Expert Systems as a Part of Pavement Management

PATRICK R. FLANAGAN AND DANIEL S. HALBACH

Expert systems have recently excited a great deal of interest in all areas of engineering. The availability of affordable software running on mini- and microcomputers has allowed this type of decision tool to move from academe to practical use. Expert systems are discussed in this paper in relation to their place in pavement management systems. The structures of these systems are defined and compared with current pavement management systems. The present state of expert systems is then reviewed. Areas in which pavement management systems can be enhanced are examined, as are the current limitations of these systems.

With the rapid increase in capability and decrease in price of mini- and microcomputers in recent years, fields once thought of as solely the province of university computer science departments have become less esoteric. A striking example is artificial intelligence, which has as a goal making computers act intelligently. Within the field of artificial intelligence, there are several related areas of study including: robotics, machine vision, machine translation, speech synthesis, game theory, and expert systems. Study and development of these areas has expanded from academe to business, resulting in rapid advances. Of all the branches of artificial intelligence, expert systems have produced a great deal of excitement and some of the most concrete results. Because of success with expert systems in other fields, interest is developing in incorporating expert systems' concepts into pavement management systems (PMSs).

The purpose of this paper is to briefly describe the history and development of expert systems, define their current state, examine some long-term research goals, and investigate the usefulness of expert systems' applications to PMSs.

HISTORY

Expert systems research was begun in the late 1950s as an attempt to automate the thought processes of scientists *(1, 2)*. Expert systems were originally built from scratch for each application usually using LISP, the most common programming language for artificial intelligence. Like most computer programs, these early systems mixed rules and data for making decisions with the problem solving process. Such an approach has several drawbacks, which are present in current PMSs and will be discussed later in this paper.

The early expert systems were followed by a landmark program called MYCIN which is still in use. MYCIN was developed by Feigenbaum and Shortliffe to assist doctors in diagnosing bacteriological diseases. MYCIN represented two major advances in the development of expert systems: *(a)* it was the first expert system able to explain why decisions were made, and *(b)* it was the first to separate the data and rules for the decision-making process from the process itself *(2)*. Both of the characteristics are important reasons for incorporating expert system techniques into PMSs.

As expert systems evolved, it became apparent that the process by which decisions were made was somewhat independent of the type of expert system. Although the type of data dictates to some degree how they are manipulated, researchers found that processes for evaluating rules could be used with other data sets for different expert systems. As a result of these observations, the data and rule set were stripped from MYCIN to form EMYCIN, which was labeled an expert system "shell" *(2–4)*. A shell can then be used to create a different expert system without the time and expense required to create a completely new inference engine.

Even though development of the concept of shells was an important evolutionary step, expert systems were still confined to mainframe applications, and were therefore beyond the reach of everyone except universities and very large corporations. A great deal of effort has been expended to adapt expert systems to mini- and even microcomputers, for which relatively sophisticated expert system development tools are now available. Just as the evolution from tailor-made single-purpose expert systems to shells made expert systems more universally accessible, the transition from mainframe to minicomputer is causing them to be adapted for use in all areas of engineering.

EXPERT SYSTEM COMPONENTS

Expert systems are composed of major and separate parts, a knowledge base, and an inference engine. In addition, there are facilities for creation and maintenance of the knowledge base.

Knowledge Base

The knowledge base is made up of data and rules by which conclusions are reached. The data set may be any type of data describing a system, such as background information, historical records, and results of tests. Rules may be laws, mathematical proofs, heuristics, gut feelings, or common sense. Rules are the standards against which the data are manipulated in order to draw conclusions about a problem. There are a number of useful structures for rules, but the most common in practice is

Austin Research Engineers, Inc., 2600 Dellana Lane, Austin, Tex. 78746.

the IF-THEN-ELSE statement. This type of rule measures whether a condition exists (IF); if it does, one action is taken (THEN); otherwise an alternative action results (ELSE). An example of such a rule is as follows:

IF (given level of pavement distress),
THEN (suggested maintenance or rehabilitation),
ELSE (check for other distress types).

Rules are meant to duplicate the knowledge that a human expert brings to the problem-solving process.

