National Cooperative Highway Research Program

NCHRP Synthesis 183

Knowledge Based Expert Systems in Transportation

A Synthesis of Highway Practice

Transportation Research Board
National Research Council

NATIONAL RESEARCH COUNCIL

TRANSPORTATION RESEARCH BOARD

2101 Constitution Avenue Washington, D.C. 20418

Office Location
2001 Wisconsin Avenue, N.W.
Washington, D.C.

Telephone: (202) 334-2934
Telex: 248664 NASWUR
Telefax: (202) 334-2003

November 13, 1992

REPLY TO: Robert J. Reilly
202/334-3224

TO: CHIEF ADMINISTRATIVE OFFICERS
STATE HIGHWAY AND TRANSPORTATION DEPARTMENTS

FROM: Thomas B. Deen
Executive Director

SUBJECT: National Cooperative Highway Research Program
Synthesis of Highway Practice 183, *Knowledge Based Expert Systems in Transportation*
Final Report, Project 20-5, Topic 22-09, of the FY '88 Program

A copy of the subject report is enclosed. Individuals who have selected the Areas of Interest shown on the title page of the report will also receive a copy .

The NCHRP staff has provided a foreword that succinctly summarizes the scope of the work and indicates the personnel who will find the results of particular interest. This will aid in the distribution of the report within your department and in practical application of the research findings. These findings add substantially to the body of knowledge concerning the use of advanced computer applications for highway design and operations. Information is provided on the history of knowledge based expert systems (KBES), current applications of these systems in transportation departments, potential applications, and hardware and software requirements. Additionally, some detailed programming information from two operational expert systems is included.

Enclosure

National Cooperative Highway Research Program

Synthesis of Highway Practice 183

Knowledge Based Expert Systems in Transportation

LOUIS F. COHN
University of Louisville
Louisville, Kentucky

and

ROSWELL A. HARRIS
University of Louisville
Louisville, Kentucky

Topic Panel

MICHAEL J. DEMETSKY, *University of Virginia*
MAS HATANO, *California Department of Transportation*
MARY ANN PAULIS, *Illinois Department of Transportation*
ROBERT J. PERRY, *New York State Department of Transportation*
KING W. GEE, *Federal Highway Administration*
GRAHAM H. POWELL, *University of California, Berkeley*
JAMES A. SCOTT, *Transportation Research Board*

Research Sponsored by the American Association of State Highway and Transportation Officials in Cooperation with the Federal Highway Administration

TRANSPORTATION RESEARCH BOARD
National Research Council
Washington, D. C.

September 1992

NATIONAL COOPERATIVE HIGHWAY RESEARCH PROGRAM

Systematic, well-designed research provides the most effective approach to the solution of many problems facing highway administrators and engineers. Often, highway problems are of local interest and can best be studied by highway departments individually or in cooperation with their state universities and others. However, the accelerating growth of highway transportation develops increasingly complex problems of wide interest to highway authorities. These problems are best studied through a coordinated program of cooperative research.

In recognition of these needs, the highway administrators of the American Association of State Highway and Transportation Officials initiated in 1962 an objective national highway research program employing modern scientific techniques. This program is supported on a continuing basis by funds from participating member states of the Association and it receives the full cooperation and support of the Federal Highway Administration, United States Department of Transportation.

The Transportation Research Board of the National Research Council was requested by the Association to administer the research program because of the Board's recognized objectivity and understanding of modern research practices. The Board is uniquely suited for this purpose as: it maintains an extensive committee structure from which authorities on any highway transportation subject may be drawn; it possesses avenues of communications and cooperation with federal, state, and local governmental agencies, universities, and industry; its relationship to the National Research Council is an insurance of objectivity; it maintains a full-time research correlation staff of specialists in highway transportation matters to bring the findings of research directly to those who are in a position to use them.

The program is developed on the basis of research needs identified by chief administrators of the highway and transportation departments and by committees of AASHTO. Each year, specific areas of research needs to be included in the program are proposed to the National Research Council and the Board by the American Association of State Highway and Transportation Officials. Research projects to fulfill these needs are defined by the Board, and qualified research agencies are selected from those that have submitted proposals. Administration and surveillance of research contracts are the responsibilities of the National Research Council and the Transportation Research Board.

The needs for highway research are many, and the National Cooperative Highway Research Program can make significant contributions to the solution of highway transportation problems of mutual concern to many responsible groups. The program, however, is intended to complement rather than to substitute for or duplicate other highway research programs.

NOTE: The Transportation Research Board, the National Research Council, the Federal Highway Administration, the American Association of State Highway and Transportation Officials, and the individual states participating in the National Cooperative Highway Research Program do not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of this report.

NCHRP SYNTHESIS 183

Project 20-5 FY 1990 (Topic 22-09)
ISSN 0547-5570
ISBN 0-309-05307-2
Library of Congress Catalog Card No. 92-064259

Price $9.00

Subject Areas
Highway and Facility Design

Mode
Highway Transportation

NOTICE

The project that is the subject of this report was a part of the National Cooperative Highway Research Program conducted by the Transportation Research Board with the approval of the Governing Board of the National Research Council. Such approval reflects the Governing Board's judgment that the program concerned is of national importance and appropriate with respect to both the purposes and resources of the National Research Council.

The members of the technical committee selected to monitor this project and to review this report were chosen for recognized scholarly competence and with due consideration for the balance of disciplines appropriate to the project. The opinions and conclusions expressed or implied are those of the research agency that performed the research, and, while they have been accepted as appropriate by the technical committee, they are not necessarily those of the Transportation Research Board, the National Research Council, the American Association of State Highway and Transportation Officials, or the Federal Highway Administration of the U.S. Department of Transportation.

Each report is reviewed and accepted for publication by the technical committee according to procedures established and monitored by the Transportation Research Board Executive Committee and the Governing Board of the National Research Council.

The National Research Council was established by the National Academy of Sciences in 1916 to associate the broad community of science and technology with the Academy's purposes of furthering knowledge and of advising the Federal Government. The Council has become the principal operating agency of both the National Academy of Sciences and the National Academy of Engineering in the conduct of their services to the government, the public, and the scientific and engineering communities. It is administered jointly by both Academies and the Institute of Medicine. The National Academy of Engineering and the Institute of Medicine were established in 1964 and 1970, respectively, under the charter of the National Academy of Sciences.

The Transportation Research Board evolved in 1974 from the Highway Research Board, which was established in 1920. The TRB incorporates all former HRB activities and also performs additional functions under a broader scope involving all modes of transportation and the interactions of transportation with society.

Published reports of the

NATIONAL COOPERATIVE HIGHWAY RESEARCH PROGRAM

are available from:

Transportation Research Board
National Research Council
2101 Constitution Avenue, N.W.
Washington, D.C. 20418

Printed in the United States of America

PREFACE

A vast storehouse of information exists on nearly every subject of concern to highway administrators and engineers. Much of this information has resulted from both research and the successful application of solutions to the problems faced by practitioners in their daily work. Because previously there has been no systematic means for compiling such useful information and making it available to the entire highway community, the American Association of State Highway and Transportation Officials has, through the mechanism of the National Cooperative Highway Research Program, authorized the Transportation Research Board to undertake a continuing project to search out and synthesize useful knowledge from all available sources and to prepare documented reports on current practices in the subject areas of concern.

This synthesis series reports on various practices, making specific recommendations where appropriate but without the detailed directions usually found in handbooks or design manuals. Nonetheless, these documents can serve similar purposes, for each is a compendium of the best knowledge available on those measures found to be the most successful in resolving specific problems. The extent to which these reports are useful will be tempered by the user's knowledge and experience in the particular problem area.

FOREWORD

By Staff
Transportation Research Board

This synthesis will be of interest to engineering managers, design engineers, traffic engineers, computer personnel, and others interested in advanced computer applications for highway design and operations. Information is provided on the history of knowledge based expert systems (KBES), current applications of these systems in transportation departments, potential applications, and hardware and software requirements. Additionally, some detailed programming information from two operational expert systems is included.

Administrators, engineers, and researchers are continually faced with highway problems on which much information exists, either in the form of reports or in terms of undocumented experience and practice. Unfortunately, this information often is scattered and unevaluated, and, as a consequence, in seeking solutions, full information on what has been learned about a problem frequently is not assembled. Costly research findings may go unused, valuable experience may be overlooked, and full consideration may not be given to available practices for solving or alleviating the problem. In an effort to correct this situation, a continuing NCHRP project, carried out by the Transportation Research Board as the research agency, has the objective of reporting on common highway problems and synthesizing available information. The synthesis reports from this endeavor constitute an NCHRP publication series in which various forms of relevant information are assembled into single, concise documents pertaining to specific highway problems or sets of closely related problems.

There is growing use of computers in transportation departments, and knowledge based expert systems (KBES) represent an area in which several highway agencies are gaining experience and obtaining promising results. This report of the Transportation

Research Board describes the current state of the practice with respect to KBES, as well as the historical development of expert systems and the more general field of artificial intelligence (AI). Experience with expert systems in transportation is summarized, including discussions of expert systems in operation and in development, based on a review of the literature and a survey of the states and experts in this field.

To develop this synthesis in a comprehensive manner and to ensure inclusion of significant knowledge, the Board analyzed available information assembled from numerous sources, including a large number of state highway and transportation departments. A topic panel of experts in the subject area was established to guide the researcher in organizing and evaluating the collected data, and to review the final synthesis report.

This synthesis is an immediately useful document that records practices that were acceptable within the limitations of the knowledge available at the time of its preparation. As the processes of advancement continue, new knowledge can be expected to be added to that now at hand.

CONTENTS

ACKNOWLEDGMENTS

This synthesis was completed by the Transportation Research Board under the supervision of Robert E. Skinner, Jr., Director for Special Projects. The Principal Investigators responsible for conduct of the synthesis were Sally D. Liff, Senior Program Officer and Scott A. Sabol, Program Officer. This synthesis was edited by Linda Mason.

Special appreciation is expressed to Louis F. Cohn, Professor and Chairman of the Department of Civil Engineering, University of Louisville; and Roswell A. Harris, Associate Professor, Department of Civil Engineering, University of Louisville, who were responsible for collection of the data and preparation of the report.

Valuable assistance in the preparation of this synthesis was provided by the Topic Panel, consisting of Michael J. Demetsky, Professor of Civil Engineering, University of Virginia; Mas Hatano, Supervising Materials and Research Engineer, California Department of Transportation; Mary Ann Paulis, Systems Development Manager, Illinois Department of Transportation; Robert J. Perry, Director, Engineering Research, New York State Department of Transportation; Graham H. Powell, Professor of Civil Engineering, University of California at Berkeley; and King W. Gee, Chief, Technical Assessment Branch, Federal Highway Administration.

James A. Scott, Transportation Planner, Transportation Research Board, assisted the NCHRP Project 20-5 Staff and the Topic Panel.

Information on current practice was provided by many highway and transportation agencies. Their cooperation and assistance were most helpful.

KNOWLEDGE BASED EXPERT SYSTEMS IN TRANSPORTATION

SUMMARY

A Knowledge Based Expert System (KBES) is an intelligent computer program that uses the knowledge and inference procedures of human experts to solve difficult problems. This is in contrast to a conventional computer program which is algorithmic in nature, using precisely defined, logical formulas and data. Most state transportation agencies have expertise in developing and using conventional programs, but few have significant experience in expert systems.

Advances in computing have dramatically changed the transportation field in the last twenty years. The next twenty years may see even more changes as computing technology continues to evolve. The widespread application of expert systems may be the key to these changes. Expert systems can capture knowledge currently residing in the transportation work force and make it available to others through knowledge based tutorials. They can automate mundane and repetitious tasks such as permitting, and provide ready access to information in manuals and codes. Knowledge based expert systems can emulate expert colleagues to advise engineers in solving difficult problems, and even assist in designing facilities.

While expert systems projects have been attempted in most areas of transportation, the largest portion has been in the areas of pavements and traffic. The potential for future applications, as assessed by the state transportation agencies and members of the American Society of Civil Engineers (ASCE) and Transportation Research Board (TRB) Expert Systems Committees, appears to be greatest in these areas, as well as in the construction area.

State transportation agencies were surveyed by questionnaire to obtain much of the information for this synthesis. The survey results clearly showed a desire among the state transportation agencies to use expert systems. In fact, several, most notably California, Connecticut, Illinois, New York, Pennsylvania, and Virginia, have developed one or more working and productive expert systems. Other state transportation agencies have prototype projects underway.

When asked to recommend topics that have exceptional potential for expert systems applications, the state transportation agencies responded with an impressive array of ideas. This suggests that there is a good understanding of expert systems among the transportation engineering community, and a strong desire to make full use of this emerging tool.

To set forth the basic principles of KBES, a primer accompanies this summary. Following a question and answer format, the primer provides a quick introduction to the vocabulary and procedures common to expert systems. More detailed information on specific subjects related to KBES is given in the body and appendices of the synthesis.

The history and some fundamental concepts of KBES are briefly described, including

the link between expert systems and the overall field of artificial intelligence. Applications of KBES to transportation issues are discussed, including applications cited in the transportation literature, as well as several current state transportation agency applications; a detailed California Department of Transportation strategic study on KBES and the results of the questionnaire conducted for this synthesis are also presented. Information is also provided on topics of interest to agencies getting started with KBES, including staffing issues and the need to maintain and support both the hardware and software for KBES beyond the initial acquisition and implementation. Some detailed technical information on KBES architecture is provided, with discussion on the various components of KBES and how such systems are implemented. The appendices include a glossary of terms associated with expert systems, describe in detail two current expert systems applications, and provide a resource guide for KBES and artificial intelligence software.

A PRIMER ON KNOWLEDGE BASED EXPERT SYSTEMS

Knowledge Based Expert Systems represent an emerging field in technology. Most practicing transportation professionals have not had the opportunity to learn about or use them. Since the concept of expert systems originated in the computer science field of artificial intelligence, a high degree of abstraction is associated with it. Most transportation professionals are not accustomed to the vocabulary and procedures common to expert systems. As a result, they may need some background information on the topic before this synthesis can be useful. In order to provide this background information, the following questions and answers were developed. By reviewing these questions and answers, it is hoped that a foundation will be laid for further understanding of the details to follow in the body of the synthesis. The questions and answers contained in this primer can also be used as an executive summary or a briefing paper to introduce the topic to agency personnel.

In later chapters, there is discussion of specific applications of knowledge based expert systems to transportation. Among the areas where development has been especially significant are pavement management, traffic signal systems, noise barrier design, and structural analysis. These and other areas show continued potential for even further work in expert systems.

1. What is artificial intelligence?
Artificial Intelligence (AI) is a specialty area of computer science that attempts to make computers behave in a way that mimics logical human behavior. Robotics, image processing, and pattern recognition, as well as knowledge based expert systems, are branches of artificial intelligence.

2. What is a knowledge based expert system?
A Knowledge Based Expert System (KBES) is a computer program that emulates human behavior in solving problems. It includes a separate reasoning mechanism that performs the same function as a human expert's brain.

3. How is an expert system "expert"?
An expert system is structured like a human brain. It includes a memory recall function and an inference mechanism. It processes information and data in much the same way a human expert draws on the skill, knowledge, and judgement gained through experience to reach conclusions about a problem.

4. What is natural language programming?
Expert systems use computer languages that consist of natural language commands,

rather than the alphanumeric characters found in conventional, algorithmic computer programs. Among the languages used for expert systems are LISP (LISt Processing), and PROLOG (logic programming). Conventional programming languages include FORTRAN, C, and Pascal.

5. Is an expert system a "black box"?
No. As with any other computer program, the results of a knowledge based expert system problem-solving effort should be evaluated for accuracy and reasonableness. Computer output should always be viewed as advisory.

6. What is the difference between an expert system and a traditional, algorithmic computer program?
An expert system solves a problem in the same style as a human expert, with logic and reasoning, while an algorithmic program simply solves a series of equations by inserting values (input data) into appropriate slots within the equations.

7. When should an expert system be used instead of a conventional program?
An expert systems approach works well when the problem to be solved is complex or ill-defined, and when judgement and experience are useful tools in finding the solution. Such solutions may call for the use of heuristics, or rules of thumb.

8. When should an algorithmic program be used instead of an expert system?
An algorithmic approach works well when the problem is quantitative in nature, such as when the solution fits the form of answers gained from solving equations.

9. What is a prototype expert system?
A prototype is a "first cut" attempt at applying an expert systems solution approach to a problem. A prototype is usually very rough and very short, and the results of the prototype development are discarded once it demonstrates the applicability of a

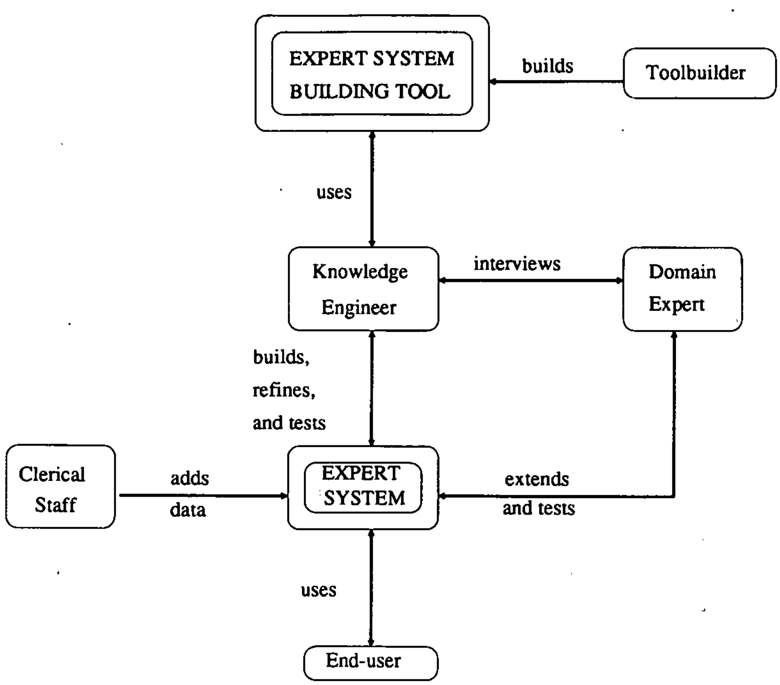


FIGURE P-1 The expert system building process.

knowledge based expert system. One may think of prototyping as batting practice in preparation for a real game.

10. Why develop a prototype first?
Successful prototype development is the best way to show that expert systems can be used to solve a given problem. The prototype is not useful in the operational sense, however, because it lacks the sophistication, attention to detail, and refined user interface needed in an expert system that will be used widely by engineers in a "real world" setting.

11. What is a domain expert?
A specialist in a given subject area, or domain, of transportation is a domain expert (Figure P-1). For example, a roadway design engineer is a domain expert regarding the subject of roadway design.

12. What is a knowledge engineer?
A person, usually proficient in computer science and computer applications, whose specialty is related to artificial intelligence software development (specifically expert systems) is often called a knowledge engineer (Figure P-1). Such a person is involved in knowledge acquisition and knowledge representation. Usually a knowledge engineer teams with domain experts to extract (or acquire) knowledge and expertise about the domain.

