenierie et le construction pour le control des codes, l'analysé des systemes technologiques, d'aid á la decision. Les systemes experts ont etés développés chez Stone & Webster pour fonctionner sur mini-ordinateurs (DEC VAX) et micro-ordinateurs (IBM-PCs) courament disponibles dans les bureau de sessin et dans les chantiers. Quelques examples des systemes utilises chez Stone & Webster comprennent l'analyse des vibrations, le diagnostique d'equipements mechaniques, le diagnostique des soudures et l'analysé des systemes. L'experience avec ces systemes a montre qu'ils peuvent pourvoir une methode utile pour assurer l'expertise technique specialiseé aux chantiers et autres sites, quand les experts humains ne sont pas disponibles. Les systemes experts peuvent aussi etre utiles dans le transfert de technologie aux pays en voie de développement.

INTRODUCTION

The term "artificial intelligence" connotes computer programs that produce results that are usually thought to reflect human intelligence. Artificial intelligence as an academic discipline emerged in the mid-1950s as a study of human problem-solving and cognition. Some subfields of artificial intelligence include understanding and translation of natural languages, machine vision, robotics, and expert systems.

An expert system is a type of artificial intelligence application that attempts to replicate the performance of some human expert on some specialized reasoning task. An expert system, more accurately called a knowledge-based system, generally consists of a body of knowledge and a logical mechanism for interpreting this knowledge.

In the most common type of expert system, knowledge is expressed and stored in the form of rules. Some of these rules may be exact scientific formulas, but most are derived from the experience of the domain experts who build the system. These rules are expressed in the form of logical If-Then-Else statements:

```
IF Condition A is True
 AND Condition B is True
 AND Condition C is True,
THEN Condition X is True (or, take Action X)
ELSE Condition Y is True (or, take Action Y)
```

The feature that distinguishes rule-based systems from conventional computer programs is that the knowledge base is kept independent of the logical mechanism (called the inference engine) that operates on the rules to produce logical conclusions. This separation has two significant results: the inference mechanism does not have to be rewritten for each application, and the rules may be changed without affecting the inference engine. Separation thus leads to increased efficiency: computer experts can concentrate on producing better inference engines and domain experts can concentrate on building the knowledge bases without being concerned about programming issues.

Current inference engines are largely based on two fundamental approaches, both conceptually simple: forward-chaining and backward chaining. In forward-chaining the inference engine matches its list of conditions against its list of rules to determine if the values for all the conditions (A, B, and C in the example above) for any rule are known to be True. If so, the rule is said to fire, and its conclusions (X in the example above) are set to True. These conclusions then become known conditions that can be used as the inputs to other rules, possibly causing them to fire. These results can then cascade through a chain of rules until the solution of the problem is reached.

In backward-chaining the inference engine selects the first hypothesis from the list of possible solutions and backtracks to determine the rules that must fire in order that this hypothesis can be proved to be true. It then backtracks through each of these rules, and their predecessors, until it arrives at a fact with no antecedent; at this point it asks the user for the value (True or False) of this fact. If the response supports the working hypothesis, the process continues until the hypothesis has been proved; if any response does not support the working hypothesis it is rejected and the inference engine begins again with the next possible solution on the list.

In academic research, expert systems have usually been written in a computer language called LISP (for LISt Processing language), developed in the early 1960s, and now commonly run on specially designed computers called LISP Machines. European and Japanese researchers have favored a language called PROLOG (for PROgramming in LOGic), which has lately been increasing in popularity in the US. However, few engineers outside the academic artificial intelligence community are proficient in these languages, and expert system development programs (shells) have recently become commercially available.

DEVELOPMENT OF EXPERT SYSTEMS

Engineering, design, and construction organizations contain large numbers of specialized experts whose knowledge could be usefully incorporated into computer systems for problem-solving. Engineering and construction work, particularly at remote locations, frequently require specialized technical knowledge on short notice. It is often impossible or uneconomical to maintain experts on site for such contingencies, but expert systems can be available 24 hours a day, 7 days a week. Rule-based systems can also be used to accumulate the combined knowledge of a number of experts, and can, in principle, be continually improved.