Inference Engine

An inference engine is a collection of processing procedures for examining data-using rules. In one sense, the form of the processing procedures defines the structure of the rules. It can also be said that the type of data and rules determine which type of inference engine is appropriate. There are a number of possible structures for decision making, including the IF-THEN-ELSE statements previously discussed, Markov chains, multidimensional decision trees, and knowledge frames. Each of these structures has its place depending on the situation.

The inference engine is also defined by the way in which the reasoning process flows. Early inference engines used a scheme of production rules *(1)*. The program determined IF-THEN rules in the knowledge base for which there was sufficient information to satisfy the conditional part of the rule. Those rules with satisfied IF statements, were then "fired." The results of fired rules were then checked to see if the goal was met. If the THEN portion of a fired rule matched the goal, the problem was solved. If none of the new information provided by fired rules matched the goal, the facts known by the program, including this new information, were then reviewed to see if additional rules could be fired. This type of reasoning is known as forward chaining or bottom up reasoning *(2)*. The flow of the process is from the low-level information up to the goal.

The process used in MYCIN was backward chaining or top down reasoning. With this type of flow, the program begins with a hypothesis, and determines if any of the THEN portions of rules match. Those that do are triggered, resulting in new information from the IF portion of the rules. The process is completed when the program either reaches facts that are a part of the data base or has to ask for more information. Backward chaining became the standard structure of subsequent expert systems, but many expert system development tools currently available allow the system developer to choose between the two methods. A third alternative is to use some combination of forward and backward chaining. With this method, the tree of information known to the program grows from both the original information and the hypothesis, hopefully meeting to prove the hypothesis.

Another aspect of the inference engine is its ability to handle uncertainty. This facility is especially important for pavement management, where data are not always easily measured and may be collected over a period of many years. Not only is there uncertainty about the correctness of the data, but many of the relationships between different parts of the pavement structure and its relationship to the environment have not been precisely quantified. As a result, there may be a great deal of uncertainty within rules. There are several ways to account for uncertainty in the decision-making process including probability, fuzzy sets, and schemes developed especially for expert systems *(5)*. The developers of MYCIN found that probability was not a concept intuitively grasped by system users, and developed a method of assigning a number from −1 to 1 for the certainty placed on additional information provided by the user.

As previously stated, MYCIN was the first expert system with the ability to explain its reasoning. In conjunction with MYCIN, a program called TEIRESIAS provided a limited ability to answer users' questions on why a certain line of reasoning was followed *(1, 2)*. Many of the expert system shells available for microcomputers have this facility at least in a rudimentary form. This ability may be as limited, as in showing users which rules were invoked, and the paths taken. The value of this feature should not be underestimated; one of the most difficult tasks of computer programming is debugging programs for errors in logic. A means of following the flow of programs is essential for ensuring that the rules drive the reasoning process as intended. Some of the available shells also allow the system developer to add a bit of explanatory text to rules as they are entered *(3)*. In response to a query from the user, the program then displays the text.

Rule Maintenance

In addition to the knowledge base and inference engine, expert systems shells have facilities to build and maintain knowledge bases. At present these include some type of word processing interface to allow the system developer to add and change rules. Although this part of expert systems development has generally received the least attention, it is the most difficult to achieve. Eliciting rules from experts is difficult because often experts don't really know how they solve problems. In many cases they have not tried to quantify the steps to a decision. Another difficulty is that experts often disagree on the causes of problems and acceptable solutions. Creating rules from divergent positions requires a very experienced system developer.

There must also be a means of querying the expert system and receiving the results. A great deal of artificial intelligence research is directed toward developing natural language interfaces. The outcome of this research would be a program that could make sense of a request made in plain English (or any other language), and respond accordingly. Much of the software being offered today claims to have natural language facilities, but there is little evidence to support those claims. Expert system development tools have not advanced to a stage at which the average person with the need for a system will have the resources to devote to develop one. As is the case with PMSs, agencies with a use for such a system will generally look outside for development.

Current Research

There are a number of enhancements to expert systems that are currently under development. One is frames, groups of interrelated rules and data *(6)*. Framing allows faster and more efficient information exchange, reduces redundancy and conflicts of rules or data, and will aid in the eventual development of model-based expert systems.

Another important area of research is rule checking. Ideally, the expert system should locate rules that:

1. Have indefinite conditions or conclusions,
2. Have become obsolete,
3. Are triggered too often,
4. Are never triggered, or
5. Conflict with other rules or data.