13. What is an inference engine?
Inference is the derivation of new facts from known facts. An inference engine is the processor of these facts (Figure P-2). For example, the inference engine for the human being is the brain. A newborn child's brain is an inference engine, but it is void of any

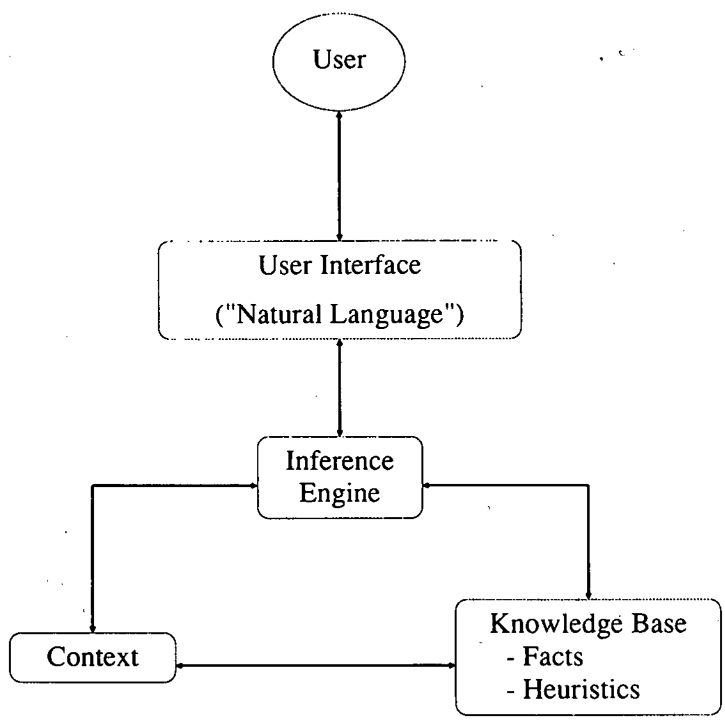


FIGURE P-2 Expert system architecture.

acquired knowledge. Acquired knowledge is analogous to the rule base (or knowledge base) in an expert system. The inference engine within an expert system is the portion of the computer program that processes knowledge.

14. What is a shell?

It is difficult and time consuming for non-computer professionals to write expert systems programs. As a result, many commercially developed, fill-in-the-blank packages, called shells, are available. A shell is the outer layer of a program that provides the user interface with the computer. These shells make expert systems development much easier, but usually restrict programming options. It is important for an expert systems developer to select a shell that contains the features suitable to the particular application.

15. What is a rule?

A rule is a two-part condition statement often used in expert systems. The first part is an IF condition, and the second part is a THEN condition. For example, a simple rule may state: *IF lane width is greater than 9 feet, THEN speed limit may be 50 mph.* A rule-based expert system contains (or represents) knowledge in the form of rules. Such an expert system is said to be a production system.

16. What is a rule base?

A rule base is the knowledge base of an expert system that is a production system. It consists of a series of rules that are linked together via computer commands built into a shell or directly through language programming commands.

17. What is knowledge acquisition?

Perhaps the greatest challenge in expert systems development is gathering the correct amount of useful and relevant knowledge for incorporation into the knowledge base. This effort is knowledge acquisition, and is usually accomplished by interviewing experts, reviewing literature, and, in the case of the domain expert who also serves as the knowledge engineer, introspection (Figure P-1). It is important to acquire static knowledge, such as facts from manuals, codes, textbooks, and so on, as well as heuristic information, such as rules-of-thumb developed by experienced experts.

18. What is knowledge representation?

Once the appropriate knowledge has been acquired, it must be encoded into a suitable format for examination by the inference engine. This encoding methodology is known as knowledge representation. The key to effective knowledge representation is to find the best way to present the knowledge so that it can be accessed efficiently.

19. How is an expert system modified?

An expert system is deemed to be operational when it has a complete knowledge base and a user interface suitable for efficient use by domain specialists. Both the knowledge base and the user interface will be linked to the inference engine through software developed directly with a programming language or a shell. If all of this development has been done properly, the only modifications likely to be needed will address changes or advances in the knowledge applied to the domain. (An example would be a change in a design criterion.) In such cases, the expert system could be modified by adding, deleting, or changing rules within the knowledge base. It would not usually be necessary to make changes to other parts of the program; this simplified method of incorporating changes is one of the strengths of the expert systems approach.

20. How is an expert system validated?
A traditional algorithmic program is validated by checking hand calculation results against numerical results produced by the program. This is a relatively straightforward task. An expert system, however, usually produces advice, the validity of which is difficult to assess with an absolute degree of certainty. The safest way to validate an expert system is to subject it to repeated encounters with case studies performed by human experts. When the expert system is able to replicate the solutions offered by the human experts on a wide array of real problems, then it could be considered validated.

21. When can the results produced by an expert system be trusted?
There is no circumstance when transportation personnel should blindly accept results from any computer program. Even after an expert system has been adequately validated, its results should be examined with a critical eye, and should be considered advisory only.

22. How much would it cost to produce a working expert system for a transportation application?
The limited experience to date indicates that an operational expert system requires from one to three person-years of effort. Range of cost in dollars for the software development would be $75,000 to $150,000 in 1992 costs, depending on the complexity of the problem and the level of refinement needed.

23. What background is needed for someone beginning expert systems programming?
Most civil engineers with training and interest in computers can learn to use programming shells with a little effort. Actual programming in an artificial intelligence language such as LISP or PROLOG, however, requires substantial effort. It should be stressed that programming professionals are usually needed if the desire is to produce an expert system with a sophisticated structure and user interface.

24. How is an expert system maintained and upgraded?
All computer programs require the ongoing attention of knowledgeable professionals to make sure they are still suitable for application, and to address possible problems or glitches. Expert systems are no exception. However, they are usually easier to modify due to the separation of the knowledge base from the inference engine.

25. What kind of computers are best for use with expert systems?
To be effective, an operational knowledge based expert system will need computational speed. Ideally, a microcomputer used for expert systems will be of the 386 class. Virtually all of the modern mini- and mainframe computers will be ample for transportation applications.

CHAPTER ONE

BACKGROUND

Purpose and Scope of the Synthesis

This synthesis is intended to provide background information on the state of the art in expert systems, past and current applications to transportation problems, areas where expert systems could be applied, and what the potential is for integrating KBES technology into the practice of transportation engineering. This document may be used as a guide for state transportation agencies in getting started with expert systems. It should also help to prevent duplication of effort among state agencies and to identify high-priority application areas for expert systems. A glossary is provided in Appendix A.

Knowledge based expert systems, a branch of Artificial Intelligence (AI), is the principal topic of this synthesis. However, expert systems do play an integrated role with the other AI branches, including robotics, neural networks, and so on. While this synthesis focuses on expert systems, there is some brief discussion of the other AI branches in various places throughout the document. Several state transportation agencies are involved with other AI tools such as neural networks, robotics, image processing, and pattern recognition.

HISTORY OF KNOWLEDGE BASED EXPERT SYSTEMS

As computer technology advanced from its origin in the mid 1940s, it became evident that this form of technology easily lends itself to a variety of tasks. Executing long and tedious computations and rapidly storing, retrieving, and sorting immense amounts of data are a few that can be named. These abilities paved the road for rapid development in a number of disciplines.

Computers were used primarily to solve formal and analytical problems. However, problems to be solved by traditional programming methods must be structured into specific sequential statements that are executable. As a result, computers were initially confined to solving only those problems with well-understood solutions. Realistically, many problems are complex and do not have algorithmic solutions. Planning, medical diagnosis, geological exploration, military situation analysis, and many issues in transportation are typical of these types of problems. Their solutions depend heavily on the manipulation of descriptive terms and the selective application of relevant pieces of knowledge (*1*). These problems are usually solved by experts using their past experience and knowledge.

During these formative years, AI researchers tried to develop techniques that would allow computers to emulate human behavior. General purpose programs were developed with the idea that a few laws of reasoning coupled with powerful computers would result in expert and superhuman performance. These attempts achieved minimal success; but the idea of expert systems caught on, and many researchers began work on narrowly defined applications (*2*).

The challenge of making a computer program intelligent inspired scientists to take a different approach. Since making a general purpose program was determined infeasible, efforts were made to develop techniques for use in more specialized programs. Therefore, the 1970s yielded progress in techniques such as representation (formulating the problem in such a way that it is more easily solved) and search (searching cleverly for a solution so it does not consume too much time or computer memory) (*2*). In spite of some success, there were no major breakthroughs.

In the late 1970s, scientists realized that the problem-solving capability of a program results from the knowledge it contains, not the formalisms and inference schemes it employs (*3*). Recognizing that knowledge is as important as reasoning, AI researchers worked on a variety of ways of representing and using knowledge, and investigated the types of problems in which logic is supplemented by rules of thumb based on experience and judgment.

Paul E. Johnson (*2*), a scientist who has spent many years studying the behavior of human experts, defines an expert as:

> [A] person who, because of training and experience, is able to do things the rest of us cannot; experts are not only proficient but also smooth and efficient in the actions they take. Experts know a great many things and have tricks and caveats for applying what they know to problems and tasks; they are also good at plowing through irrelevant information in order to get at basic issues, and they are good at recognizing problems they face as instances of types with which they are familiar. Underlying the behavior of experts is the body of operative knowledge we have termed expertise. It is reasonable to suppose, therefore, that experts are the ones to ask when we wish to represent the expertise that makes their behavior possible.

The task of capturing and formalizing human expertise, however, proved to be more difficult than initially believed. Though generally accepted rules or formulas are easy to find and encode, human intelligence goes beyond these formalities. The human reasoning processes often use rules of thumb, intuitive feelings, and vague statements derived from years of experience. It was this information that expert systems needed to be truly successful.

At this time, the field of knowledge engineering became an entity. Knowledge engineers assumed the responsibility of working with domain experts to find methods through which a problem could be solved. The next step was to encode the captured knowledge in a format that a computer could accept and manipulate.

In the late 1980s, one of the main trends in AI lay in developing mechanisms by which the machine could learn from its mistakes and acquire new knowledge. Some early results indicated such features could be extremely useful, allowing the knowledge base to be constantly updated and expanded without additional encoding by the human expert (*2*).

These research efforts led to the development of special-purpose computer programs that contain expert knowledge in some narrow problem area. These programs are referred to as knowledge based expert systems (KBES), or simply expert systems. Typically, the user interacts with an expert system in a "consultation dialogue" in the same way as with a human expert: explaining the problem, performing suggested tasks, and asking questions about proposed solutions. Much of the research in this area has been focused on developing an interface between the user and the system, providing a "friendly" and efficient means for the non-expert human to access and use the resident expert knowledge. Other significant efforts have been made toward endowing these systems with the ability to explain their reasoning, both to make the consultation more acceptable to the user and to help the domain expert improve the system's reasoning processes.

Knowledge based expert systems have great potential for solving problems that lack explicit algorithms. They have been defined as intelligent computer programs that use knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solutions (*3*). These expert systems gain their ability to solve problems not by following step-by-step procedures, but by following a few general procedures. Stored within the computer's memory are facts, rules, and other knowledge about solving a problem (domain knowledge). When solving a particular problem, the computer employs specific facts related to the problem that are provided by the user, as well as its domain knowledge and general problem-solving procedure. To arrive at a specific solution, the computer uses this information in a manner comparable to a human expert.

Operational expert systems have already been developed in a number of disciplines. Many of these systems have been designed to attain the level of performance of a human expert in a specific domain. Among the most well known of these systems are CHINA, a highway noise barrier design program (*4*); PARADIGM, a pavement management program (*5*); MYCIN, a medical diagnosis and therapy system (*6*); PROSPECTOR, a mineral exploration system (*7*); DIPMETER, a system that interprets the geologic information gathered during the drilling of oil wells (*8*); and R1, a system that configures new computer installations (*9*). These systems are all operational and have exhibited their ability to perform at a level equal or superior to that of human experts in their respective areas of expertise. Appendix B contains details of CHINA and PARADIGM.

These are but a few of the examples of expert systems available on the market today. Many systems are available in a variety of fields: diagnosis, design, learning, knowledge acquisition, intelligent assistance, and image analysis (*10*). The number and domain of expert systems are increasing as both research and industry continue to successfully use the technology. It is impossible to accurately determine the number of existing systems since most universities have some activity and many industrial expert systems are regarded as highly proprietary (*11*). The U.S. military has also been very active in AI and KBES research and development; this work, however, is not routinely made available to the general scientific and engineering communities. Appendix C supplies a resource guide for KBES and AI software.

THE CONCEPT OF KNOWLEDGE BASED EXPERT SYSTEMS

The introduction of computers has had more impact on the ability of today's engineer to provide more and better-designed products essential to this nation's vitality than any other single technological advance. With the introduction of KBES technology, a new tool is available that has the potential to have an equally significant impact on the profession. Expert systems represent a relatively new field of artificial intelligence, and promise to provide benefits to those engineers willing to embrace this new technology. True expertise in any domain is a scarce resource gained from many years of study and experience. Expert systems can offer an efficient means of making that expertise available to more practicing engineers than is feasible through any other means.

Ironically, the increased activity relative to expert systems in engineering has presented today's engineer with a unique set of new problems. First, so called "expert systems" may be constructed that do not embody or that incorrectly embody the essential knowledge-related characteristics described earlier. Without the expert knowledge inherent in a reliable expert system, the program could lead the user to erroneous conclusions. Programs that employ only textbook knowledge or use only algorithmic solutions to problems are not true expert systems.

In addition, some engineers may see expert systems as a panacea for solving all difficult problems. That is not the intent of this technology. Expert systems are computerized tools that can be used by engineers as aids to help them solve complex problems more easily and effectively.

A related concern is that expert systems are not suited for application to all practical problems. The most obvious requirement for applying an expert system to a given problem is that a human expert must first exist from whom to draw the expert knowledge. The primary sources of that expert's knowledge must be judgment and experience.

The problem itself should involve a narrow domain of application (*12*). Expert systems are particularly appropriate for those problems where:

- No algorithmic solutions are available;
- The model may have to change to meet the requirements of different locations;
- Human experts are scarce and the transfer of knowledge by conventional means is too slow or costly; and
- Substantial knowledge about the problem exists in the minds of the experts, in the experience and heuristics, instead of in textbooks and journals.

A direct result of this AI and KBES activity in the last decade is the development of a number of generic tools (referred to as shells) that allow the experts in a specific subject area (i.e., domain) to construct working expert systems with little experience in artificial intelligence programming techniques.

In fact, availability of these tools has increased much more rapidly than the ability of most potential users to construct reliable expert systems. As will be pointed out, the ability of an expert system to solve a problem is not dependent on the particular programming shell it uses, but rather on the completeness and accuracy of the knowledge base it uses to reach its conclusions. Thus, the critical link in constructing a reliable expert system is the knowledge acquisition process.

The expert system differs from conventional computer programs in a number of ways. The major characteristics (Figure 1) that make expert systems distinct from conventional computer programs include (*13*):

- The data pertinent to the problem (knowledge base) and the control knowledge (inference engine) are kept separate from each other. Unless the user is familiar with the applicable programming techniques in the conventional computer program, it is difficult to change the program or the knowledge contained within the program. The expert system is characterized by a clear separation of the problem-specific knowledge and the mechanism that controls the operation of the system. This separation allows the program to be modified by simply editing the knowledge base.
- The domain knowledge used in the program is readable and understandable.
- Expert systems have some form of explanation facility which, upon the user's request, can explain the reasons that preceded its conclusion.
- The knowledge based expert system can be used with a subset of its ultimate knowledge base, and its knowledge base can be incrementally extended over a period of use without major restructuring.

Expert systems can be classified into different system types according to their function (*2*):

- An interpretation system explains the features of input and output data by inferring what problem state matches the existing information.
- A prediction system infers probable outcomes based on known circumstances.

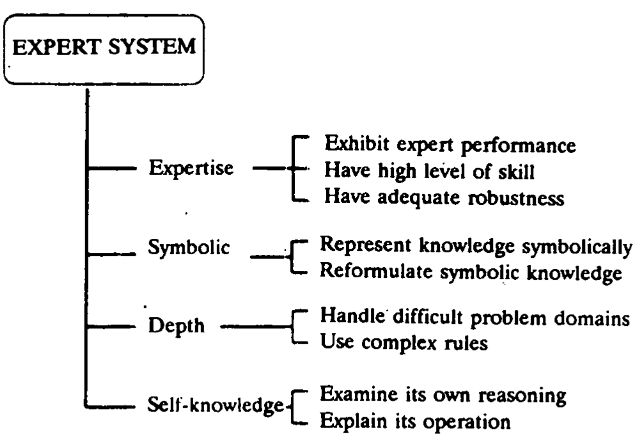


FIGURE 1 Expert system characteristics (from *13*).

- A diagnosis system observes deviations and interprets data in order to detect malfunctions.
- A monitoring system oversees system behavior and compares these observations to desirable system behavior in order to identify any errors in the plan or possible mistakes produced by the system.
- A design system creates a configuration for an object that meets all necessary constraints.
- A repair system corrects any malfunctions or failures detected in the system.
- A control system is a complicated system that incorporates many of the characteristics of the other systems already discussed. It interprets information, diagnoses the problem, predicts results, formulates, carries out, and monitors a plan.

Several properties are common to almost all definitions of expert systems (*14*). These are:

- Expertise,
- Symbol manipulation,
- Uncertainty,
- Complexity,
- Reasoning, and an
- Explanation facility.

Without expertise, the power of the expert system would be drastically limited. Therefore, expertise is the cornerstone upon which the entire system is built.

The next two properties, symbol manipulation and uncertainty, are closely related. Computers are no longer restricted to the quantitative manipulations of the past since the development of symbol processing languages. The fact that expert systems allow for the interpretation of symbols, words, and phrases makes them practical for application in qualitative fields where some values are unknown. This property enables the knowledge engineer to encode the expertise in natural language, which is easier to read and understand.

The fourth characteristic of any expert system is complexity. Again, this property may seem obvious. If the field of knowledge is not complicated, an expert system would not be needed. Looking beyond the obvious, one must appreciate a system that can handle large quantities of interrelated information and draw conclusions from specific information given. Expert systems provide a means of encoding and accessing areas of knowledge deemed far too complex for conventional programming.

The properties named last, reasoning and explanation capabilities, are vital to any expert system. These properties enable the system to be practical for industry and education. The expert system is instructed how to reason, or to think, given certain facts. Once a conclusion is drawn, the system should be able to explain to the user how the decision was reached. Without this capability, the user may fail to trust the conclusion of the system and, in turn, question the advantage of such systems at all.

CHAPTER TWO

CURRENT APPLICATIONS IN TRANSPORTATION

REVIEW OF THE TRANSPORTATION LITERATURE

In addition to the books, articles, and journals referenced in the following chapters, there are several documents with applications specific to transportation issues that contribute to the understanding of KBES. These publications are briefly summarized in the following paragraphs.

ASCE Journal of Transportation Engineering, March 1987 (*15*)

This special section of the *Journal* contained four papers that were expanded and submitted after being presented at the 1986 ASCE Annual Convention in Seattle, Washington. Three of those papers concerned surface transportation, and one concerned air transportation.

The first, "Expert System for Traffic Signal Setting Assistance," by Zozaya-Gorostiza and Hendrickson, presented an experimental KBES to assist in traffic signal setting for isolated intersections. In contrast to existing computer aids, the system is applicable to intersections of highly irregular geometries. Algorithmic processes to evaluate signal settings and decision tables to identify traffic flow conflicts are invoked by the expert system; phase distribution of flows is performed by applying heuristic rules.

The second paper, "Designing Noise Barriers Using the Expert System CHINA," by Harris et al., describes CHINA (Computerized HIghway Noise Analyst), a KBES developed to interact with an existing FORTRAN model.

"Surface Condition Expert System for Pavement Rehabilitation Planning," by Ritchie et al., describes SCEPTRE, a KBES that assists highway engineers in planning cost-effective flexible pavement rehabilitation strategies at the project level. SCEPTRE is based on the premise that pavement surface condition data are critical in the analysis and design of pavement rehabilitation strategies. In project design, this information is usually used by a limited number of pavement engineering specialists who apply experience and judgment to formulate design and investment decisions. Both SCEPTRE, as part of the PARADIGM system, and CHINA are discussed in detail in Appendix B.

The final paper, "Application of Expert Systems in Air Traffic Control," by Gosling, proposes a number of potential applications of KBES in air traffic control. Among the proposed application areas are air traffic flow management, controller support functions, system failure management, training, and system configuration planning. The paper also presents a prototype KBES to assist in air traffic flow management.

ASCE Journal of Computing in Civil Engineering, October 1987 (*16*)

The impetus for this special issue was a publication project sponsored by the ASCE Committee on Expert Systems. Among the papers was "Software for Expert Systems Development," by Ortolano and Perman. This paper presented useful information on the use of shells, programming languages, and programming environments. The remaining papers in the special section synthesized ongoing KBES activity in geotechnical engineering (Santamarina and Chameau), construction planning (Hendrickson et al.), drought management planning (Ortolano and Steinemann), construction (Ashley and Levitt), and structural engineering (Allen).