The choice of computer depends on whether the focus of the effort is on development or delivery. Most universities and artificial intelligence companies emphasize development; if so, the obvious choice for a computer is a LISP Machine. This choice may necessitate an additional step: in order to deliver the expert system to users, the developmental system may have to be translated to a program that will run on the computers that the users have available. At the Stone & Webster Engineering Corporation (SWEC), the strategy emphasizes the delivery of expert systems to the ultimate users and the development of expert systems on the same computer that will be used to execute them. Because of its wide distribution in design offices and on construction sites, the computer most often used at SWEC for the delivery of expert systems is the IBM PC.

The decision to use a microcomputer places a number of constraints on the size of the expert system. In order to achieve its objectives, SWEC has written its own expert system development shell, the Microcomputer Artificial Intelligence Diagnostic System. This expert system shell is composed of two modules: a Compiler and an Executor. The Compiler assembles and condenses the rule base and the Executor carries out the inference process (forward-chaining) at run time.

In this system, the developer creates rules in an English-like rule format using any text editor. Text strings can be almost unlimited in length, so that the developer can clearly explain the intent and meaning of all questions addressed to the user, rather than using cryptic computer jargon. Tutorials may also be written by the developer; these are available to the user on demand at appropriate points in the inference process to explain any technical points or terminology that may be unfamiliar.

Based on SWEC's experience with its own inference engine and with commercial development programs, several important factors in the selection of an inference engine are:

1. The developmental system should be usable by a domain expert with no programming background.
2. The developer should be able to add ample text to explain to the user what the expert system is doing and why.
3. Rules should be compiled or condensed to protect any proprietary information in the knowledge base.
4. The developmental system should provide an interactive interpreter, to facilitate prototyping.
5. The developmental system should provide debugging aids and documentation of the rule base.
6. The inference engine should be able to back up, restart, and automatically document a user session.
7. The inference engine should provide links to applications programs written in common computer languages (FORTRAN, Basic, Pascal, etc.).
8. The program should have a separate run-time module, with an unlimited royalty-free license to copy. This is essential if one wishes to distribute the result to users.

Even with improved expert system shells, human experts may not feel comfortable dealing directly with computers if they have had no previous computer experience. Mediating between the domain expert and the computer is often a person called a knowledge engineer, who is familiar with the operation of the computer and the development of expert systems. The functions of the knowledge engineer are to ask questions of the expert and to formulate the responses into rules. Having devised a rule base, the knowledge engineer then debugs the expert system by inputting test cases and checking the system's responses against those of the human expert. This debugging requires a number of trial-and-error iterations, modifying rules and adding new rules, until the performance of the system is satisfactory to the expert. Although generally computer scientists perform the role of knowledge engineers, SWEC has found that engineers familiar with the subject matter are at least equally suitable.

The goal of the developmental effort at SWEC is to provide the facilities for the domain experts to generate their own expert systems, with only minimal assistance from knowledge engineers. As an engineering-construction company, SWEC has a great number of expert engineers and specialists, compared to the number of available knowledge engineers. With the proper tools, there is no reason why engineers and construction personnel should not be able to develop expert systems just as they have developed computer applications programs in the past.

In spite of the fact that the field of artificial intelligence is concerned with human cognition, very little attention has been paid to how people generate rules. Virtually all research has been concerned with the methods for processing the rules after the rules have been generated, and almost none with the methods by which rules are to be derived in the first place. To meet this deficiency, SWEC has written its own Guideline for the Development of Expert Systems. This publication, for the benefit of potential domain experts, explains the fundamentals of expert systems and the most effective ways to generate and test rules. With this document, domain experts are able to develop expert systems in their own fields, with some consulting help from knowledge engineers.

For example, one of the points stressed in the Guideline is that the rule base should be prototyped. That is, instead of attempting to write down all the knowledge that he possesses, the expert should think of one possible solution to the given problem and one rule that would lead to this conclusion. This single rule constitutes a simple, but complete, prototype knowledge base. The expert should then add new rules interactively, observing the effects of each, and modifying the previous rules as necessary.

When the prototype is stable, it should be tested and evaluated by other experts in the same discipline who have had no direct responsibility in the generation of the initial rule base. The prototype expert system should then be tested in the field by potential users, who are not necessarily experts. The comments and criticisms of these users may generate further refinements of the rules and may uncover problems not foreseen by the specialists. It may not be feasible for the expert system to solve every problem, but it should fail safely: whenever it cannot find a correct solution it should so notify the user. It may prove to be desirable to discard the prototype after its purpose has been served, and to develop a more efficient system based on the experience gained with the prototype.