The first three cases are based primarily on the common sense and experience of the expert. At present, the sophistication of expert systems is well below what is required to simulate common sense; however, the last two potential problems are manifested much more directly and distinctly. There is some current effort to include these two types of rule checking into the inference engine.

Recent research has also focused on improving the inference engine's explanation capabilities. Software currently available allows explanatory text to accompany rules. This is a somewhat superficial solution to a problem that might be more thoroughly handled by backward chaining. In essence, the inference engine would retrace its steps in the decision process to expose a critical path. The explanation process has many applications, including debugging, rule calibration, and teaching. As microcomputer speed increases and storage capacities are expanded, much more emphasis will be placed on this aspect.

As described earlier in this paper, the ability to handle uncertainty in the knowledge base is an important feature of the inference engine. Though some recent effort has been made to develop this feature, the primary focus of current research is on soliciting information about uncertainty from the user.

PMS as an Expert System

Pavement management is an excellent test bed for expert systems as it can be argued that PMSs in their current states are rudimentary expert systems, much like the precursors to MYCIN. Current PMSs lack a clear division between their inference engine (normally a single decision tree) and the rule base (typically breakpoints for pavement distress severities and extents). PMSs also lack any explicit explanation capability. As PMSs evolve, there is considerable opportunity for advancing expert system research, primarily in the area of rule manipulation. Data requirements have generally been established. In some areas there is a history of data collection, and there is general agreement on how to quantify pavement serviceability and failure *(7)*.

PMSs provide a unique environment for rule-based evolution for three reasons. First, with continued periodic data collection, there will be opportunities to develop rules from the data to replace the heuristics originally supplied by the experts. Second, as the pavement management system works to improve the road network, the goals of the PMS will change. Third, pavement management is a field in which the recognized experts, whose knowledge will originally be incorporated into the knowledge base, have as counterparts local experts whose experience with local climate, traffic, equipment, materials, work rules, and politics is equally important to the development of a comprehensive knowledge base. It has been suggested that the term expert system is a misnomer, that these systems ought to be called knowledge systems instead, to emphasize the importance of information supplied by those other than acknowledged experts *(8)*. In short, a PMS's rule base should never remain static. The unique organization of expert systems will satisfy this requirement and provide a valuable opportunity to improve the system as well.

LONG-TERM RESEARCH

As computer hardware continues to develop in processing speed and storage capabilities, and as artificial intelligence software becomes more standardized and mature, the field of expert systems will experience phenomenal development. This development will enable computers to simulate human intelligence more accurately rather than to simply respond to input with programmed responses. Two noteworthy possibilities with respect to PMS are Model Based Expert Systems and Intelligent Data Bases.

Model Based Expert Systems

Until now the mechanisms with which the inference engine of an expert system has drawn conclusions are simple rules. These rules, most often in the form of IF-THEN-ELSE statements, are nothing more than sets of conditions associated with instructions to be followed or conclusions to be drawn. In essence, the rules direct the investigation of the inference engine but give no insight into why a line of reasoning is followed. A new generation of model-based expert systems will include the reasons for making inferences and deductions in a certain way *(2)*. Mathematical and heuristic models are by no means new to computer science, engineering, or pavement management; however, the development of an inference engine that can use models to process data, yet remain functionally independent from the knowledge, is far from trivial. Just as the organization of a rule-driven inference engine determines the structure of the rules, so must the model-driven inference engine determine the structure of the models. There must therefore be strict definitions for the form and purpose of the models in the knowledge base.

As the rules in a rule-based system become proven, mature, and properly framed, they will no doubt provide assistance in developing models for future systems. But as models are developed, there must be a concurrent effort to develop model-checking capabilities corresponding to rule-checking features now being researched. This is of foremost importance as models are dimensionally far more complex than are rules. The cause-and-effect relationship of a rule is inherent in its structure. The results of applying individual rules to data are generally transparent. The difficulty in proofing rules comes primarily in their interactions with each other. Models on the other hand are composed of rules. They are often based on intuition and common sense, and applying a model to data can result in actions completely opaque to the user. Thus any inconsistencies, conflicts, or omissions in a model because of unwarranted assumptions may result in subtle errors or divergences which may easily go undetected.