Expert Systems for Civil Engineers: Technology and Application, 1987 (*17*)

This book was also a project of the ASCE Committee on Expert Systems, with funding provided by the U.S. Army Corps of Engineers' Construction Engineering Research Laboratory (CERL). The first three chapters of the book discuss KBES components, languages, tools, and implementation issues. Chapters Four through Eight discuss applications in the various subdisciplines of civil engineering, including structures, geotechnical, construction, environmental, and transportation.

The transportation chapter was prepared by Ritchie and Harris. It includes a definition of transportation engineering, reasons for applying KBES to transportation problems, and a series of specific KBES applications.

The reasons for applying KBES to transportation problems include:

- Many transportation tasks lack explicit numerical algorithms.
- Many transportation tasks are complex or ill-defined.
- Conventional computer tools are often of limited use.
- Human judgement and experience are heavily used in transportation.

Virtually all of the KBES applications discussed in the transportation chapter were in the prototype stage at the time. They are shown in Table 1.

The chapter concludes with an example application of SCEPTRE on an actual pavement rehabilitation project of the Washington State Department of Transportation (WSDOT). SCEPTRE recommended that the 1.2-mile, two-lane highway in western Washington be preleveled or milled, and then rehabilitated with a medium asphalt concrete overlay (Figure 2). The rating of the section of pavement in question was expected to degrade from 100 percent down to 40 percent after seven years, at which time its service life would be complete. SCEPTRE's recommendations proved to accurately model the decisions made by the WSDOT engineers for this section of highway.

(ASCE publications are available from the American Society of Civil Engineers, 345 East 47th Street, New York, NY 10017)

TABLE 1
KBES APPLICATIONS IN TRANSPORTATION

KBES	Application
LOGOIL	Crude oil distribution logistics
CHINA	Highway noise barrier design
TRALI	Traffic signal setting
EXPERT-UFOS	Large-scale transportation network design
INTERSECTION ADVISOR	Improved intersection operation
HERCULES	Post-disaster network advisor
STREET-SMART	Policy tool for street layout
SCEPTRE	Pavement rehabilitation management
ROSE	Preventive maintenance for highways
PRESERVER	Road maintenance strategies
PIARS	Bridge paint ranking system
PARADIGM	Pavement rehab. analysis and design

Workshop on Knowledge Based Expert Systems in Transportation, 1990 OECD/VTT (18)

In June 1990, the Organization for Economic Cooperation and Development (OECD) and the Technical Research Center of Finland (VTT) jointly sponsored this major workshop/symposium on KBES in transportation. The *Proceedings* that resulted provide an excellent two-volume set. The first volume contains six papers (125 pages) concerning the development and validation of expert systems as they relate to transportation problems. The remaining four sections of the first volume contain papers on specific KBES applications including: *Traffic Management and Control; Traffic Impact Evaluation; Highway Analysis and Planning;* and *Highway Management.* The papers were drawn from authors all over the world, and reflect the state of the art in KBES development into 1990. Of the 15 KBES projects described, all but three were in the prototype stage. All three of the operational expert systems discussed were in the highway management area. They included:

GARPEE. This is a comprehensive network-level KBES developed to assist with the identification of feasible rehabilitation strategies for flexible and rigid pavements. GARPEE was developed by Haas at the University of Waterloo. It contains more than 1,000 "tree branches" and over 500 combinations of 96 rehabilitation strategies. GARPEE considers pavement surface condition, ride quality, loading response, traffic, geometric, and structural data to assign possible rehabilitation strategies to pavement sections in need of some form of rehabilitation. GARPEE uses the Pavement Management System (PMS) to generate needs reports and priority programming recommendations. It has been applied to at least one Canadian city's pavement management needs.

RMPES. The Road Management Planning Expert System (RMPES) is a KBES developed by Pikkarainen for the Finnish National Road Administration (FinnRA). RMPES presents the current state of road and traffic conditions by calculating summaries of the road data bank. Future conditions are predicted using simulation models in which the control parameters are objectives set by the planner. RMPES then presents the costs and quantitative measures necessary for the achievement of set objectives. This presentation is in the form of a calculated budget for the planning period. RMPES has been tested and used in at least seven road districts in Finland.

HDM-III and BSM Enhancements. Two expert systems have been developed by Makarachi and Tillotson of the University of Birmingham in England. These expert systems are part of a project aimed at the intelligent integration of economic appraisal of highway improvements, using the World Bank HDM-III Model, and maintenance management strategies with the program BSM (Burrow-Snaith Model).

HDM-III predicts life-cycle costs for different highway and maintenance options. Construction costs, maintenance costs and road user costs are estimated annually for the life of the road, each with appropriate discounts. The input data required by HDM-III are divided into eleven subgroups, resulting in an extensive and complex input activity. The expert system developed by Makarachi and Tillotson is actually an intelligent front-end which preprocesses the input data to reduce possible errors by less-experienced users.

BSM is a computer-based integrated management system that assists the road maintenance engineer in assigning priorities for maintenance expenditures on paved roads. A large amount of road condition information is stored in a database for use in suggesting and prioritizing possible maintenance activities. The expert system scans the BSM data files to locate as many data as possible for use in HDM-III. The expert system then runs HDM-III and predicts appropriate deterioration levels. The run is intended to establish roadway deterioration limits in the specific BSM system, and then match these limits to available funds.

Volume II of the workshop *Proceedings* is a *Guide for Developing Knowledge Based Expert Systems,* prepared by Wentworth of FHWA while on temporary assignment to OECD. This 30-page document is based on "Developing Expert Systems," the 1988 Technology Share report FHWA-TS-88-022.

(VTT publications are available from the Government Printing Centre (Valtion Painatuskeskus) Hakuninmaantie 2, POB 516, 00101 Helsinki, Finland)

Developing Knowledge Based Expert Systems, FHWA Technology Share Report FHWA-TS- 88-022 (19)

This 27-page document reflects current practical experience for developing KBES applications in highway-related areas. Included in this report are discussions of:

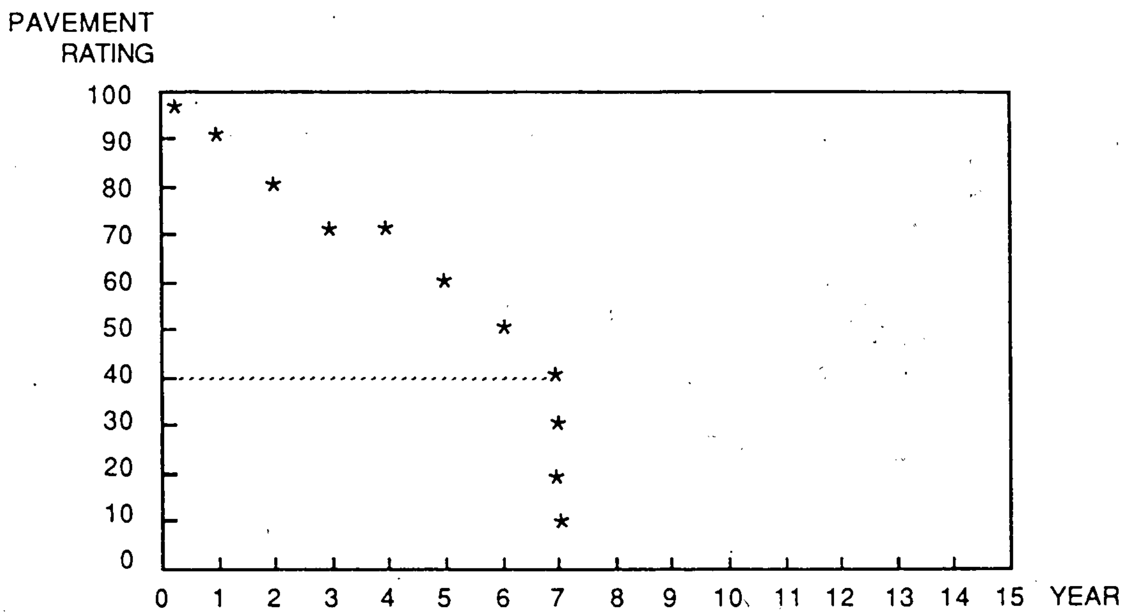
- Problem types amenable to solution by KBES;
- Description of the major components of a typical KBES, and how these components interact in a problem-solving situation;
- A step-by-step guide for building a KBES; and
- Identification of expected benefits of expert systems.

This is a useful document for anyone contemplating the initiation of KBES activity.

Other applications that FHWA has sponsored or is sponsoring are summarized in the following list:

Segment: SR - 530, SR - 5 to SR - 9

Detailed Analysis for: PRELEVEL OR MILL AND MEDIUM ASPHALT CONCRETE OVERLAY



Expected Service Life	=	7 years
Standard Deviation of Service Life	=	3 years
Desired Minimum Service Life	=	5 years
Probability of Exceeding Minimum Service Life	=	74%

Total Construction Cost	= $	242820.	(total segment)
Annualized Cost	= $	43498.	(total segment)
Annualized Cost	= $	13369.	(per lane-mile)

PERFORMANCE INDEX = 22.4

Current Values:

1. Segment title: SR - 530, SR - 5 to SR - 9
2. Length: 6442. ft
3. Width 32. ft
4. Existing WSDOT pavement rating 40.
5. Unit construction cost: $9.45 per square yard
6. Discount rate: 6.00 %
7. Performance curve form: Linear

FIGURE 2 Example application of SCEPTRE (from *17*, p. 139).

- ESAP. Asphalt Concrete Pavement Construction Guide is an expert system for advising inexperienced field inspectors on asphalt paving projects.
- ROAD HUMPS (speed bumps). This expert system is designed to assist highway engineers in the optimum design and application of road humps.
- PAMEX (Flexible Pavement Maintenance/Rehabilitation). This is an expert system that will assist highway engineers in the selection of appropriate pavement maintenance or rehabilitation strategy.
- Expert System for Signal Design at Isolated Intersections. This system assists in the complete design of detector placement, signal phasing and timing for an actuated signal at an isolated intersection.

- Freeway Incident Management Expert System. This system will assist authorities in mobilizing the appropriate responses after verifying the freeway incident and details.
- HWYCON. An expert system developed under the Strategic Highway Research Program (SHRP) for analyzing concrete durability.

(Additional information on these and other FHWA activities in the development of expert systems may be obtained from the Technology Assessment Branch, Office of Technology Applications, HTA 11, Federal Highway Administration, 400 7th Street, S.W., Washington, D.C. 20590.)

In addition to the transportation literature contributions already discussed, two other reports warrant mention. Both resulted from early efforts by Transport Canada to stimulate KBES activities in transportation. The first was the 1985 "Workshop on the Application of Expert Systems to Transportation," and resultant *Proceedings* (Transport Canada Report TP-7209E). This effort was similar to the OECD Workshop in that it overviewed the concepts of KBES, and then discussed specific application areas.

The second Transport Canada contribution occurred in 1986 with the publication of "Expert Systems: Their Application in the Canadian Transportation Sector" (Transport Canada Report TP-7328E). In addition to providing an overview of Canadian transport and KBES, this report included the following chapters of interest:

- Target Transportation Applications
- Available Literature of Expert Systems Applied to Transportation
- Market Trends in Expert Systems
- Available Tools for Implementing Expert Systems
- Current Canadian Expertise
- The Supply of Knowledge Engineers
- Strategies for Transport Canada
- Appendices: Tool Encyclopedia,Canadian Inventories, Foreign Research

As a follow-up to this successful effort, the Canadians scheduled another Conference on Expert Systems in Transportation in 1992.

(Transport Canada documents can be obtained from the Research Planning and Coordination Centre, Transport Canada, Place de Ville, Tower C, Ottowa, ON K1A 0N5.)

STATE TRANSPORTATION AGENCY APPLICATIONS

The extent of KBES activity completed and ongoing throughout the state transportation agencies was ascertained for this synthesis with the aid of two questionnaires, one distributed to the state transportation agencies, and the other to members of the TRB and ASCE Expert Systems Committees. Copies of the questionnaires are included in Appendix D.

Essentially, the questionnaires sought to determine if the agency has sponsored or performed any KBES work, and if so, the status of the work with regard to its being a useful production level tool. Each KBES project included by the respondent was to be assessed in its progress toward implementation in one of these categories:

- Conceptual,
- Prototype under development,
- Prototype developed and under testing,
- Detailed KBES under development,
- Finished KBES in use, or
- Project terminated.

In addition, the questionnaires asked the respondents to list up to five suggested KBES applications with exceptional potential.

In response to the questionnaire, 15 state transportation agencies indicated some level of KBES activity, and 20 state transportation agencies indicated no current or planned activity. Many in this latter group, however, suggested some ideas that could be appropriate for KBES. Sixteen state transportation agencies did not respond. Table 2 is a tabulation of the responses by the state transportation agencies with KBES experience. In the table, the numbers indicate that the state has had some KBES project at the stage of development the number represents, but the numerical sequence does not necessarily match the activity areas listed in the last column.

Summary of State Activities

Below is a state-by-state discussion of responses from those state transportation agencies that indicated previous or ongoing

TABLE 2
TABULATION OF QUESTIONNAIRE RESPONSES BY STATE TRANSPORTATION AGENCIES

State	Stage of Development	KBES Activity Areas
California	1,2,3,4,5	Hazardous materials; Traffic incident mgt.; Water quality; Concrete products; KBES priority
Connecticut	3,5	Pavement rating; Impact attenuator design
Illinois	1,3	KBES priority; Emergency response
Kansas	1,3	Concrete construction; Concrete pavements
Maryland	3	Freeway incident management
Minnesota	4	Processing truck permits
New Jersey	3,5	Noise barrier design Infrastructure risk management
New York	3,4,5,6	Snow problem location; Asphalt paving inspection; Pavement marking; Concrete analysis; Infrastructure risk management; Steel bridge inspection
Oklahoma	1	KBES state of the art
Oregon	3	Truck weight analysis
Pennsylvania	3,5	Automated bridge design/drafting Structural failure analysis
South Dakota	3	Processing truck permits
Texas	2,3,4	Bridge rail retrofit Constructability enhancement Pavement analysis
Utah	3	Construction evaluation
Virginia	5	Traffic control in work zones Disposition of old bridges

1= Conceptual 2= Prototype in development 3= Prototype under testing
4= Detailed KBES in development 5= KBES in use 6= Project terminated

work in KBES. The amount and type of information reported, including the stage of development for particular KBES, varies for each state, based on that state's response to the questionnaire.

California

The California Department of Transportation (Caltrans) has been a leader in the development and use of KBES, and in fact has four projects completed and in use. Two are in the hazardous materials area; these are SITE (Site Investigation and Training Expert Advisor) and TANKS (Tank Advisor and Knowledge Systems), both developed by Ritchie at the University of California-Irvine.

SITE assists Caltrans staff in initial site assessment and preliminary site investigation of suspected hazardous waste sites on proposed and existing highway projects. SITE also serves as a training tool for Caltrans personnel learning to deal with hazardous materials issues.

In the preliminary site investigation (PSI) area, SITE helps identify possible contaminants and features of the parcel in question. Based on the chemicals expected to be contained at each land use, SITE extracts relevant chemical testing methods from the Caltrans Statewide Hazardous Waste Management program. It then recommends field investigation strategies for confirming hypotheses relating land uses to features and expected contaminants. SITE develops a workplan that includes site features of concern, contaminants of concern, the number and locations of monitoring wells, bore holes, samples required, chemical tests required, and estimated costs.

TANKS generates an appropriate preliminary site investigation strategy to confirm the type and extent of contamination from underground fuel tanks. It considers fuel storage tanks, waste oil tanks, and piping associated with those tanks. Given basic information on the conditions surrounding a leak, TANKS assists the user in determining what action, if any, is needed, and provides a list of additional information items needed to adequately establish a preliminary site investigation.

Both SITE and TANKS are designed to operate on Apple Macintosh SE microcomputers with more than two megabytes of memory. They use HyperCard and NEXPERT OBJECT.

Caltrans has also developed 4RSCOPE. This is an integrated database management expert system to assist engineering staff in determining appropriate design features to be included in highway rehabilitation projects. It is an interactive package that runs under FoxBASE and EXSYS on a microcomputer platform.

Prototype KBES projects are under development by Caltrans in a variety of other areas. These include incident traffic management, encroachment permit processing, water quality, and concrete specialty products. A project related to disaster planning and management is under development using a Sun SPARCII workstation.

A long-range study for implementation of KBES technology throughout Caltrans has been commissioned. This study, completed in 1988, was jointly prepared by the Universities of Louisville and California-Irvine. It is discussed in detail in the next chapter.

Connecticut

The Connecticut DOT (ConnDOT) has been active on a variety of KBES projects. One project that is operational is the Pavement Rating and Analysis System (PRS). This is a network-level pavement condition rating tool that uses the Connecticut photolog laser videodisc (PLV) retrieval system. Closeups of photolog images are viewed singly and rated for cracking and other types of distress. A simulated driveover is also performed using normal forward-facing photo images. The resultant output provides both detailed listings of distress for each image and averaged information for the entire section including a score on a 1-100 scale. KBES elements are embedded in the PRS package, which executes on IBM-compatible 386-class microcomputers. This entire package was developed in-house by ConnDOT's Division of Research.

Also under development entirely within ConnDOT is a microcomputer-based decision tree program to determine recommended treatments for pavements with known condition scores and traffic volumes. This project is in the prototype development stage and is a joint effort by the Division of Research and the Office of Engineering (Pavement Management Section).

ConnDOT has sponsored work by Logie and Carney at Vanderbilt University to develop a KBES in the area of impact attenuator design. The result is CADS (Connecticut Attenuator Design System), which is in the prototype testing stage. CADS optimizes the design of a site-specific crash cushion when supplied with basic information concerning the dimension of the site and design speed of the roadway. CADS incorporates the crash testing guidelines and performance requirements of *NCHRP Report 230, Recommended Procedures for the Safety Performance Evaluation of Highway Appurtenances (20)*, along with an accurate mathematical model of the vehicular and occupant impact responses. The implementation language is Turbo Pascal using an object-oriented programming approach.

Illinois

Several KBES projects have been initiated by the Illinois Department of Transportation (IDOT). These include participation in an Illinois Agency Panel Study assessing the potential for KBES use throughout Illinois state agencies. The IDOT Bureau of Information Processing is responsible for the study, which is a background research effort using a survey form similar to questionnaires distributed as part of this synthesis.

In addition, IDOT has a prototype KBES in the testing phase for use by the department's Communications Center in emergency response situations. IDOT has another KBES project which is in the conceptual stage. This project will focus on IDOT's bridge management system, and will attempt to relate it to a wide spectrum of administrative policies from preventive maintenance to bridge replacement. Ultimately, IDOT envisions that a new knowledge based comprehensive bridge management system will result from this project.

Kansas

Kansas DOT (KDOT) is sponsoring two KBES research projects and anticipates more. The first is entitled "Prototype Expert System For Resolution of Concrete Construction Problems," conducted by Professors Fan, Huang, and Russell of Kansas State University. The goal of this PC-based system is to identify and classify pavement construction problems on structures, such as plastic shrinkage cracking, thermal cracking, low cylinder strength, rough riding surface, spalling of sawed concrete, rained-on concrete, honeycombing, cold joints, incomplete consolidation, and poor finishing. The knowledge base for these problems includes several ACI Standards, the ACI *Manual of Concrete Inspection*, and the KDOT *Standard Specifications*.

The second KDOT project is an extension of the first to include slabs and pavements. It is being performed by the same team from Kansas State University.

Maryland

Maryland DOT is developing a prototype KBES in its Office of Traffic. This is a KBES for Freeway Incident Management, and is a variation of the MIST system. It is intended to analyze traffic on all routes in the vicinity of an incident and manage traffic diversion and flow until the incident is cleared.

Minnesota

Minnesota DOT (MnDOT) has contracted to develop the ROUTEBUILDER KBES. This project, which began in 1988, will find and assign truck routes within the state of Minnesota and will assist staff in processing over-dimensional permits. The system will guide the administrator through the process of collecting the necessary information, find an appropriate legal route between two points for the load in question, and automatically issue the permit to the trucking company, either in hard copy, or via teletype. The fee for each permit will no longer be based on the straight-line route between the two points; rather, it will be based on actual mileage of the route, thus capturing additional revenue.