APPLICATIONS OF EXPERT SYSTEMS

Several expert systems are under development in the engineering and construction field. For example, the Center for Building Technology at the National Bureau of Standards is currently developing expert systems to evaluate concrete durability according to the ACI Code and to analyze air infiltration in buildings and to suggest remedies to prevent it. Stanford University is developing expert systems in construction safety and the U. S. Army Construction Engineering Research Laboratory has a number of construction engineering systems under development. A few of the expert systems developed and in use at Stone & Webster are described below.

Pump Diagnostics. A microcomputer expert system called Pump-Pro has been developed to diagnose difficulties with startup or operation of centrifugal pumps. The target user group includes mechanics, millwrights, startup engineers, and maintenance foremen. The system first asks the user to identify one or more symptoms from a set of common pump behavioral characteristics, such as vibration, low flow, low pressure, etc. The Pump-Pro system then relates the given symptoms to one or more of a set of common causes of pump problems and suggests one or more remedies for the problem. If the user does not understand some of the technical terms used by the program, he may ask for tutorials explaining these concepts. The system was developed using SWEC's own inference engine and contains more than 460 rules. Over 400 copies of the program have been distributed to users and the original system has been enhanced and tailored to meet the needs of specific clients.

Vibration Diagnostics. This expert system analyzes the causes of vibrations in rotating machinery. Although a diagnostic system like Pump-Pro, it operates at a higher level of expertise, in that it requires as input the results of a spectral analysis of the vibration. It uses the measured frequencies and amplitudes of the vibration, plus other information provided by the user in response to the system's questions, to infer the probable causes of the vibration and to suggest remedies to eliminate it. This expert system was developed simultaneously for both the IBM PC and the VAX 11/780 computer.

Welding Diagnostics. This expert system diagnoses problems such as porosity, undercut, cracking, etc., that may arise in field or shop welding due to improper welding technique or environmental conditions. It is intended to be used by welders, welding supervisors, and welding inspectors in the field. As in the diagnostic systems described above, the expert system asks the user questions concerning observed welding defects that will lead to the identification of the causes of the problem and recommendations for its resolution.

Other expert systems in use in the welding area assist welding supervisors in the selection of the proper welding procedures to weld any two metals and in the selection of welders with the proper qualifications to make a given weld.

System Dispatch. Generating stations in an electric power system are dispatched, or put on line, in an economic order determined by their marginal costs of production. Algorithms exist to find the optimal dispatch order, but these optimization algorithms may not consider certain practical operational factors that are known to the plane operators but cannot be expressed mathematically. SWEC has developed an expert system which combines the optimization algorithms with a set of rules that express the practical constraints on plant operations. The result is a solution that is optimal while satisfying the given heuristic rules.

Dial-up Expert Service. In order to provide all users with access to the most up-to-date versions of these expert systems, as well as others, SWEC has developed a dial-up expert service, called EXSTRA (EXpert System Remote Access) Service. Any user with an IBM PC-compatible computer, a telephone modem, and an EXSTRA Service access diskette can dial up a central computer system on which the latest versions of these expert systems are resident. The user can then execute the expert system on the central computer just as if it were on his own machine. By this approach, the user also has access to larger expert systems that cannot be conveniently distributed on diskettes. SWEC human experts are available on call to back up the expert system if the problem exceeds the program's capabilities.

CONCLUSIONS

SWEC's experience in the development and use of expert systems indicates that these programs can be useful in the delivery of technical expertise to construction sites and other locations when and where human experts may not be available. Microcomputers are ubiquitious on construction sites and in design offices, and microcomputer-based expert systems, although less powerful than systems based on specialized LISP machines, are feasible now.

Experience shows that microcomputer expert systems can be developed by engineers and technical specialists with the assistance and guidance of knowledge engineers. Expert system consultations are also available over telephone connections to remote users with microcomputer terminals and dial-up facilities. Practical microcomputer-based expert systems have been developed by SWEC, have been distributed in hundreds of copies, and are in use now by SWEC and client personnel at headquarters offices and construction sites.