Intelligent Data Bases

A possible solution to the problems of rule/model development and checking may be found in an area of research known as intelligent data bases. Intelligent data bases are distinguished from expert systems in that expert systems use rules or models to derive conclusions from data, whereas intelligent data bases attempt to create rules and models from data. Early work in this area has involved pattern matching, dealing primarily with symbolic representation of data. Although this abstract form of correlation shows promise, researchers remain far from producing computer programs that can derive rules or models from data without human assistance. Nevertheless, the structure of expert systems' rules and models and the organization of the knowledge base into frames provide an excellent conceptual environment for the human expert to apply his experience, expectations, and understanding to improve rules and develop new models. Furthermore, while an intelligent data base may be years in coming, the same concepts can begin to be applied to monitoring rules or models supplied by human experts. Monitoring would be in the form of: *(a)* flagging data that are exceptions to rules, *(b)* indicating divergence in the data from conditions predicted by the models, and *(c)* indicating those rules or models that are either triggered more or less often than expected.

CONCLUSIONS

Unquestionably, expert systems will need considerably more development before they can live up to their expectations. This, however, is not to say that they provide no benefits in their current stage of development. In fact, expert systems present several advantages over PMSs as they are currently formulated.

First, the structure of an expert system is well defined without limiting the analytical or theoretical approaches to data reduction. This can also give definition to an as yet unstandardized category of computer programs, PMS. The most beneficial aspect of this structure is in the separation of the inference engine and the knowledge base. Once the mechanics of the inference engine and the structure of rules and data have been determined, the computer code need not be rewritten whenever new rules or data are added. Thus, the burden on the system programmer is relieved and the maintenance of the PMS is placed back in the hands of the experienced engineer. The pavement engineer is most familiar with the data and is responsible for the answers produced by the system. It is therefore appropriate that he be entrusted with the architecture of the system. Furthermore, the flexibility of the knowledge base allows for continual improvements and updates, and provides the local engineer with a means to customize the knowledge base by incorporating his special knowledge of local conditions.

The second advantage of expert systems is that they can directly address uncertainty in the knowledge base. Uncertainty in data can be handled by appropriate rules and input structures. Rule uncertainty is currently being addressed with techniques such as Markov chains. The task of quantifying uncertainty in the data and rules is still the responsibility of the engineer; however, once the uncertainty has been determined, expert systems can support a structured means for analytically or heuristically accounting for uncertainty.

A third important advantage of expert systems is their ability to explain the reasoning employed to reach conclusions. An inference engine should have the ability not only to use the rules and data in the knowledge base to draw conclusions, but also to retrace its path to explain which rules and data were critical to the conclusion. The benefits of explanation are threefold:

1. Erroneous or inconsistent rules are exposed,
2. The user can have more confidence in the answers received from the system, and
3. The system can be used as a learning tool.

Finally, it should be noted that expert systems are only intended to aid the engineer. They are not substitutes for experience and common sense. As stated previously, the most difficult and important part of developing an expert system is soliciting experts' knowledge. There will inevitably be failures when trying to quantify knowledge that is based on many years of experience. Some expert advice is derived entirely from the expert's intuition, which is inherently unquantifiable although still a valuable source of information. Thus, expert systems are at best tools to organize and enhance PMSs that show great future potential. They will never replace an experienced pavement engineer, and should not be touted as the final solution for PMSs.

REFERENCES

1. A. Barr and E. A. Feigenbaum. *The Handbook of Artificial Intelligence.* William Kaufmann, Inc., Los Altos, Calif., 1981.
2. J. Amsterdam. Expert Systems. *Popular Computing*, May 1985, p. 70.
3. J. Helliwell. GURU: Brave New Expert System? *PC Magazine*, May 27, 1986, p. 151.
4. P. Kinnucan. Software Tools Speed Expert System Development. *High Technology*, March 1985, p. 16.
5. K. Lecot and D. Stott Parker. Control Over Inexact Reasoning. *AI Expert*, Premier 1986, p. 32.
6. J. Rothfelder. Is There Intelligent Life in the PC? *PC Magazine*, Jan. 14, 1986, p. 139.
7. The AASHO Road Test, Report 5: Pavement Research. In *Special Report 61E*, HRB, National Research Council, Washington, D.C., 1962.
8. P. Harmon and D. King. *Expert Systems.* John Wiley and Sons, Inc., New York, 1985.

Publication of this paper sponsored by Committee on Pavement Management Systems.