The system will run on a PC LAN (local area network), and will incorporate MnDOT's road system information housed on mainframe computers. The hardware includes a file server and a series of remote workstations.

New Jersey

New Jersey DOT (NJDOT) has one KBES project that is completed and operational, and another that is in the prototype testing stage. The operational KBES is CHINA (Computerized HIghway Noise Analyst), which is a noise barrier design tool. CHINA, developed for NJDOT by Harris and Cohn of the University of Louisville, is discussed in more detail in Appendix B.

The prototype KBES project in NJDOT is entitled "Infrastructure Risk Management," and is a research project jointly sponsored by the New York State Department of Transportation and the Region II University Transportation Research Consortium (UTRC). The goal of this project is to develop and demonstrate an integrated mathematical programming/KBES approach for use in prioritizing bridge maintenance and rehabilitation. The development medium is Rule Master, by the Radian Corporation. This project is discussed in more detail in the section on New York.

New York

Another agency that has been a leader in KBES activity is the New York State Department of Transportation (NYSDOT). One project is operational, three are under development as part of a regional university research effort, and two are ongoing in-house.

The operational project is a KBES that evaluates and analyzes identified snow problem locations. It helps agency personnel to determine the nature and magnitude of the problem, and to identify possible solutions, without requiring extensive knowledge of passive snow control methods. The problem domain is limited to blowing (whiteout) and drifting snow as it affects highways. Essentially, the program examines drift potential and predicts the effects of road redesign or snow fence installation. The programming environment is the GURU shell from Micro Database Systems, and the platform is IBM-compatible microcomputers. Several support programs are written in BASIC.

The two projects underway internally within NYSDOT are in the implementation stage (detailed development). The first, by Williams of Rutgers University and the NYSDOT Construction Division, provides an interactive reference manual for inexperienced paving inspectors. The database contains standard paving specifications, reference documents, and articles of expert advice. The prototype system was constructed for an IBM-PC 286-class platform using the Hyperties hypertext system.

The other internal KBES is a pavement marking policy advisor project, developed by D. F. Kaminski of the NYSDOT Buffalo Regional Design Group. It provides guidance on the proper application of department policy regarding short-term, temporary, and final pavement markings. The rules that make up the knowledge base were taken from various NYSDOT policy manuals and engineering instruction documents. Production Rule Language (PRL) and the LEVEL5 shell serve as the programming environment, with the platform being MS-DOS microcomputers.

The three projects underway through the University Consortium are in various stages of prototype development. The KBES known as CONCEX (for CONCrete EXpert) is under development by Williams and Balaguru at Rutgers University. The capabilities of CONCEX include:

- Calculating compressive strength for a given concrete formulation;
- Estimating slump, air content, and 28-day compressive strength from known quantities of constituent materials;
- Diagnosing slump and air content variation of fresh concrete at a construction site;
- Predicting 28-day compressive strength from accelerated strength tests; and
- Predicting compressive strength at various stages.

CONCEX is written with a combination of the shell Rule Master 2 and FORTRAN 77. The platform is an IBM-compatible microcomputer.

Vanmarcke and Santamarina of Princeton and Polytechnic Universities, respectively, are developing a KBES project on infrastructure risk management in which they are prioritizing bridges for maintenance and rehabilitation. This system, a joint project between NYSDOT and NJDOT, combines mathematical programming with expert systems. The mathematical programming is in the form of powerful risk analysis software packages to express uncertainty in deterioration estimation. The integer programming model was created using GAMS (General Algebraic Modeling Systems) by Meeraus.

The other NYSDOT-related project underway through the UTRC program is intended to develop a KBES for steel bridge superstructure inspection and evaluation. Chen, of the State University of New York at Buffalo, is performing this work, in cooperation with the NYSDOT Engineering Research and Development Bureau.

Oklahoma

While it does not have a specific application KBES project completed or underway, the Oklahoma DOT is sponsoring a state-of-the-art study of KBES. The effort is being conducted by the University of Oklahoma School of Civil Engineering, with the goal to recommend specific applications of KBES suitable for the department.

Oregon

Oregon DOT (ODOT) is sponsoring a KBES project for truck weight analysis and vehicle routing in the permits area. The project is in the prototype testing phase. The KBES is configured to run on a PC LAN (local area network), and uses Structured Query Language (SQL) to communicate with any ODOT databases. This project and configuration are quite similar to the one underway for the Minnesota DOT.

Pennsylvania

Pennsylvania DOT (PennDOT) has been involved in KBES activities since the mid 1980s, when it developed BRADD-2, the "Bridge Automated Design and Drafting" system. This system designs the number and size of beams, relying on knowledge based experience, to provide the most economical bridge superstructure for given constraints.

PennDOT is also developing a KBES to detect incipient failure in bridge structural members. Heuristic evaluation will determine if a hidden failure or crack has occurred. This system is in the prototype testing phase, with additional research and funding needed to move it into the operational phase.

South Dakota

South Dakota DOT has a KBES project to develop an intelligent advisor for one-stop processing of truck permits. The goal of this project is to assist the state employee in giving guidance to trucking company personnel. The guidance to be given relates to determining the legal requirements needed to cross the state of South Dakota. The system, which is in the prototype development stage, will run on a PC LAN using OS/2 level relational databases.

Texas

The Texas Department of Transportation (TxDOT) has sponsored several research contracts in the KBES area. These projects have been conducted by transportation research groups at the University of Texas (Center for Transportation Research, CTR) and Texas A & M University (Texas Transportation Institute, TTI).

Roschke, of TTI, has a detailed KBES under development for bridge rail retrofit analysis and design. This project will incorporate judgement, intuition, experience, and other expertise of recognized bridge engineers to emulate human problem solving. System modules include a database containing codified specifications, crash test data, rail dimensions, cost-benefit evaluation of alternative designs, and structural analysis.

This KBES platform is a 386-level microcomputer. It uses Microsoft C, FORTRAN, WINDOWS, and NEXPERT OBJECT. A graphics component allows the user to display rail drawings from the bridge rail database to view physical appearance.

Another interesting KBES project is in the prototype development stage. This is a constructibility enhancement program by Hugo et al. of CTR. (Constructibility enhancement means improving those characteristics of a project that make it easier and more efficient to build). The logic planning for the detailed KBES has been completed. Final development will be on a microcomputer using the GUIDE program and HYPERTEST.

As part of this project, several tools were developed, including the Highway Constructibility Guide and the Hierarchical Objectives Diagramming Technique, which was used to document constructibility enhancement strategies. Efforts are underway to organize these and other tools into an improved constructibility communication network. The KBES portion of the project lies in the development of a user-friendly knowledge base to help access the large number of strategies and tactics identified in the process.

A project by Scullion of TTI is nearing completion. This project has produced Micro-PES, an expert system for analyzing information gathered by TxDOT engineers in their annual network-level Pavement Analysis Evaluation System (PES). This KBES is executable on an IBM PC-XT or higher level microcomputer. Included in Micro-PES are three subsystems:

- A one-year Maintenance and Rehabilitation (M&R) estimation system, which contains a series of decision trees developed by experienced engineers from the TxDOT Maintenance and Pavement Design Divisions. These trees relate pavement distress to appropriate M&R strategies for flexible, jointed, and continuously reinforced concrete pavements.
- The RAMS-district optimization program, which selects an optimum combination of projects within a fixed budget level. The optimization uses 0-1 integer programming techniques and uses maintenance effectiveness as an objective function.
- A routine maintenance estimate system which permits estimates of type, amount, and cost of routine maintenance requirements for any highway or network of highways.

Utah

The Utah DOT has a KBES in the prototype testing phase. The project is for the development of a prioritizing process for evaluating construction projects. The scope of the KBES will include the entire spectrum from small maintenance activities to full-scale major construction projects.

Virginia

Two KBES projects have been completed by the Virginia DOT (VDOT). The first produced TRANZ, a prototype KBES for traffic control in highway work zones. This work was performed by Demetsky and Faghri of the Virginia Transportation Research Council (VTRC), and included a background volume overviewing the applicability of KBES to transportation engineering. Other volumes published on TRANZ address demonstration and evaluation. Work on the TRANZ prototype was completed in 1990, and is available for use by VDOT traffic engineers.

The problem addressed by TRANZ requires the selection of appropriate traffic control measures and management strategies for the protection of the freeway user and the work force, as well as for the movement of maximum traffic volume around work zones on limited access, primary, and secondary highways. Selection of these measures is usually made by a supervising engineer based on knowledge of options in the *Manual of Uniform Traffic Control Devices* (MUTCD) and the *Virginia Work Area Protection Manual*. The goal of TRANZ is to capture the heuristic knowledge and experience of the supervising engineer, as well as the more quantitative information contained in the MUTCD and the Virginia Manual. When tested for actual work zone problems in Virginia during the evaluation phase, TRANZ performed well every time.

TRANZ is written for the IBM-compatible PC environment using the EXSYS shell. For comparative purposes, an alternative version was prepared using the LISP programming environment.

The other KBES project sponsored by VDOT produced DOBES (Disposition of Old Bridges Expert System). This work was performed by Zuk and Newburgh of the VTRC, and was initiated by a 1986 study overviewing the potential for KBES application to bridges and pavements. DOBES provides recommendations as to whether a bridge should be rehabilitated, improved, replaced, abandoned, or simply maintained. The recommendations are based on a wide range of attributes and weighting factors, and are balanced against relative cost of each action. DOBES is written in LISP and can be used on any IBM-PC compatible microcomputer.

OTHER KBES PROJECTS

The questionnaire that was distributed to members of the TRB and ASCE Expert Systems committees brought forth information on additional KBES projects that are underway. A sampling of the more significant are:

FASTBRID—a KBES for bridge fatigue, developed by Melhem and Wentworth of the FHWA Turner-Fairbank Highway Research Center. This prototype helps the user organize fatigue inspection, evaluate the inspection results, and recommend corrective measures. FASTBRID uses the EXSYS shell for IBM-compatible microcomputers.

FRED—a KBES for freeway incident management, developed by Ritchie and Prosser of the University of California-Irvine. This prototype is a real-time expert system for managing non-recurring congestion on urban freeways in Southern California. FRED has been applied to a section of the Riverside Freeway in Orange County. FRED uses G2, a real-time KBES development software package, with external functions written in C. The hardware platform is a Sun SPARCI workstation, a RISC-based Unix machine.

ROSE—a KBES for routing and sealing asphalt pavements, developed by Hajek of the Ontario Ministry of Transportation and Communications (MTC). ROSE is based on the MTC pavement monitoring and evaluation procedures. It interacts with the existing MTC pavement management information data bank, and contains within its knowledge base the decision logic for when to rout and seal. ROSE is a completed project that is operational and in use by MTC staff. The development tool is EXSYS, and the platform is IBM-PC compatible microcomputers.

OVERLAY—a KBES for advising pavement design engineers about suitable asphalt concrete rehabilitation strategies, also developed by Hajek of MTC. The knowledge base for OVERLAY was obtained by interviewing the 17 pavement design and evaluation professionals who are primarily responsible for Ontario's pavement rehabilitation strategies. The results of this effort show that it is possible to organize and manipulate the knowledge base to produce recommendations that reflect the policies of a large agency. OVERLAY is also developed with EXSYS for IBM-PC compatible microcomputers. It is in the detailed development stage.

SEG—a KBES to help determine causes for random segregation in asphalt pavement, developed by Elton of Auburn University. This system is in use by the Kansas Department of Transportation. The development tool for SEG is the shell INSIGHT 2+ (now called LEVEL5), and the platform is IBM-PC compatible microcomputers.

CLAES—a KBES to implement the Construction Lien Act in Ontario, developed by the Transportation Technology and Energy (TTE) Branch of the Ontario MTC. CLAES assists with the administration of highway contracts and is used by MTC regions and districts. Another TTE KBES is Med Rev, a prototype KBES to assist with the review of drivers licenses where medical issues are involved. Development tools for these KBES projects include PROLOG, C, LEVEL5, Pascal, and DOS WINDOWS. They are designed to execute on 286-level IBM-PC compatible microcomputers.

Production-Stage KBES Applications

As shown in the preceding discussions, there have been many efforts made toward developing useful KBES products. Most of these are in the prototype stage, with few being in the production stage, or completed and available for use. Two in the production stage, PARADIGM and CHINA, are discussed in detail in Appendix B, in the hope that this detailed level of discussion will provide some useful insight into the specifics of KBES development.

CHAPTER THREE

POTENTIAL APPLICATIONS OF KBES IN TRANSPORTATION

State transportation agencies desiring to develop KBES projects have a wide range of possible problem areas from which to choose. Some state transportation agencies, most notably Caltrans, have conducted strategic studies to provide insight on priority setting. The Caltrans study is discussed below, followed by a summary of questionnaire responses on potential applications.

CALTRANS STRATEGIC STUDY

In June 1987, Caltrans executed a contract with the University of California's Institute of Transportation Studies (ITS) to identify and prioritize potential application areas for KBES development within the department. The work specified in the contract was performed by Ritchie at the University of California-Irvine, and by Cohn and Harris at the University of Louisville.

Among the tasks accomplished was a review of Caltrans' organizational characteristics and operating procedures. Also, a series of in-depth interviews was conducted with a group of Caltrans managers and engineers. These interviews were held during the week of September 14-18, 1987, in Sacramento. Each interview was scheduled for one hour, and was based on the interviewee's preliminary responses to a previously completed questionnaire.

The contractors also developed criteria which could be used to assist in prioritizing the candidate projects. In addition, a category system to further aid in directing Caltrans implementation efforts was developed (*21*). Each is discussed below.

Criteria for Determining Priorities—Three criteria were used to determine the relative implementation priority for each candidate KBES project. The criteria were used in a qualitative sense by the contractors, to determine the final recommended priorities. The criteria are:

- Potential Savings to Caltrans—Some expert systems, if properly implemented, could result in savings of significant funds as a result of reduced manpower needs, better use of materials, and so on. Those candidate projects with the highest potential for savings, as judged by the contractors, received the highest priority in this category.
- Potential for Straightforward Implementation—Those candidate KBES projects that were assessed as easiest to implement received the highest priority in this category. Among the factors used in this assessment were clarity of the problem definition and solution options, cohesiveness of the knowledge base, and the availability of expertise.
- Potential for Active Support in Caltrans—For a KBES project to be successfully implemented, it is important that there be decision makers and experts within Caltrans who understand the nature of KBES development, and who are enthusiastic about the project.

Assignment of Priorities for Implementation—The three criteria defined above were used to place each candidate KBES project in one of four categories determining a sequence of implementation. Only those projects with the very highest potential for savings, straightforward implementation, and Caltrans support were placed in Category 0. These are the projects that Caltrans was recommended to implement immediately, as a means to demonstrate the great utility of KBES. This category was limited to the two most significant projects identified in the study: Disaster Planning and Management, and Hazardous Waste Site Characterization.

The Category 1 projects have potential payoffs significant enough that their development was recommended as soon as possible (ideally within a year). Projects in categories 2 and 3 merited implementation, but it was suggested that they be developed by Caltrans personnel who were proficient in KBES development through the experience gained in assisting consultants with the implementation of the Category 0 and 1 projects. The following list rates projects by priority.

LISTING OF CALTRANS CANDIDATE PROJECTS BY PRIORITY CATEGORY

Category 0 Recommended for Immediate Implementation

Hazardous Waste Site Characterization
Disaster Planning and Management

Category 1 Recommended for Implementation Within One Year

Pavement Rehabilitation Project Development
Design Standards Exceptions Advisor
Hazardous Waste Mitigation Options Advisor
Incident Traffic Management
Highway Planning Project Design Advisor
Assessing Effectiveness of Traffic Mitigation Strategy

Category 2 Recommended for Implementation After One Year

Route Location Study Advisor
Route Concept Report Advisor
Sections 16(b) and 18 Advisor
Regional Transportation Planning Evaluation Advisor
Financial Data and Trend Interpreter
Transportation Permit Advisor
Encroachment Permit Advisor
Safety Hardware Advisor
Hazardous Waste Site Evaluation Advisor
Leaking Underground Fuel Tank Advisor
Security Analysis Advisor
Revegetation/Erosion Control PS&E Advisor

Visually Assessing Highway Projects Advisor
Bid Pattern Interpreter
ROW/Utilities Interaction Advisor
Transit Capital Improvement Project Ranking Advisor
Accident Analysis Advisor
Hydrologic Analysis Advisor
Hydraulic Analysis Advisor
Water Management Advisor
Vegetation Control Advisor
Railroad Relocation Advisor
Category 3 Recommended for Future Implementation
Scenic Resource Evaluation Advisor
STIP/Obligation Plan Development Advisor
Technology Transfer to Local Agencies
Equipment Repair Advisor
Software Selection Advisor
Traffic Operations Center Advisor
"Landscaped Freeway" Status Advisor
Incident Response Advisor
Traffic Signal Operations Advisor
Transit Network and Operation Planning Advisor
Impact Assessment Advisor
Signal Timing Advisor
Utility Policy and Procedures Advisor
Concept Development Advisor
Environmental Planning Advisor

SUMMARY OF QUESTIONNAIRE RESULTS

The questionnaire, sent to state transportation officials in all states, asked each respondent to list up to five problems in transportation that had exceptional potential for KBES application. The responses are summarized in Table 3.

It is obvious from the wide array of applications suggested by the state transportation agencies that KBES has potential in many areas. "Hard side" issues, such as bridge design and pavement analysis, have been and will continue to be considered especially appropriate for expert systems. In addition, real-time operational transportation activities, such as incident management and traffic signal system design, are also viewed as good candidate applications.

In general, the questionnaire responses show that the state transportation agencies have an understanding of KBES and the types of problems that can be addressed. This is an indication that more KBES activity throughout the country is likely as successful applications are implemented.

TABLE 3
POTENTIAL KBES PROJECT AREAS SUGGESTED BY STATE TRANSPORTATION AGENCIES

Topic Area	Recommending State
Bridges	
Structural steel fabrication problem resolution	Kansas
Bridge evaluation and management	Arizona Minnesota New Hampshire Pennsylvania Tennessee
Bridge inspection	New York
Bridge design	Oklahoma Oregon South Dakota Tennessee Wyoming
Bridge painting alternatives	Michigan
Estimation of bridge deck life	Michigan
Hydraulics	Missouri Tennessee
Seismic design standards and guides	South Carolina
Construction	
Construction problem resolution	Kansas
Estimating and scheduling	Arizona Illinois Oklahoma Utah
Geotechnical investigations	Connecticut Kansas Idaho Utah
Highway materials testing	Kansas
Bid analysis (fraud detection)	Maryland
Jobsite problem analysis	Maryland
Pile driving	Nevada
Concrete mix design	New York
Environment and hazardous materials	
Environmental impact assessment	Arizona Pennsylvania
Community participation	Pennsylvania
Hazardous material advisor	Illinois
Routing of hazardous material haulers	Arizona

TABLE 3 *continued*

Topic Area	Recommending State
Maintenance and management	
Maintenance strategies	Oklahoma
Equipment maintenance	Arizona Oklahoma
Traffic barrier system design	New York
English language access to database	Kansas
Project coordination/communication	Minnesota Missouri
Training	Minnesota
Manpower planning	Oregon
Project selection	Illinois
Needs assessment and priority rating	Wyoming
Highway improvement program management	Wyoming
Preprocessors for complex computer programs	Virginia
Pavements and roadways	
Pavement rehabilitation	Alabama Idaho Maryland Virginia
Pavement design	Connecticut Michigan Mississippi Missouri Oklahoma Oregon Nevada Tennessee
Pavement evaluation and management	Arizona New Hampshire
Roadway design standards	Minnesota New Hampshire Utah
Weight enforcement	Oklahoma
Intersection analysis and design	Connecticut
Traffic and transit	
Traffic network management	Maryland Minnesota
Real time traffic control	Virginia
Emergency response/incident management	Illinois Virginia
Reducing tort liability in traffic accidents	Virginia
Traffic projection	Mississippi Missouri
Quality assessment of traffic data	North Dakota

CHAPTER FOUR

GETTING STARTED WITH KBES

A variety of issues should be considered by an agency when getting started with KBES, including organizational issues and KBES project selection.

The Caltrans study produced three criteria that could be used to determine which potential KBES projects should receive the highest priority. These were:

- Potential for savings to the agency,
- Potential for straightforward implementation, and
- Potential for active support in the agency.

These three criteria are clearly applicable for any state transportation agency. A KBES project that will not result in improved efficiency, performance, or other tangible savings measure for the state transportation agency is not a good investment of resources. The project should also have a good likelihood for success (straightforward implementation) and should be enthusiastically supported within the appropriate state transportation agency functional unit. Like any new technology, an unacceptably high failure rate on early projects may cause a termination of the entire KBES effort within the agency.

STAFFING CONCERNS AND REQUIREMENTS

The challenge in programming at this level is often in the development of an acceptable user interface; however, software to aid in KBES development is abundant. Shells are available that go so far as to be "fill in the blank." The most demanding part of KBES development, given the software tools on the market, is usually taken to be knowledge acquisition.

The effective transportation KBES builder, then, is one who is comfortable with computers and programming, but not necessarily a computer expert. That person must have familiarity with the transportation field, especially in the specific domain of the KBES. In transportation, most KBES projects have been undertaken by civil engineers, either on university faculties or on the staff of a state transportation agency. There may not always be the need to retain the services of a specialist "knowledge engineer" for the type of application that typifies KBES projects in transportation. Civil engineers with modern educational experiences and a working knowledge of computers are ideally suited for KBES activity. Naturally, these civil engineers should be motivated toward KBES development, and should also be comfortable with human interaction, since knowledge acquisition is a vital part of the process.

However, when civil engineers perform the function of the knowledge engineer/programmer, the final product (operational KBES) may be lacking in program sophistication. The best circumstance is to engage experienced computer professionals, such as those found in the computer services functional areas of most state transportation agencies, when they are available. By combining technical domain experts with programming specialists, the best KBES projects are likely to be produced.

Few KBES projects in transportation have made it to the production stage. Based on discussions with the developers of those that have, it appears that a one to three person-year level of effort is needed to reach that stage. Typically, an operational expert system in transportation will cost between $75,000 and $150,000 (in 1992 costs) for software development.

MAINTENANCE AND SUPPORT

Like any computer program, an expert system will require knowledgeable people to keep it current, to provide advice on its use, and to address any possible inconsistencies it may generate. However, expert systems are easier to modify due to their natural languages and the separation of the rule base from the inference engine (development tool). This separation makes modifications and updates more straightforward, with fewer chances for inadvertently introducing errors into the programming logic.

A state transportation agency with a major commitment to KBES should arrange for in-house expertise and understanding of AI and KBES. This expertise may best be located within the computing function of the agency.

HARDWARE AND SOFTWARE ISSUES

Hardware

A basic issue to consider is the hardware environment in which the program will be used. This is referred to as the delivery environment, or platform, for the system. The possibilities include LISP machines, mainframes, personal computers, and workstations (*22*). The most common platform in state transportation agencies is the IBM-compatible microcomputer.

Another issue concerns how large the system will be. This is usually measured by the number of rules in the knowledge base. A small system will contain up to 500 rules; from 500 to 1,500 rules is considered medium-sized; and more than 1,500 rules is a large system (*23*).

The need for a system to interact with the host operating system, external databases, or spreadsheets can also be an important issue. If the system will call or be called by other applications, this must also be planned and accommodated. In addition, execution speed, integration with traditional information processing systems, and input/output must also be considered.

The level of computer sophistication of the KBES designer will determine how user-friendly the environment must be. If graphics will be required, this should be planned from both the hardware and software viewpoints.

Finally, costs must be considered. Costs of acquiring the development environment are relative to the program's size and complexity. They may include a site license, any needed runtime licenses, training of agency personnel, and continuing support of the system once it has been purchased. The need for hardware or software support is primarily determined by the qualifications of the user and the complexity of the particular product.

Software

Evaluation of a particular system includes both objective and subjective measurements. The following features and characteristics may be considered: overall power and flexibility, knowledge entry interface, user interface, runtime speed, memory requirements, and training and support.

One aspect of power and flexibility is found in the way facts are represented. One possibility is attribute-value pairs, called A-V pairs, such as color (attribute) is red (value). An example of an object-attribute-value triplet, called O-A-V, would be the box's (object) size (attribute) is large (value). Facts may be represented as true or false or as a mathematical relation. Some tools allow the use of symbolic variables, higher mathematical functions, and use of numeric variables.

Relationships between facts may be represented by IF-THEN rules. Some tools allow an OR to be used between rules. Special rules may be written to handle the arithmetic operations or to work with external subroutines.

Inference and control could be handled by forward or backward chaining. More than one goal could be established, as well as having confidence or certainty factors.

The knowledge engineering interface deals with the actual creation of the knowledge base. The length of time required to complete the knowledge base is partially dependent on learning the individual characteristics of the particular programming environment (editor, compiler, etc.).

Runtime speed is an important criterion for any environment. This can be determined either by timing the system's response or from the manufacturer's literature.

The documentation provided with the environment and the support offered by the manufacturer are important criteria which are measured subjectively.

CHAPTER FIVE

KBES ARCHITECTURE

A typical expert system is made up of four primary components (*1*). These components include:

- A knowledge base containing the domain-specific facts and heuristics associated with a particular field;
- A rule interpreter, or inference engine, that can use the knowledge base to solve a domain-specific problem;
- A context, or a global database or work space that maintains the problem status, the input data, and the relevant history of the actions taken by the system on the current problem; and,
- A user interface.

THE KNOWLEDGE BASE

The knowledge base contains the facts and rules that represent the expert's knowledge. Much of the knowledge in expert systems is heuristic, that is, rules of thumb or simplifications that effectively limit the search for a solution (*2*). The heuristics are mostly private, little-discussed rules of good judgment that characterize expert-level decision making in the field (*3*). Facts and rules are used by the expert system to emulate the reasoning of an expert and make an intelligent decision. It is therefore necessary that the system embrace a great deal of knowledge about the problem domain. Another important note to make here is that since the knowledge is constantly growing and maturing, the knowledge base must be readily understandable so that revisions can be made easily (*17*). Knowledge can be gathered from a multitude of sources, such as interviews with experts, and examinations of manuals, research reports, and textbooks. Once collected, the knowledge should be organized and represented in a way that facilitates decision making.

The method of constructing and ordering knowledge pertaining to a specific problem in order to solve that problem is referred to as knowledge representation. The expert system profits from knowledge representation by making the problem easier to understand, easier to solve, and easier to revise. Knowledge can be represented in the knowledge base in several ways (*24*). The most widely accepted method for incorporating knowledge into the knowledge base is through rules. Other methods include frames, objects, and semantic networks (see Glossary).

Rule-based knowledge representation is the most popular method of representing domain-specific knowledge due to its simplicity. In this case, the domain knowledge is represented as a set of rules (*2*). Each rule is made up of a condition part and an action part: IF (condition) THEN (action). A simple rule for analyzing rural highway accidents might be formulated as follows:

IF the predominant characteristic of the accident site is moisture on the pavement
and skid number is known
and skid number is less than or equal to 30
THEN the probable cause of the accident is poor skid resistance.

The condition part of each rule is tested against the evidence provided by the user, as well as the facts and knowledge pertaining to the problem being considered (*2*). At any given point, the condition part may or may not be satisfied. The action part will be executed only when the condition part is proven to be true, and it will be added to what is known. The result of an action part may change the set of facts in the knowledge base. Any new facts added to the knowledge base may be used to form matches with the condition part of the rules (*2*).

One of the principal advantages of using a rule-based knowledge representation method in an expert system is that this method simulates the way many experts use the knowledge in different problem domains. As a result, using the rule-based knowledge representation method assists the expert system's designer in capturing and representing the relevant details of a problem domain.

INFERENCE ENGINE

For the expert system to be able to apply its knowledge to a particular problem, it must have some mechanism for inferring conclusions from particular conditions. The inference engine is the component of an expert system that manipulates the knowledge contained in the knowledge base to solve the problem at hand. It performs two principal tasks. First, it examines existing facts and rules, adding new facts when possible. Second, it determines in what order the inferences are made (*25*). Several different methods can be used for applying rules. The two methods commonly employed by the knowledge based expert system are forward chaining (data driven) and backward chaining (goal driven).

In the first approach, the condition portions of the rules are checked against available facts to determine whether or not they are true. All the rules whose condition parts are proven to be true are fired (executed). As a result of this execution, new facts are created and added to the body of known and validated facts. The inference engine then re-evaluates the rules. It continues this process in an effort to deduce a solution to the problem at hand (*13*). This procedure is especially helpful when many hypotheses (solutions) exist and there are few available data. Its principal shortcoming is the ineffectiveness of mandating as input all potential facts for all conditions. At times, all potential facts may not be known or pertinent.

In backward chaining, the inference engine begins the search with a goal, or hypothesis, and works backward through the rules in an effort to evaluate whether or not all the data support the goal or hypothesis being considered (*13*). An alternative goal or hypothesis is examined when the initial goal or hypothesis is not proven to be consistent with the known facts. Backward chaining is advantageous if the number of possible solutions is limited.

The inference mechanism may also have the capability to report to the user how rules are applied and how conclusions are reached. The user can display the chain of rules used to reach the resulting conclusion. Examining this chain of rules can help the user understand the logic employed in applying the rules.

Some inference engines provide the capability to look at facts that are either supplied by the user or deduced by the expert system. The user can list the facts and the associated characteristics. Other capabilities might include listing the rules, listing similar rules, and listing numeric values used in the expert system.

The concept of certainty factors is also used in some inference engines. Certainty factors are those values that assign a numeric value to the probability that a given statement is true. Certainty factors can range from zero to one depending on the amount of certainty. For example, if a certainty factor of one is given to a statement, then the statement has a 100 percent probability that it is true. On the other hand, if a statement has only a 50-50 chance of being true, then a certainty factor of 0.5 would be assigned.

SYSTEM WORK AREA (CONTEXT)

The system work area or context is a space containing information describing the problem under study. Typically, information in the context includes the problem data, the solution status, and the action history. The problem data may be further classified as user input data, and data derived through the application of rules.

USER INTERFACE

The user interface acts as a vehicle through which the user and the expert system communicate with each other. Due to the fact that many users of expert systems are not programmers, the interface should be user-friendly to facilitate a direct and effective communication link. Although not standardized, the user interface usually contains two standard components: an explanation module and a knowledge acquisition module. The explanation module permits the user to ask about the reasoning of an expert system or the method used to infer results. The explanation module reveals the chain of rules used to obtain the results in response to the inquiry made. Because of the knowledge acquisition module, the domain expert does not have to fully understand the format of the knowledge representation. This module permits an expert to enter knowledge or rules in a limited natural language syntax. The module then translates the rules into a representation which the inference engine can manipulate. The expert is able to verify that this representation is correct during validation.

EXPERT SYSTEM DEVELOPMENT

There are five primary steps normally associated with the knowledge acquisition process or building of an expert system (Figure 3). The steps outlined below demonstrate the sequence traditionally followed; however, the actual steps employed will vary depending on the characteristics of the problem, the selected objectives, and available development tools (*2*).

Identification

The first step in developing an expert system is to clearly identify the features of the problem to be addressed. This involves specifying the problem characteristics, available resources, and goals. The intention is to segregate the knowledge pertinent to solving the problem and to identify major terms and attributes of the problem. At this stage, several iterations of potential solutions are necessary to assure that the available resources are adequate for the problem at hand. Among the needed resources are time, personnel (domain and knowledge engineers), hardware, and funds. Recognizing the goals to be accomplished is essential at this stage. Goals vary depending on the characteristics of the problem.

Conceptualization

The key concepts and relations identified throughout the previous step are made explicit during the conceptualization step. On developing the key concepts and relations, effort should be made to implement an initial system, or preliminary prototype KBES. This step mandates constant communication between the knowledge engineer (system developer) and the domain expert and is usually very time consuming.

Formalization

Terms, key concepts, and relations relevant to the problem domain are used to determine how the problem can be represented in a formal way within the selected tool or framework. Extensive communication between the knowledge engineer and the expert is required to formulate a detailed design of the system.

Implementation

The formalized knowledge from the previous step is employed to develop a prototype expert system using the previously selected tool (programming language, shell, etc.). Because one reason for implementing the initial prototype is to determine how effective its design is, implementation should advance quickly. It is very likely that the initial code will be modified or abandoned based on the results of the prototype.

Testing

The purpose of this stage is to evaluate the prototype system. The system should be tested with a wide range of problems to determine its deficiencies. Based on its performance on example problems, revisions are made by the knowledge engineer so that the system conforms to the standards set by the domain experts.

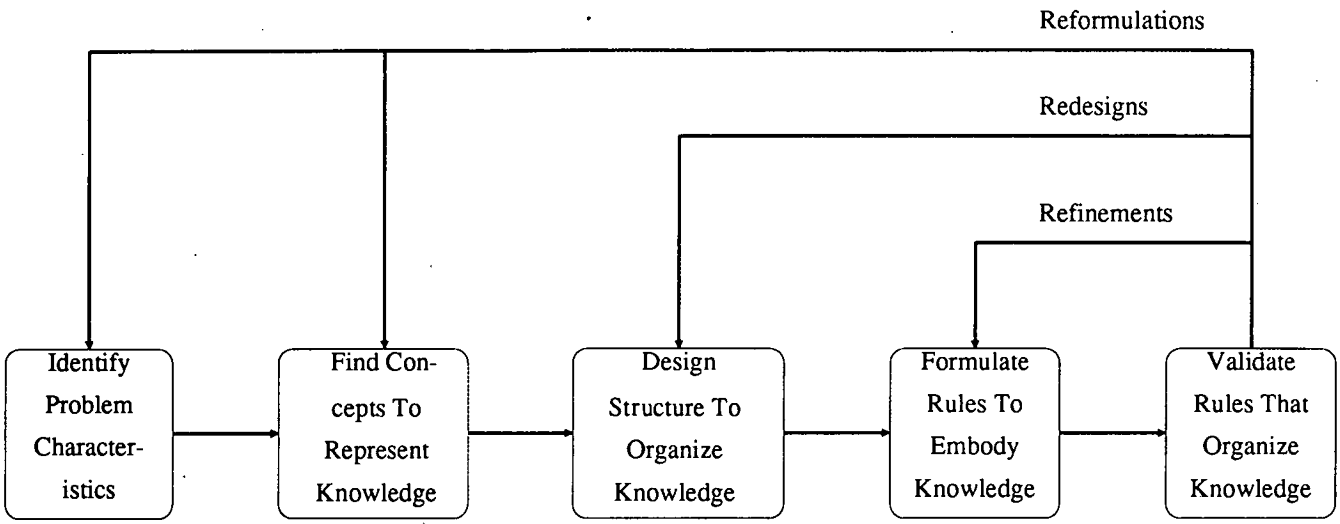


FIGURE 3 Steps in building an expert system.

IMPLEMENTATION ISSUES

The major shortcoming of KBES activities to date has been the inability of developers to bring operational systems on-line. The issues related to implementation include those described below.

Knowledge Acquisition

A key issue in the development of an expert system is knowledge acquisition. Knowledge acquisition may be defined as the process of obtaining problem-solving expertise from some source and transforming it into a computer program (*1*). Numerous sources of knowledge are available and include human experts, textbooks, journal articles, and personal experience.

Conceptually, the simplest way to put knowledge into a program is through direct coding by the domain expert. In fact, this was the standard mode of building AI programs in the 1950s and 1960s (*6*), because the main emphasis at that time was on constructing basic systems that were capable of addressing only a few problems. In other words, AI researchers could be their own programmers and experts for the game-playing, puzzle-solving, and mathematical programs being studied at that time.

However, as expert systems began to address more and more practical problems and their associated complexities, a quite natural step was to include a knowledge engineer, who was well-versed in the discipline of programming and the concepts of expert system architecture.

Knowledge Engineering

Applying knowledge without understanding can diminish the potential success of any endeavor. Understanding (i.e., experience, skill, wisdom), as used in this context, can therefore produce increased computational speed, reduced errors, and increased adaptability (*26*). The concept of knowledge engineering is therefore aimed at building skilled expert systems by first extracting the expert's knowledge and then organizing it in an efficient manner (*27*).

Knowledge engineering, as is typical with other engineering fields, consists of a combination of theory and practice (*27*). The field of knowledge engineering is characterized by three main points: most knowledge based systems exhibit properties common to other intelligent problem-solving systems, whether human or artificial; the design and organization of a given system must be governed by the type and complexity of the problem it will address, as well as the available heuristic knowledge; the knowledge engineer must be able to convert the capacity for intelligent action, inherent in knowledge itself, into appropriate forms that can be used by the system.

There are three basic types of data the knowledge engineer can use to improve the efficiency or effectiveness of an expert system (*27*). These are facts, beliefs, and heuristics, which express rules of good judgment in situations where valid algorithms do not exist. Thus expert systems, like human experts, are distinguished by the quality and quantity of the knowledge they possess.

Knowledge as the Measure of Performance

The discussion of knowledge naturally leads to the need for a formal definition of the term. Knowledge can be defined as a combination of the symbolic descriptions that characterize the definitional and empirical relationships in a given area, and the procedures for manipulating those descriptions (*1*). Expertise can be thought of as detailed knowledge about a specific domain, that is, an understanding of the domain problems and the necessary skills to solve those problems (*1*). This knowledge typically consists of two types: public and private. Public knowledge has generally been quantified to the extent that it may be considered static and is found in textbooks and journals. Human experts, on the other hand, usually possess private knowledge that has not found its way into the published literature. This private knowledge is made up of rules of thumb that have become known as heuristics (*28*).

An important reason for focusing on knowledge is the recognition of it as a scarce resource (*1*). Traditionally, the process of transforming a human trainee into an expert is a multi-year process, involving extensive education and experience. The benefits of extracting knowledge from a human expert and encoding it in a computer program include a significant reduction in the amount of time required to develop expertise in a given domain, which leads directly to substantial cost savings.

Evaluation

Evaluation involves three major activities: validation, verification, and certification. Consistency, completeness, and correctness are the major goals of evaluation. To date, mostly qualitative evaluation methods have been developed. As knowledge and experience in this area are gained, new quantitative methods are being developed.

Issues on KBES evaluation are not standardized, due to the difficulties that are inherent to these systems, and to the huge variety of programming and working environments. However, advances in structuring evaluation procedures have reached acceptable levels (*29*). These methods narrow the variability in evaluation to acceptable levels, at least in procedures to select attributes and techniques of evaluation.

EXPERT SYSTEMS DEVELOPMENT TOOLS

Expert systems development tools range from interpreters of relatively simple languages to very elaborate tools (*24*). A development tool that helps the user select appropriate knowledge representation and develop it for a specific problem should have the following attributes:

- Use clear knowledge representation schemes. This is very important in the beginning stages of the development.
- Provide aids for poorly structured or poorly understood problems.
- Compare alternative representations of a given set of knowledge.
- Help the user select a representation scheme. This could be done with examples or by using a tutorial.
- Provide translation to a standard knowledge representation for portability.
- Provide a variety of interlinked representations that use ordinary knowledge forms as well as more sophisticated artificial intelligence forms.

The development tool should allow for different levels of expertise of the developers using it. The simpler tools are shells into which knowledge is inserted in a structured fashion. However, for the more sophisticated developer, there should be a wider choice of strategies and representations (*30*).

In the knowledge base creation there may be an editor or word processor, either menu driven or line entry. Compilation could be by increments or by batch. A help facility should be available. Tools for debugging would be an editor, a syntactic checker, a tracing facility, and a library facility. To provide the most useful user-interface, there could be screen formatting, graphics utilities, or animation creation. The developer could control the inference engine for searches, setting priorities, or accessing the knowledge base (*30*).

The end-user interface must be user-friendly, and the development tool must allow for this type of facility. Responses to the user may be by menu or line commands. There may be multiple or uncertain responses from the user. Responding to "how," "why," and "what if" queries will increase the user's understanding, as will providing graphics when appropriate.

Interfaces may be available to other hardware and software. Input and output may be from or to other computers, including LISP machines, standard microcomputers, workstations, and mainframes (*31*).

Just as it is important to construct an expert system with the correct structure, it is equally important to implement it into an acceptable computer language. Several languages have been designed that are especially useful for AI applications.

Recently, the creation of software development tools that can be used on personal computers have caused an increase in the use of expert systems in many industries and offices (*32*). Today, these microcomputer-based expert systems are at the stage of early development and marketing.

General purpose tools for expert system development have been created for use on microcomputers, minicomputers, and mainframes. As computer memory size increases, so does the capability of having expert systems that increase in complexity and the number of rules that can be handled for each expert system application.

Because some relatively capable expert systems tools are already available on microcomputers, it is possible to experiment with KBES ideas and potentialities without making a sizable investment in resources.

KBES development tools vary from those tailored to people with little programming experience to packages designed for the more sophisticated user with some programming experience. The latter are tools for those users who wish to have maximum flexibility in constructing knowledge bases and the accompanying user interface. Very large expert systems, with many rules and large data bases, may not be appropriate for development on standard microcomputers. The size of the expert system may exceed the memory capacity of the microcomputer, or the computing time required to find solutions may be unreasonable. Beyond this limitation, microcomputer implementations of KBES development tools are no less capable of producing useful expert systems than their mini- and mainframe counterparts (*32*).

Below is an annotated review of some of the more common development tools that are or have been used in transportation applications, as shown by questionnaire results. Appendix C contains a more comprehensive listing of available tools.

ART—The Automated Reasoning Tool from Inference Corporation is written in LISP and runs on LISP machines. It provides a technique called viewpoints that allows hypothetical reasoning, a way to structure the database by defining the contexts in which facts and rules apply. ART supports rules, frames, and procedure-oriented representations and provides a graphical monitor and standard debugging facilities. Suitable applications include planning and scheduling, simulation, configuration generation, and design.

KEE—The Knowledge Engineering Environment from IntelliCorp was originally developed for genetic engineering. KEE is implemented in LISP and is available on LISP machines. Its basic representation paradigm is frames and slots. Using the object-oriented programming technique, slots can contain instructions, rules, procedures, or pointers to another frame to indicate an inheritance relationship. It supports multiple knowledge bases and forward and backward chaining. Its support environment includes graphics-oriented debugging and an explanation facility to indicate inference chains. Suitable applications are diagnosis, monitoring, real-time process control, planning, design, and simulation.

M.1.—A product of Teknowledge, Inc., M.1. is implemented in both PROLOG and C and is available for the microcomputer. The knowledge representation is rule-based, and it supports backward chaining. Syntax is English language with an interactive debugging tool for tracing, facilities for explaining the reasoning, and a user query when there is missing information. M.1 is most suitable for classification.

Personal Consultant—Personal Consultant is a product of Texas Instruments, implemented in IQLISP, which is a dialect of LISP. Personal Consultant uses both frames and rules representation and supports forward and backward chaining. It is available for use on microcomputers and the TI LISP machine. The user interface is window-oriented and has an explanation facility.

EXSYS—A product of EXSYS, Inc., this software is implemented in C language for microcomputers and uses rules of the IF-THEN-ELSE type. EXSYS is a backward-chaining system that includes a runtime module and a report generator. EXSYS can interface to other products to provide frames and a blackboard facility. External program calls can be made for data acquisition and program execution.

LEVEL5 (formerly Insight2+)—LEVEL5 is a product of Level Five Research. It is implemented in Pascal and is available for microcomputers and VAX machines. The system is rule-based and can support both backward and forward chaining. Facts are represented as objects with valued attributes. It can access external programs and databases. The user interface is menu driven and user-friendly.

OPS5—A development language originated at Carnegie Mellon University, OPS5 is available from Digital Equipment Corporation (DEC) for use on VAX equipment, including the VAX Workstation. It is the basis for R1/XCON, an expert system for configuring VAX systems. OPS5 is a forward-chaining, production-rule tool. The support environment includes editing and debugging facilities. It is implemented in BLISS, MACLISP, and FRANZ LISP.

CLIPS—Developed by NASA at the Johnson Space Center in Houston, Texas, CLIPS is a C Language Integrated Production System and provides high portability, low cost, and easy integration with external systems. It is a rule-based system which supports forward chaining and can be used on a wide variety of computers. It also provides tools for debugging.

NEXPERT OBJECT—Neuron Data developed this hybrid rule- and object-based tool. Its structured knowledge is represented as objects, and its judgemental knowledge as rules. Objects can be created either by using the object editor or dynamically from rule conditions. NEXPERT includes a user-friendly graphics interface, working with MS/WINDOWS. Its primary platform is an IBM-compatible microcomputer, although it is adaptable to other platforms as well.

SUMMARY

The development tool can be selected to fit an individual knowledge representation need. There are inductive tools available that build a system from general statements of knowledge to arrive at conclusions, and deductive tools, which take an explicit set of rules and goals to interact with the user in determining the facts required to satisfy the goals. Most of the available development tools are structured to execute on standard microcomputers, but some of the more elaborate ones require specialized symbolic computing hardware and minicomputers (*33*).

Regardless of which hardware and/or software medium the developer chooses, a system that is truly a KBES must have certain characteristics. These include an inference engine, a separate knowledge base, a system work area (context), and an effective user interface. The user interface is especially important, since the operational KBES will likely be used extensively by people who are not necessarily highly computer literate.

CHAPTER SIX

CONCLUSIONS AND RECOMMENDATIONS

CONCLUSIONS REGARDING PRESENT KBES ACTIVITY

- Expert systems have been successfully implemented in several state highway agencies in the areas of pavement management and analysis, noise barrier design, hazardous materials, and bridge analysis.
- Several impediments prevent widespread implementation throughout the state highway agencies. These include lack of fundamental understanding by agency management concerning the potential for artificial intelligence and KBES applicability in the transportation workplace; lack of a proven track record of that potential being realized; and shortage of staff currently trained in KBES programming and knowledge.
- There has been significant interest across the country in potential KBES activity. Many state highway agencies are currently undertaking projects that may well lead to implemented, useful expert systems programs.
- A number of tools are available to shorten the "learning curve" in the agencies. Chief among these are the shell and programming environments that can make the programming phase of KBES development relatively simple.
- The level of effort required to produce a useful KBES for a typical state highway agency application is likely to be on the order of one to two person-years.
- Development and completion of the knowledge base is the most critical phase of the effort. It is vital that the knowledge base be comprehensive, useful and relevant.

RECOMMENDATIONS REGARDING FUTURE KBES ACTIVITY

- Microcomputer-based platforms should be used. The typical transportation KBES application will fit comfortably on such a machine. IBM-compatible microcomputers at the 386 level are normally powerful enough for a transportation application.

A major exception to this recommendation is the agency that has a strong and effective commitment to centralized, mainframe computing. Such an agency is in a position to add KBES to its available tools, and therefore use those AI software packages developed primarily for mini- and mainframe computing.

- Maximum use should be made of commercially available shells. This will facilitate programming, and allow relatively more focus on issues such as knowledge acquisition, verification, and validation.
- No KBES should become a "black box." Regardless of the platform or programming environment, engineers and other technically competent staff should maintain understanding and control over the results and recommendations coming from an expert system.
- Avoid "re-inventing the wheel." State transportation agencies with an interest in particular application areas should interact with agencies and universities who have undertaken KBES projects in those areas. They should also seek guidance from Federal Highway Administration engineers who are involved in KBES and advanced technologies, and coordinate with TRB Committee A5008 on Expert Systems.
- Seriously investigate opportunities that may be suitable for KBES. This is especially important for state transportation agencies not yet active in KBES, which is clearly an emerging technology that will provide major dividends for those willing to make a commitment. Knowledge based expert systems represent the most promising avenue on the software side of advanced technology applications to transportation.

REFERENCES

1. Hayes-Roth, F., D.A. Waterman, and D.B. Lenat, *Building Expert Systems*, Addison-Wesley Publishing Co., Reading, MA (1983).
2. Waterman, D. A., *A Guide to Expert Systems*, Addison-Wesley Publishing Company, Reading, MA (1986).
3. Feigenbaum, E.A., "Knowledge Engineering for the 1980s," Computer Science Department, Stanford University, Stanford, CA (1982).
4. Harris, R. A., L. F. Cohn, and W. Bowlby, "Designing Noise Barriers Using the Expert System CHINA," *Journal of Transportation Engineering*, American Society of Civil Engineers, New York, NY, Vol. 113, No. 2 (March 1987) pp. 127–138.
5. Ritchie, S. G., "Expert Systems in Pavement Management," *Transportation Research*, Vol. 21A, No. 2, Pergamon Press, Oxford, U. K. (1987).
6. Buchanan, B.G., and E.H. Shortliffe, *Rule-Based Expert Systems*, Addison-Wesley Publishing Co., Reading, MA (1984).
7. Duda, R.O., J. Gaschinig, and P.E. Hart, "Model Design in the Prospector System for Mineral Exploration," *Expert Systems in the Micro Electronic Age*, University of Edinburgh, Scotland (1979).
8. Davis, R. et al., "The Dipmeter Advisor: Interpretation of Geologic Signals," *Proceedings of the Seventh IJCAI* (1981).
9. McDermott, J., "R1: A Rule-Based Configurer of Computer Systems," Technical Report CMU-CS-80-119, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA (1980).
10. Buchanan, B.G., "Expert Systems: Working Systems and the Research Literature," *Expert Systems*, Vol. 3, No. 1 (January 1986).
11. Forsyth, R., *Expert Systems*, Chapman & Hall, London, England (1984).
12. Gevarter, W.B., "An Overview of Expert Systems," NBSIR 82-2505, National Bureau of Standards, Washington, D.C. (May 1982).
13. Kostem, C. N., and M. L. Maher, *Expert Systems in Civil Engineering*, American Society of Civil Engineers, New York, New York (1986).
14. Sriram, B.G., and M.D. Rychener, "Expert Systems for Engineering Applications," *IEEE Software*, Vol. 3, No. 2 (March 1986) pp. 3–6.
15. Sinha, K. C., ed., "Special Secion: Use of Expert Systems in Transportation Engineering," *Journal of Transportation Engineering*, Vol. 113, No. 2, (March 1987) American Society of Civil Engineers, New York, NY, pp. 107–210.
16. Palmer, R. N., "Special Issue: Expert Systems in Civil Engineering," *Journal of Computing in Civil Engineering*, Vo. 1, No. 4 (October 1987) American Society of Civil Engineers, New York, NY, pp. 223–320.
17. Maher, M. L., *Expert Systems for Civil Engineers: Technology and Application*, American Society of Civil Engineers, New York, NY (1987) 148 pp.
18. Jamsa, H., *Proceedings, Organization for Economic Cooperation and Development Workshop on Knowledge Based Expert Systems in Transportation*, Vol 1 and 2, VTT Symposia 116 and 117, Espoo, Finland (June 1990).
19. Wentworth, J. A., *Developing Knowledge Based Expert Systems*, Technology Share Report FHWA-TS-88-022, Office of Technology Applications, U. S. Department of Transportation, Federal Highway Administration, Washington, D. C. (1988).
20. Michie, Jarvis D., *NCHRP Report 230: Recommended Procedures for the Safety Performance Evaluation of Highway Appurtenances,* Transportation Research Board, National Research Council, Washington, D.C.(1981).
21. Ritchie, S.G., L.F. Cohn, and R.A. Harris, *Transportation Research Record 1187: Development of Expert Systems Technology in the California Department of Transportation,* Transportation Research Board, National Research Council, Washington, D.C. (1989).
22. Mettrey, W., "An Assessment of Tools for Building Large Knowledge Based Systems," *AI Magazine* (Winter 1987).
23. Harmon, P., *Expert Systems: Tools & Applications,* Wiley & Sons, Inc., New York, NY (1988) p. 71.
24. Dym, C., and R.E. Levitt, *Knowledge Based Systems in Engineering,* McGraw-Hill, Inc., New York, NY (1991).
25. Harmon, P., and D. King, *Expert Systems: Artificial Intelligence in Business*, John Wiley & Sons, Inc., New York, NY (1985).
26. Lenat, D.B., F. Hayes-Roth, and P. Klahr, "Cognitive Economy," Report HPP-79-15, Heuristic Programming Project, Computer Science Department, Stanford University, Stanford, CA (1979).
27. Hayes-Roth, F., "The Knowledge Based Expert System: A Tutorial," *IEEE Spectrum* (September, 1984) pp. 11–28.
28. Lenat, D.B., "The Nature of Heuristics," *Artificial Intelligence*, Vol. 19 (1982) pp. 189–249.
29. Hite, L. A., and D.P. Miller, "Designing a Testing Strategy for Expert Systems," Virginia Polytechnic Institute and State University, Blacksburg, VA (1988).
30. Gevarter, W.B., "The Nature and Evaluation of Commercial Expert System Building Tools," *Computer* (May 1987) pp. 26–28.
31. Gevarter, W.B., "Expert Systems: Limited But Powerful," *IEEE Spectrum* (August 1983) pp. 39–45.
32. Shafer, D.G., "Microcomputer-based Expert Systems: Where We Are, Where We Are Headed," *Expert Systems*, Vol. 4, (October 1985) pp. 188–195.
33. Citrenbaum, R., "Selecting a Shell," *AI Expert* (September 1987) pp. 3–34.

APPENDIX A
GLOSSARY OF AI AND KBES TERMS

Agenda. An ordered list of actions. Some knowledge systems store and reason about possible actions; for example, whether to pursue a particular line of reasoning. HEARSAY uses agenda-based control.

Algorithm. A systematic procedure that, if followed, guarantees a correct outcome. In developing a conventional program, the programmer must specify the algorithm that the program will follow.

Artificial Intelligence. "A subfield of computer science concerned with the concepts and methods of symbolic inference by a computer and the symbolic representation of the knowledge to be used in making inferences. A field aimed at pursuing the possibility that a computer can be made to behave in ways that humans recognize as intelligent behavior in each other." (*1*)

Attribute. A property of an object. For example, net worth is an attribute of a loan applicant. Attributes are associated with values in specific cases; thus, A. Smith's net worth is $34,000.

Backtracking. The process of backing up through a sequence of inferences, usually in preparation for trying a different path. Planning problems typically require backtracking strategies that allow a system to try one plan after another as unacceptable outcomes are identified.

Backward Chaining (back-chaining). One of several control strategies that regulate the order in which inferences are drawn. In a rule-based system, backward chaining is initiated by a goal rule. The system attempts to determine if the goal rule is correct. It backs up to the IF clauses of the rule and tries to determine if they are correct. This, in turn, leads the system to consider other rules that would confirm the IF clauses. In this way, the system backs into its rules. Eventually, the back-chaining sequence ends when a question is asked or a previously stored result is found.

Blackboard Architecture (HERESAY Architecture). An expert system design in which several independent knowledge bases each examine a common working memory, called a "blackboard." An agenda-based control system continually examines all of the possible pending actions and chooses the one to try next.

Breadth-first Search. In a hierarchy of rules or objects, breadth-first search refers to a strategy in which all rules or objects on the highest level are examined first, resulting in a search across all branches of the hierarchy tree.

Certainty. The degree of confidence one has in a fact or relationship. As used in AI, this contrasts with probability, which is the likelihood that an event will occur.

Certainty Factor (Confidence Factor). A numerical weight given to a fact or relationship to indicate the confidence one has in the fact or relationship. These numbers behave differently from probability coefficients. In general, methods for manipulating certainty factors are more informal than approaches to combining probabilities. Most rule-based systems use certainty factors rather than probabilities.

Chunk. A collection of facts stored and retrieved as a single unit. The limitations of working memory are usually defined in terms of the number of chunks that can be handled simultaneously.

Common LISP. A dialect of LISP intended to serve as a standard version of LISP that will run on a number of different machines. The first efforts to develop such a dialect have already met with some difficulties. LISP is such an easy language to tailor that people implementing it rarely resist customizing it for the particular computer they are using.

Compiled Knowledge. As a person acquires and organizes knowledge into chunks and networks, the knowledge becomes compiled. Some individuals compile knowledge into more and more abstract and theoretical patterns (deep knowledge). Others compile knowledge as a result of practical experience (surface knowledge). Most people begin by acquiring theoretical knowledge and then, when they finish their schooling, they recompile what they have learned into practical heuristics. Expertise consists of large amounts of compiled knowledge.

Context Tree (Object Tree). In EMYCIN, the context tree forms the backbone of the consultant program. It is a structured arrangement of the objects (contexts) or conceptual entities that constitute the consultation domain. There may be one or more contexts. A static context tree is an arrangement of context types (e.g., a patient for whom cultures have been prepared). A dynamic context tree is an arrangement of context instances (e.g., John Smith with a morning culture and an afternoon culture).

Context-Parameter-Value Triplets (Object-Attribute-Value Triplets). One method of representing factual knowledge; it is the method used in EMYCIN. A context is an actual or conceptual entity in the domain of the consultant (e.g., a patient, an aircraft, or an oil well). Parameters are properties associated with each context (e.g., age and sex of a patient or location and depth of an oil well). Each parameter (or attribute) can take on values: the parameter age could take the value "13 years."

Control (of a knowledge system). The method used by the inference engine to regulate the order in which reasoning occurs. Backward chaining, forward chaining, and blackboard agendas are all examples of control methods.

Deep Knowledge. Knowledge of basic theories, first principles, axioms, and facts about a domain. This contrasts with surface knowledge.

Depth-first Search. In a hierarchy of rules or objects, depth-first search refers to a strategy in which one rule or object on the highest level is examined and then the rules or objects immediately below that one are examined. Proceeding in this manner, the system will search down a single branch of the hierarchy tree until it ends. This contrasts with breadth-first search.

Diagnostic/Prescriptive Consultation Paradigm. Consultation paradigms refer to generic approaches to common types of problems. The diagnostic/prescriptive paradigm is used for problems that require the user to identify symptoms or characteristics of a situation in order to determine which of several alternative solutions may be appropriate. Most expert systems and tools are designed to handle this paradigm.

Domain. A topical area or region of knowledge. Medicine, engineering, and management science are very broad domains. Existing knowledge systems only provide competent advice within very narrowly defined domains.

Dual Semantics. The idea that a computer program can be viewed from either of two equally valid perspectives: procedural semantics (what happens when the program is run) and declarative semantics (what knowledge the program contains).

EMYCIN. The first expert system building tool. EMYCIN was derived from the expert system MYCIN. After the developers of MYCIN completed that system, they decided that they could remove the specific medical knowledge from MYCIN (hence, Essential MYCIN). The resulting shell consisted of a back-chaining inference engine, a consultation driver, and several knowledge acquisition aids. This shell, or tool, could then be combined with another knowledge base to create a new expert system.

Exhaustive Search. A search is exhaustive if every possible path through a decision tree or network is examined. Exhaustive search is costly or impossible for many problems. A knowledge based system will often search exhaustively through its knowledge base.

Experiential Knowledge. Knowledge gained from hands-on experience. This typically consists of specific facts and rules of thumb (surface knowledge). This is in contrast with deep knowledge of formal principles or theories.

Expert System. As originally used, the term referred to a computer system that could perform at, or near, the level of a human expert. Evaluations of MYCIN place its competence at or near that of highly specialized physicians. Configuration systems like XCON (R1) may well exceed human competence. As the term is currently being used, it refers to any computer system that was developed by means of a loose collection of techniques associated with AI research. Thus, any computer system developed by means of an expert system building tool would qualify as an expert system even if the system was so narrowly constrained that it could never be said to rival a human expert. Some practitioners would prefer to reserve "expert system" for systems that truly rival human experts and use "knowledge system" when speaking of small systems developed by means of AI techniques.

Expertise. The skill and knowledge possessed by some humans that result in performance far above the norm. Expertise often consists of massive amounts of information combined with rules of thumb, simplifications, rare facts, and wise procedures in such a way that one can analyze specific types of problems in an efficient manner.

Explanation. Broadly, this refers to information presented to justify a particular course of reasoning or action. In knowledge systems, this typically refers to a number of techniques that help a user understand what a system is doing. Many knowledge systems allow a user to ask "Why," "How," or "Explain." In each case, the system responds by revealing something about its assumptions or its inner reasoning.

Fact. Broadly, a statement whose validity is accepted. In most knowledge systems, a fact consists of an attribute and a specific associated value.

Forward Chaining. One of several control strategies that regulate the order in which inferences are drawn. In a rule-based system, forward chaining begins by asserting all of the rules whose IF clauses are true. It then checks to determine what additional rules might be true, given the facts it has already established. This process is repeated until the program reaches a goal or runs out of new possibilities.

Frame (Object or Unit). A knowledge representation scheme that associates an object with a collection of features (e.g., facts, rules, defaults, and active values). Each feature is stored in a slot. A frame is the set of slots related to a specific object. A frame is similar to a property list, schema, or record, as these terms are used in conventional programming.

Frame Based. Frame based representation provides a method to represent knowledge that is highly structured. Knowledge is usually represented by record-like objects. Frames may be linked together hierarchically for performance and frames may inherit characteristics from other "parent" frames. Procedures can be linked to a frame to be invoked when a given frame is read or changed.

Heuristic. A rule of thumb or other device or simplification that reduces or limits search in large problem spaces. Unlike algorithms, heuristics do not guarantee correct solution.

Heuristic Rules. Rules written to capture the heuristics an expert uses to solve a problem. The expert's original heuristics may not have taken the form of IF-THEN rules, and one of the problems involved in building a knowledge system is converting an expert's heuristic knowledge into rules. The power of a knowledge system reflects the heuristic rules in the knowledge base.

Hierarchy. An ordered network of concepts or objects in which some are subordinate to others. Hierarchies occur in biological taxonomies and corporate organizational charts. Hierarchies ordinarily imply inheritance; thus, objects or concepts higher in the organization "contain" the objects or concepts that were beneath them. "Tangled hierarchies" occur when more than one higher-level entity inherits characteristics from a single lower-level entity.

High-Level Languages. Computer languages lie on a spectrum that ranges from machine instructions through intermediate languages including FORTRAN and COBOL to high-level languages such as Ada and C. High-level languages incorporate more complex constructs than the simpler languages.

Human Information Processing. A perspective on how humans think that is influenced by how computers work. This approach to psychology begins by focusing on the information that a person uses to reach some conclusion and then asks how one could design a computer program that would begin with the same information and reach that same conclusion. Espoused by Herbert Simon and Allan Newell, this perspective currently dominates cognitive psychology and has influenced the design of both computer languages and programs.

IF-THEN Rule. A statement of the relationship among a set of facts. The relationships may be definitional (e.g., IF female and married, THEN wife), or heuristic (e.g., IF cloudy, THEN take umbrella).

Induction System (Example-driven System). A knowledge system that has a knowledge base consisting of examples. An induction algorithm builds a decision tree from the examples, and the system goes on to deliver advice. Induction systems do not facilitate the development of hierarchies of rules.

Inference. The process by which new facts are derived from known facts. A rule combined with a rule of inference and a known fact results in a new fact.

Inference, Data-directed. Inferences that are driven by events rather than goals. See Forward Chaining.

Inference, Goal-directed. Inferences that are driven by goals rather than data. See Backward Chaining.

Inference Engine. That portion of a knowledge system that contains the inference and control strategies. More broadly, the inference engine also includes various knowledge acquisition, explanation, and user interface subsystems. Inference engines are characterized by the inference and control strategies they use.

Inheritance. A process by which characteristics of one object are assumed to be characteristics of another. If we determine that an animal is a bird, for example, then we automatically assume that the animal has all of the characteristics of birds.

Inheritance Hierarchies. When knowledge is represented in a hierarchy, the characteristics of superordinate objects are inherited by subordinate objects. Thus, if we determine that an auto loan is a type of loan, then we know that the credit check procedures that apply to all loans apply to auto loans.

Interface. The link between a computer program and the outside world. A single program may have several interfaces. Knowledge systems typically have interfaces for development (the knowledge acquisition interface) and for users (the user interface). In addition, some systems have interfaces that pass information to and from other programs, data bases, display devices, or sensors.

INTERLISP. A dialect of LISt Processing (LISP). A programming environment that provides a programmer with many aids to facilitate the development and maintenance of large LISP programs.

Knowledge. An integrated collection of facts and relationships which, when exercised, produces competent performance. The quantity and quality of knowledge possessed by a person or a computer can be judged by the variety of situations in which the person or program can obtain successful results.

Knowledge Acquisition. The process of locating, collecting, and refining knowledge. This may require interviews with experts, research in a library, or introspection. The person undertaking the knowledge acquisition must convert the acquired knowledge into a form that can be used by a computer program.

Knowledge Base. The portion of a knowledge system that consists of the facts and heuristics about a domain.

Knowledge Engineer. An individual whose specialty is assessing problems, acquiring knowledge, and building knowledge systems. Ordinarily, this implies training in cognitive science, computer science, and artificial intelligence. It also suggests experience in the actual development of one or more expert systems.

Knowledge Representation. The method used to encode and store facts and relationships in a knowledge base. Semantic networks, object-attribute-value triplets, production rules, frames, and logical expressions are all ways to represent knowledge.

Knowledge System. A computer program that uses knowledge and inference procedures to solve difficult problems. The knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model of the expertise of skilled practitioners. In contrast to expert systems, knowledge systems are often designed to solve small, difficult problems rather than large problems requiring true human expertise. In many cases, small knowledge systems derive their utility from their user-friendly nature rather than from their ability to capture knowledge that would be difficult to represent in a conventional program.

Language-Tool Spectrum. A continuum along which various software products can be placed. At one extreme are narrowly defined tools that are optimized to perform specific tasks. At the other extreme are general purpose languages that can be used for many different applications.

Large, Hybrid System Building Tools. A class of knowledge engineering tools that emphasizes flexibility. The systems are designed for building large knowledge bases. They usually include a hybrid collection of different inference and control strategies. Most commercial hybrid tools incorporate frames and facilitate object-oriented programming.

Large, Narrow System Building Tools. A class of knowledge engineering tools that sacrifices flexibility to facilitate the efficient development of more narrowly defined expert systems. At the moment, most large, narrow tools emphasize production rules.

LISP. A programming language based on List Processing. LISP is the language of choice for American AI researchers.

Logic. A system that prescribes rules for manipulating symbols. Common systems of logic powerful enough to deal with knowledge structures include propositional calculus and predicate calculus.

Long-term Memory. A portion of human memory that is exceedingly large and contains all of the information that is not currently being processed.

MacLISP. A dialect of LISP that is tuned for efficiency, but is less friendly as a developmental environment.

Meta-. A prefix indicating that a term is being used to refer to itself. Thus, a meta-rule is a rule about other rules.

Monotonic Reasoning. A reasoning system based on the assumption that once a fact is determined it cannot be altered during the course of the reasoning process. MYCIN is a monotonic system; and, thus, once the user has answered a question, the system assumes that the answer will remain the same throughout the session. Given the brief duration of most MYCIN sessions, this is a reasonable assumption.

Multivalued Attribute. An attribute that can have more than one value. If, for example, a system seeks values for the attribute restaurant, and if restaurant is multi-valued, then two or more restaurants may be identified.

MYCIN. An expert system developed at Stanford University in the mid 1970s, this research system is designed to aid physicians in the diagnosis and treatment of meningitis and bacteremia infections. MYCIN is often spoken of as the first expert system. There were other systems that used many of the AI techniques associated with expert systems, but MYCIN was the first to combine all of the major features with the clear separation of the knowledge base and the inference engine. This separation, in turn, led to the subsequent development of the first expert system building tool, EMYCIN.

Natural Language Programming. The branch of AI research that studies techniques that allow computer systems to accept inputs and produce outputs in a conventional language like English. At the moment, systems can be built that will accept typed input in narrowly constrained domains (e.g., data base inquiries). Several expert systems incorporate some primitive form of natural language in their user interface to facilitate rapid development of new knowledge bases.

Neural Networks. Neural networks and expert systems use significantly different approaches to problems. Neural networks' strength is not in capturing how experts resolve problems but in finding patterns in data that are important. Neural networks have been successfully employed in luggage scanning systems to differentiate small discrepancies between packaged food, clothing, and explosives. Neural networks have been developed to solve problems in signal processing, noise filtering, process control, forecasting, and many other applications.

Expert systems capture data on how experts approach problems, whereas neural networks capture data that are important to the experts. Expert systems and neural networks can be very powerful when used together. The luggage-scanning system uses its neural network to discriminate between similar patterns of important items of data, whereas the expert system identifies luggage that is empty or contains unimportant or few items.

Nonmonotonic Reasoning. Reasoning that can be revised if some value changes during a session. In other words, nonmonotonic reasoning can deal with problems that involve rapid changes in values in short periods of time. If one were developing an online expert system that monitored the stock market and recommended stocks to purchase, one would want a system that used nonmonotonic reasoning and was thus able to revise its recommendations continually as the prices and volumes of stock changed.

Object (Context, Frame). Broadly, this refers to physical or conceptual entities that have many attributes. When a collection of attributes or rules are divided into groups, each of the groups is organized around an object. In MYCIN, following medical practice, the basic groups of attributes (parameters) were clustered into contexts, but most recent systems have preferred the term "object." When a knowledge base is divided into objects, it is often represented by an object tree that shows how the different objects relate to each other. When one uses object-oriented programming, each object is called a frame or unit and the attributes and values associated with it are stored in slots. An object is said to be "static" if it simply describes the generic relationship of a collection of attributes and possible values. It is said to be "dynamic" when an expert system consultation is being run and particular values have been associated with a specific example of the object.

Object-Attribute-Value Triplets (O-A-V Triplets). One method of representing factual knowledge. This is the more general and common set of terms used to describe the relationships referred to as Context-Parameter-Value Triplets in EMYCIN. An object is an actual or conceptual entity in the domain of the consultant (e.g., an oil well). Attributes are properties associated with objects (e.g., location, depth, productivity). Each attribute can take different values (e.g., the attribute depth could take on any numerical value from 0 to 60,000 feet).

Parallel Processing. An architecture for computer machinery that allows a computer to run several programs at once, using several central processors to simultaneously process information.

Production (Production Rule). The term used by cognitive psychologists to describe an IF-THEN rule.

Production System. A production system is a human or computer system that has a database of production rules and some control mechanism that selects applicable production rules in an effort to reach some goal state. OPS5 is an expert system building tool that is normally referred to as a production system; it was initially developed in an effort to model supposed human mental operations.

Programming Environment (Environment). A programming environment is about halfway between a language and a tool. A language allows the user complete flexibility. A tool constrains the user in many ways. A programming environment, such as INTERLISP, provides a number of established routines that can facilitate the quick development of certain types of programs.

PROLOG. A symbolic or AI programming language based on predicate calculus.

Prototype. In expert systems development, a prototype is an initial version of an expert system, usually a system with from 25 to 200 rules, that is developed to test effectiveness of the overall knowledge representation and inference strategies being employed to solve a particular problem.

Reasoning. The process of drawing inferences or conclusions.

Representation. The way in which a system stores knowledge about a domain. Knowledge consists of facts and the relationships between facts.

Resolution (Resolution Theorem Proving). The inference strategy used in logical systems to determine the truth of an assertion. This complex, but highly effective, method establishes the truth of an assertion by determining that a contradiction is encountered when one attempts to resolve clauses, one of which is a negation of the thesis one seeks to assert.

Robotics. The branch of AI research that is concerned with enabling computers to "see" and "manipulate" objects in their surrounding environment. AI is not concerned with robotics, as such, but it is concerned with developing the techniques necessary to develop robots that can use heuristics to function in a highly flexible manner while interacting with a constantly changing environment.

Rule (IF-THEN Rule, Production). A conditional statement of two parts. The first part, composed of one or more IF clauses, establishes conditions that must apply if a second part, composed of one or more THEN clauses, is to be acted upon. The clauses of rules are usually A-V pairs or O-A-V triplets.

Rule-based Program (Production System). A computer program that represents knowledge by means of rules.

Semantic. Refers to the meaning of an expression. It is often contrasted with syntactic, which refers to the formal pattern of the expression. Computers are good at establishing that the correct syntax is being used; they have a great deal of trouble establishing the semantic content of an expression. For example, consider the sentence, "Mary had a little lamb." It is a grammatically correct sentence; its syntax is in order. But its semantic content (its meaning) is very ambiguous. As we alter the context in which the sentence occurs, the meaning will change.

Semantic Networks. A type of knowledge representation that formalizes objects, values, and nodes and connects the nodes with arcs or links that indicate the relationships between the various modes.

Slot. A component of an object in a frame system. Slots can contain intrinsic features such as the object's name, attributes and values, attributes with default values, rules to determine values, pointers to related frames, and information about the frame's creator, etc.

Surface Knowledge (Experiential or Heuristic Knowledge). Knowledge that is acquired from experience and is used to solve practical problems. Surface knowledge usually involves specific facts and theories about a particular domain or task and a large number of rules of thumb.

Symbolic versus Numeric Programming. A contrast between the two primary uses of computers. Data reduction, database management, and word processing are examples of conventional or numerical programming. Knowledge systems depend on symbolic programming to manipulate strings of symbols with logical rather than numerical operators.

Uncertainty. In the context of expert systems, uncertainty refers to a value that cannot be determined during a consultation. Most expert systems can accommodate uncertainty. That is, they allow the user to indicate if he or she does not know the answer. In this case, the system either uses its other rules to try to establish the value by other means or relies on default values.

(Assistance from the Illinois Department of Transportation in preparing this glossary is appreciated by the authors.)

REFERENCE

1. E.A. Feigenbaum and P. McCorduck, *The Fifth Generation*, Addison-Wesley, Reading, MA (1983).

APPENDIX B
Detailed Discussion of Two KBES Applications
PARADIGM—The Pavement Rehabilitation Analyst
CHINA—The Highway Noise Barrier Designer

PARADIGM

The Pavement Rehabilitation Analysis and Design Mentor

Stephen G. Ritchie, Manuan Kim, and Neil A. Prosser, University of California-Irvine

(Much of this section is extracted from the *Proceedings* of the OECD Workshop on Expert Systems in Transportation, Espoo, Finland, June 26-28, 1990.)

INTRODUCTION

The Pavement Rehabilitation Analysis and Design Mentor (PARADIGM) is a microcomputer-based, integrated set of interacting expert systems and algorithmic models. It consists of three major components: the Surface Condition Expert for Pavement Rehabilitation (SCEPTRE), the Overlay Design Heuristic Adviser (OVERDRIVE), and network optimization. SCEPTRE assists highway engineers in identifying cost-effective flexible pavement rehabilitation strategies at the project level. OVERDRIVE assists in detailed design of asphalt concrete overlays on existing flexible pavement.

The objective of the research on which this KBES is based was to develop the first operational and integrated version of PARADIGM for flexible pavements, emphasizing asphalt concrete surfaces and overlays. Separate prototype versions of SCEPTRE (*1,2*) and OVERDRIVE (*3*) were integrated, tested, evaluated, and revised to produce PARADIGM, version 1.0. In addition, several conventional, algorithmic programs were developed or modified and incorporated within PARADIGM.

A significant feature of PARADIGM is the interfacing of symbolic reasoning and algorithmic models. An external, conventional program for life-cycle costing of overlay designs in OVERDRIVE was adapted for SCEPTRE so that it could also estimate the net present value of feasible rehabilitation and maintenance (RAM) strategies by comparison with the do-nothing alternative. The program also enables collection and modification of the data that are to be used in network-level decision making. A highway network investment decision model was developed that uses project-level information generated by SCEPTRE to derive an optimized network-level rehabilitation plan, subject to user-specified budget constraints. The problem is solved using an integer programming approach and interfaces to a project-level database. Two algorithms may be implemented; one very quickly produces an approximate solution, the other takes longer to provide an exact solution.

OVERVIEW

PARADIGM version 1.0 is a forward chaining knowledge based expert system employing natural English language IF-THEN-ELSE production rules in its knowledge base with links to conventionally coded external programs. The system was developed using the EXSYS KBES development environment, and C, Pascal, and FORTRAN programming tools. The PARADIGM knowledge base contains more than 470 rules, and the amount of conventional code is quite extensive.

PARADIGM consists of three main component systems: SCEPTRE, OVERDRIVE, and Network Optimization. These three main systems are represented and controlled through the production rules in the knowledge base of PARADIGM. The broad structure of PARADIGM is presented in Figure B-1. The three systems are overviewed in the following subsections.

SCEPTRE

SCEPTRE evaluates project-level pavement surface distress and other user inputs and recommends feasible rehabilitation strategies for subsequent detailed analysis, design, and network optimization.

Surface condition evaluation is typically based on interpretation of field measurements relating to three performance indicators (*2*): ride quality, safety, and surface distress. This evaluation does not usually provide sufficient information for design purposes, which is ordinarily obtained from additional assessment of structural adequacy. However, evaluation of a pavement's surface condition enables a judgment to be made regarding the pavement's adequacy for current service and probable causes of surface distress, as well as the need for structural evaluation. It is also used to determine the need and priority for various maintenance and rehabilitation strategies, based upon expert judgment.

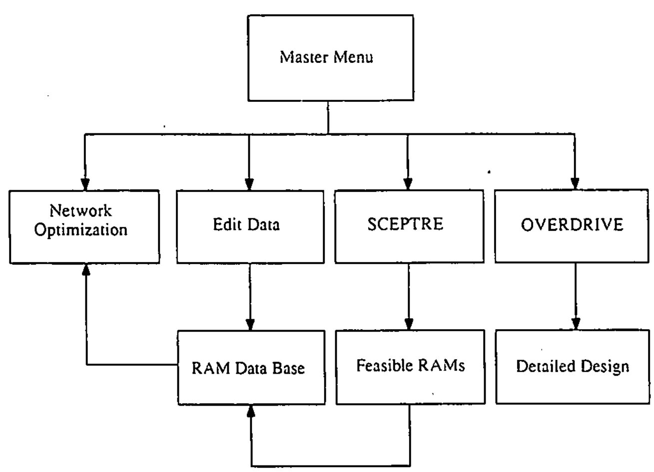


FIGURE B-1 Overview of paradigm structure.

Of the three performance indicators used in the pavement surface condition evaluation, the most important is surface distress. This is usually obtained by a manual inspection and visual analysis of the pavement surface. For flexible pavements with bituminous surfaces, it includes the severity and extent of distresses such as alligator cracking and transverse cracking. This information is vital to the assessment of current and future pavement condition and rehabilitation needs.

SCEPTRE queries the user for inputs that are used by the system to make inferences and reach conclusions, based on a collection of facts and heuristics that have been incorporated into the system's knowledge base. Seven basic factors to be obtained from the user are as follows:

- Type of surface course,
- Type of surface distress,
- Amount of surface distress,
- Severity of surface distress,
- Existing pavement performance (rate of deterioration),
- Traffic levels, and
- Climate.

The knowledge base has been constructed using the combined expertise of two pavement specialists with extensive experience in pavement rehabilitation in Washington state and Texas.

In identifying the feasible rehabilitation and maintenance strategies (RAMs), SCEPTRE considers a list of 23 possible options. Based on user inputs, SCEPTRE refines this list to form an appropriate subset. There are 10 basic strategies in SCEPTRE, as listed in Table B-1. Depending on the combined existence and severity of several distress types in the pavement segment, many of the strategies are modified to include a recommendation to first "prelevel" or "prelevel or mill" the pavement as part of the rehabilitation strategy. These variations on the 10 basic RAMs result in 23 total RAMs. The specific types of surface distress are compatible with those used in the Washington State Department of Transportation (WSDOT) pavement management system (PMS).

An example rule in SCEPTRE is:

RULE NUMBER: 140

IF

(1) Need to run SCEPTRE
and (2) An assessment of alligator cracking is needed
and (3) The regional climate has marine dominance
and (4) The percent length of both wheel paths alligator cracked is 25 percent or more.
and (5) The severity of alligator cracking involves spalling, or spalling and pumping.
and (6) The observed or predicted service life of the existing pavement is 10 years or more.
and (7) The AADT (Average Annual Daily Traffic) on this segment is more than 4,000 vehicles/lane

THEN

(1) Do Nothing
and (2) Thin asphalt concrete overlay
and (3) Medium asphalt concrete overlay
and (4) Thick asphalt concrete overlay
and (5) [RAM1] IS GIVEN THE VALUE 010101
and (6) [RAM11] IS GIVEN THE VALUE 110603
and (7) [RAM12] IS GIVEN THE VALUE 120903
and (8) [RAM13] IS GIVEN THE VALUE 131103
and (9) [WASH] IS GIVEN THE VALUE "WEST"

When all IF portion premises are true, the feasible strategies, 1, 2, 3, and 4 in the THEN portion are recommended by SCEPTRE as a list of reasonable solution options to be considered later in the process. The values in the rest of the THEN portion are stored in a file named "PASS.DAT." This file is an input file for an external program which will be executed later in the system. The six-digit numbers in the THEN portion represent RAM identification, service life, and the standard deviation of the service life.

SCEPTRE also performs cost-effectiveness analysis, based on life-cycle costs and pavement performance, for each feasible strategy. The cost-effectiveness analysis is described in a later subsection.

TABLE B-1
BASIC REHABILITATION AND MAINTENANCE STRATEGIES (RAMs) IN SCEPTRE

Basic Strategies
1. Do nothing
2. Fill cracks
3. Fog seal
4. Friction course
5. Chip seal
6. Double chip seal
7. Thin asphalt concrete overlay (0.10 ft or less)
8. Medium asphalt concrete overlay (0.10 - 0.25 ft)
9. Thick asphalt concrete overlay (0.25 ft or more)
10. Mill and replace

OVERDRIVE

OVERDRIVE is an expert system for assessment of existing structural adequacy and the design of flexible asphalt concrete overlays on existing flexible pavement.

OVERDRIVE is based on the component analysis overlay design method. This is a traditional design method that involves a comparison between the existing pavement structure in terms of its component layers, and a new full-depth design that takes into account site-specific conditions.

Evaluation of the existing structure focuses on determining the effective thickness of each layer of the pavement structure. OVERDRIVE is applicable to existing pavement structures containing up to three layers, such as a surface course, base course, and sub-base. The effective thickness for the structure is given by the sum of the effective thicknesses of each layer. Each effective thickness represents an equivalent depth of new asphalt concrete, and is given by the product of the actual layer thickness and an appropriate conversion factor. A major part of OVERDRIVE is devoted to selecting these conversion factors for an existing structure.

For surface courses and some types of base layers, the conversion factors are derived from assessment of surface distresses such as the severity of rutting and the severity and extent of alligator cracking. Performing an effective thickness analysis requires knowledge of the following items for each individual structure:

- Number of existing layers
- Thickness of each layer
- Layer material type
- Layer condition

This information can be obtained from past records and a field survey, or by limited sampling and testing of in-place materials. OVERDRIVE incorporates 17 combinations of layer and material type.

An example of a rule in OVERDRIVE is:

RULE NUMBER: 310

IF

(1) Need to run OVERDRIVE
and (2) The surface course type is asphalt concrete
and (3) The percent length of both wheel paths alligator cracked is 10 percent or more but less than 25 percent
and (4) The severity of alligator cracking involves spalling, or spalling and pumping
and (5) The severity of rutting depth is 0″ or more but less than 0.5″

THEN

(1) [S1] is given the value 0.4
and (2) Surface course conversion factor is determined

In this case, when the left side premises ((1) through (5) above) are true, a conversion factor ([S1]) of 0.4 is assigned. In other words, the original surface course has an equivalent thickness of new asphalt concrete of 40 percent of its actual thickness.

A new design for a full-depth asphalt concrete pavement over the existing subgrade is based directly on the elastic layered theory approach of the Asphalt Institute (*4*). OVERDRIVE allows the use of default parameter values for design purposes when actual values are not available, as may be the case for users in many smaller local agencies. Determining new full-depth design thickness is performed by interfacing the knowledge base to an external Pascal program that contains equations replicating Asphalt Institute design curves. Based on the effective thickness and new full-depth construction thickness, a simple calculation is required to determine the structural design thickness of any necessary asphalt concrete overlay for the given service conditions.

Once all factors are identified and determined, the system presents one of the following three displays:

1. Based on your input data, and my preliminary evaluation, the existing pavement section is not structurally adequate, and an overlay is required to provide greater structural capacity. The thickness of asphalt concrete overlay required is approximately 3 inches.

2. The existing pavement section is structurally adequate, and no overlay is required to provide greater structural capacity.

3. The severity of rutting is such that reconstruction, or milling and overlay, of the existing pavement section is required. A determination of the required overlay thickness in this case is beyond my present capability.

The third conclusion is presented only when the rutting depth is more than 1.5 inches.

The knowledge base of OVERDRIVE is the result of knowledge engineering efforts with a pavement specialist combined with a synthesis of state-of-the-art and other reports, papers, and manuals relating to the Asphalt Institute overlay design method for asphalt concrete overlays on flexible pavement.

OVERDRIVE can also perform life-cycle cost analysis of both the overlay and the do-nothing alternative through an interface to an external program. This life-cycle cost analysis process is described in the next subsection.

A powerful feature that is exploited in OVERDRIVE is the ability of the user, at the end of the design session, to view and change any of the inputs for that session and have OVERDRIVE automatically redesign the structural thickness of an overlay. If, due to the user's changes, OVERDRIVE requires further information, this will be requested from the user. However, it is not necessary to reenter all the inputs.

Whenever the system requests information, it is possible to enter "why," to which the system will respond with a listing of the applicable rule(s) it is attempting to verify, along with any attached notes or references inserted to aid in explaining each rule.

LIFE-CYCLE COST ANALYSIS

Even when the most effective strategy has been chosen, one of several satisfactory design schemes must still be selected. The controlling factor in this decision can be economic considerations. To make a responsible choice, the costs of several strategies should be compared over some finite period of time (the cost-analysis period).

Both SCEPTRE and OVERDRIVE include their own external FORTRAN and Pascal programs, respectively, for life-cycle cost analysis. SCEPTRE performs approximate life-cycle cost analysis of feasible maintenance and rehabilitation strategies, while OVERDRIVE performs detailed life-cycle cost analysis of each overlay design thickness. Each system interfaces with its external program by passing out data the program needs, and letting the external program perform intensive numerical calculations and by customizing the display and printing of results. The following two subsections describe each life-cycle cost analysis in SCEPTRE and OVERDRIVE.

Cost Analysis for Feasible RAMs

For each RAM strategy determined to be feasible by SCEPTRE, a cost-effectiveness analysis based on life-cycle costs and pavement performance can be performed at the user's request.

The basis of the analysis for each RAM is a pavement performance curve, similar to that used in the WSDOT PMS, as shown in Figure B-2. This curve relates the age of the pavement to an overall pavement condition rating, which is a function of both ride quality and the surface condition rating. As a pavement ages, its condition gradually deteriorates, therefore, WSDOT defines two rehabilitation levels, one where the pavement "should" be rehabilitated and one where the pavement "must" be rehabilitated. These levels correspond to pavement condition ratings of 60 and 40 out of 100, respectively. The minimum allowed pavement condition corresponds to the "must level" of 40. Beyond this level, temporary fixes will only retard the rate of deterioration. When a rehabilitation treatment is applied, the pavement rating increases abruptly and a new performance cycle begins.

At this time, only a linear approximation has been implemented to represent such pavement performance curves in PARADIGM. The expected service lives of feasible RAMs are subjective estimates provided by experts and incorporated in SCEPTRE's knowledge base. These service lives enable the derivation of performance curves for each RAM, assuming that upon construction of each strategy the pavement rating returns to about 100. The reason for deriving these pavement performance curves is to enable a visual display of expected pavement deterioration for each RAM to be presented to the user.

Life-cycle costs are determined for each RAM based on unit costs that include construction, maintenance, and user costs. The default values of the unit costs are based on average 1986 WSDOT project costs, and can be interactively modified or updated by the user. Life-cycle costs are then converted to annualized costs per lane mile, which facilitates comparison between RAMs with different service lives, and also projects with different geometries and lengths.

This system also derives the net present value of each feasible RAM. The net present value refers to the net cumulative present value of a series of costs and benefits stretching over time. In this case, the comparison is between the present value of costs for two pavement strategies, generally the strategy to be considered and the do-nothing alternative (continued application of routine maintenance only), and assumes that the present value of benefits in both cases is the same. The net present value so defined should not be the only factor in selecting the most appropriate strategy, but it is an important factor.

Before finishing this analysis, the system passes out data to a RAM database for network optimization. The data can be overwritten or edited for network optimization.

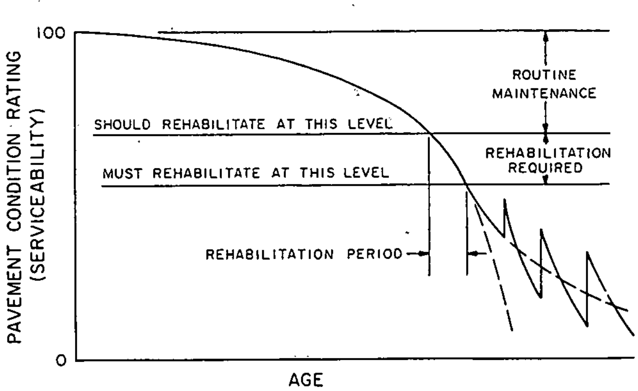


FIGURE B-2 WSDOT pavement performance curve.

Cost Analysis for Overlay Design

Life-cycle cost analysis in OVERDRIVE involves more accurate costing than in the case of SCEPTRE, because an actual design thickness has been determined, whereas the thickness for SCEPTRE's costing is approximate. OVERDRIVE performs this life-cycle costing of each design in comparison with the do-nothing alternative, as in SCEPTRE.

For the overlay, life-cycle costs are based on overlay construction costs, routine maintenance costs, user costs due to traffic delays, and any salvage value at the end of the analysis period. The system can calculate the quantities of these factors by using the overlay thickness and user inputs (overlay length, overlay width, discount rate). Unit costs are again based on those provided by WSDOT. Pavement performance curves are used to model pavement deterioration over time.

This life-cycle cost analysis determines the net present value of the overlay case in comparison with the do-nothing case. The net present value can then be considered by the user in the final decision-making process.

NETWORK OPTIMIZATION

This subsection describes how the project-level results from the rehabilitation model can be used for network-level pavement management analyses. The problem involves selecting the optimal set of RAMs over a network of segments when the next year's total construction budget is constrained. SCEPTRE provides estimates of both construction costs and net present values for feasible RAMs on each segment. The chosen objective in PARADIGM involves maximization of net present value, with the solution indicating those RAMs, if any, that should be constructed on each segment in the next year. Only one RAM can be constructed on any given segment, and partial investments in a RAM are not possible.

This problem can be formulated as an integer program with binary decision variables.

The problem formulation is:

maximize

$$\sum_{i=1}^{m} \sum_{j=1}^{n} NPV_{ij} X_{ij}$$

subject to:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij} \leq B$$

$$\sum_{j=1}^{n} X_{ij} \leq 1 \quad (i = 1,2 \ldots, m)$$

$$X_{ij} = 0 \text{ or } 1 \text{ (for all ij)}$$

where

X_{ij} = decision variable for the jth strategy on segment i
m = total number of segments
n = total number of strategies
NPV_{ij} = net present value for the jth strategy on segment i
C_{ij} = construction cost for the jth strategy on segment i
B = total construction budget

Network optimization is performed by an external C program in PARADIGM. When the user executes the program, the data file containing all relevant segment data must exist. The database is constructed by running SCEPTRE as many times as necessary for the total number of segments to be considered in the analysis. The input data include segment identification, strategy identification, construction cost, and net present value. Before or after running the program, the data can be displayed, modified, or deleted from the system. The total budget is obtained from the user.

Two algorithms are implemented that can produce either an exact solution (by implicit enumeration) or an approximate solution (by Lagrangian relaxation). The approximation can be calculated much faster than the exact solution and is useful for large pavement networks with many segments.

The output of the program is a list of segments and their rehabilitation strategies, if any, that are part of the optimal solution for projects to be constructed in the coming year.

SUMMARY

This section has described the development and structure of version 1.0 of PARADIGM, the Pavement Rehabilitation Analysis and Design Mentor. PARADIGM is a microcomputer-based, integrated set of interacting expert systems and algorithmic models. It includes three major components: SCEPTRE, OVERDRIVE, and Network Optimization.

Case-study applications of PARADIGM have been performed using, in part, actual field data provided by the Washington State DOT (WSDOT). SCEPTRE has been successful in identifying the feasible RAMs and most cost-effective RAM strategy compared with the actual decision of WSDOT.

Although there are variations between the major overlay design methods, the comparative performance of OVERDRIVE has been found to be most encouraging. OVERDRIVE continues to be used on a regular basis in practice by WSDOT.

In terms of future development, the knowledge base in PARADIGM should be expanded to enable data from non-destructive testing methods to be used, and to assist with the design and decision making for rigid pavement rehabilitation. Eventually, PARADIGM should be capable of addressing all existing pavement types and rehabilitation strategies, as well as innovative new approaches and materials. It is desirable for PARADIGM to address overlay material types other than asphalt concrete. The results of case study applications suggest that refinement to allow for limited cracking of the surface course layer, which is more likely for thicker layers, should be investigated.

The expertise in SCEPTRE, in particular, currently emphasizes state-maintained flexible pavements and rehabilitation practices in Washington state. Further research should refine and adapt the knowledge base in PARADIGM for local agencies.

Finally, because PARADIGM is intended to serve as a professional application package, it is desirable for the knowledge base to be continuously refined and validated through a variety of additional new test cases.

CHINA

Computerized Highway Noise Analysis

Roswell A. Harris, and Louis F. Cohn, University of Louisville

INTRODUCTION

The expert system CHINA (Computerized HIghway Noise Analyst) was originally developed to address a nationwide lack of experience in the acoustic design of highway noise barriers (*5*). The initial version of this system was developed on a VAX 11/780 using FRANZ LISP and a development tool called GENIE.

The development of computerized tools for highway noise prediction and barrier design began after years of research dating back as far as 1963 (*5*). Since that first research, many algorithms have been developed and revised which were incorporated into SNAP 1.0, the first program to implement the Federal Highway Administration Model for Traffic Noise Prediction. Further revisions led to the development of STAMINA 1.0 and finally to the implementation of the Barrier Cost Reduction (BCR) procedure that consists of STAMINA 2.0 and OPTIMA.

The BCR procedure leads the design engineer toward a balanced noise barrier design. A balanced design is one that avoids both weak links that sacrifice barrier performance, and excessively strong links that result in an overly expensive design. The development of this procedure is the result of nearly one million dollars of research (*5*).

Figure B-3 shows the BCR flow diagram. The user may choose to loop to step three through an appropriate response to an OPTIMA question. This provides the analyst with a method for revising the barrier design at some or all segments. If the user wishes to return to step two in order to change some of the other input information, such as the design noise level at each receiver, it would be necessary to exit OPTIMA and re-run the program. A loop back to step one may be desirable in some instances where the user needs to change the baseline barrier. In this case, the user must also re-execute STAMINA.

Although a more cost-effective design is not theoretically possible than one obtained by choosing perfectly balanced E/C ratios (effectiveness/cost), such a design is not necessarily desirable. By choosing cost-effectiveness as the only design criterion, the engineer may neglect to consider many important design issues, such as insertion loss goals (or final noise levels) and aesthetics. It is this type of issue or constraint that cannot be implemented into a standard algorithm, and which provides the motivation for the development of new tools, such as an expert system, for the design of an optimal barrier. One of the big advantages of expert systems is the ability to add, delete, and revise the knowledge that governs their behavior. The original implementation of CHINA was intended to be continually expanded with new rules, a feature that would easily be accomplished in such an easy-to-use environment.

DEVELOPMENT APPROACH

Development of the CHINA design tool for the personal computer began with emphasis on the most critical portion: the

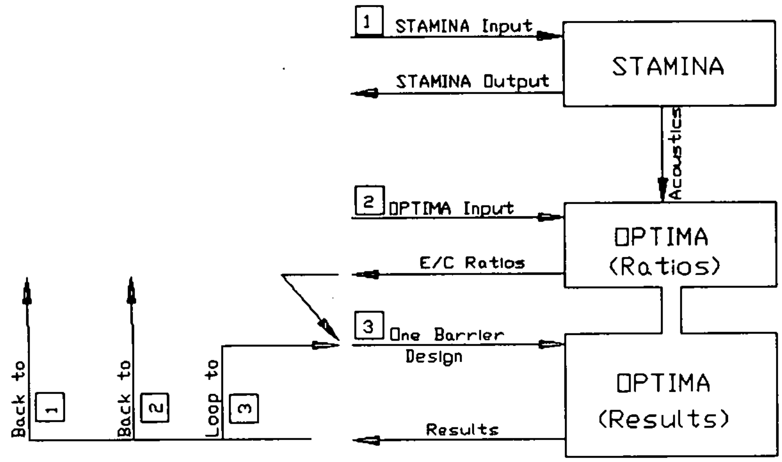


FIGURE B-3 Barrier cost reduction flow diagram.

expert system for barrier design. The idea was to develop a series of routines that could be used to effectively design a noise barrier, test these routines independently of one another, and develop a control structure that would coordinate their activities. Once this portion, which is the true expert system portion of the design tool, was completed, an interface could be developed around it to provide the user with system configuration control, parameter control, and data and graphic interfaces.

The selection of an inference engine/development shell was made prior to the conversion of the rule base to the personal computer. The tool chosen for the project was the EXSYS Expert System Development Package, developed by EXSYS, Inc. of Albuquerque, New Mexico. EXSYS is an expert system shell written in C that encodes knowledge in the form of IF-THEN production rules.

EXSYS was chosen as the rule base development tool for several reasons. First, the rule base is coded in a syntax that resembles natural language (*6*), making the rules easy to understand. Second, EXSYS is relatively inexpensive as compared to other shells, allowing CHINA to be implemented and available at low cost. Third, EXSYS is able to call external programs.

Following the successful coding of the rules in EXSYS, OPTIMA was revised to work smoothly with the expert system implementation. This involved establishing a flag system and importing and exporting parameters and design data. OPTIMA was not programmed to output data in a format directly compatible with EXSYS. Instead, the output was written to an intermediate file along with extra data that might be included in future expansions of the rule base. This allows the rule base to be expanded or the inference engine to be changed without the need to make further revisions to OPTIMA.

The next step built the conversion routine that reads the OPTIMA output data and creates a file that can be read by EXSYS. Another routine had to be developed to convert the EXSYS output data into a barrier design that could be evaluated by the next pass through OPTIMA.

The modules described above were tied together through a DOS batch file completing the development of a prototype system. This prototype was then tested to determine feasibility of the expert system.

Once the prototype was deemed feasible, a control mechanism was developed to execute the expert system. Upon completion of the control program, the user interface was developed to provide file access, parameter maintenance, and a graphic interface. This completed the development of the CHINA design tool.

Figure B-4 shows the hierarchical relationship among program modules in the CHINA design system. Development of these modules was a three-phase project. The first phase, development and testing of the system components, included the definition and configuration of the rule base. Phase two, development of a control structure, coordinated the activities of the system modules. The third phase involved the development of a user interface complete with design and configuration aids. The language chosen for the control module and most other program modules was Pascal. The primary reason for choosing Pascal was to provide programs that can be easily revised by any user. The availability of Turbo Pascal documentation, and its popularity with both advanced and novice programmers, made it an ideal choice. There are several other reasons for choosing Pascal. Turbo Pascal provides access to MS-DOS function calls. This is very desirable for loading and executing non-Pascal files, for redirecting input, and for moving through the directory structure. Flexible file access tools provided in Pascal are of great benefit since the same data must be read and written by FORTRAN, BASIC, and Pascal programs, as well as EXSYS. Finally, Pascal provides many screen formatting tools that allow an aesthetic display to be generated.

External Variables of the Rule Base

The CHINA rule base was designed to operate on data provided by external programs. Therefore, most of the variables contained in the rules are not assigned values in the rule base. They are instead expected to obtain their values from an external source. Furthermore, the CHINA rule base is designed to operate on receiver/barrier-segment pairs. That is, each set of variables contains information on one particular receiver and one particular barrier-segment treated together.

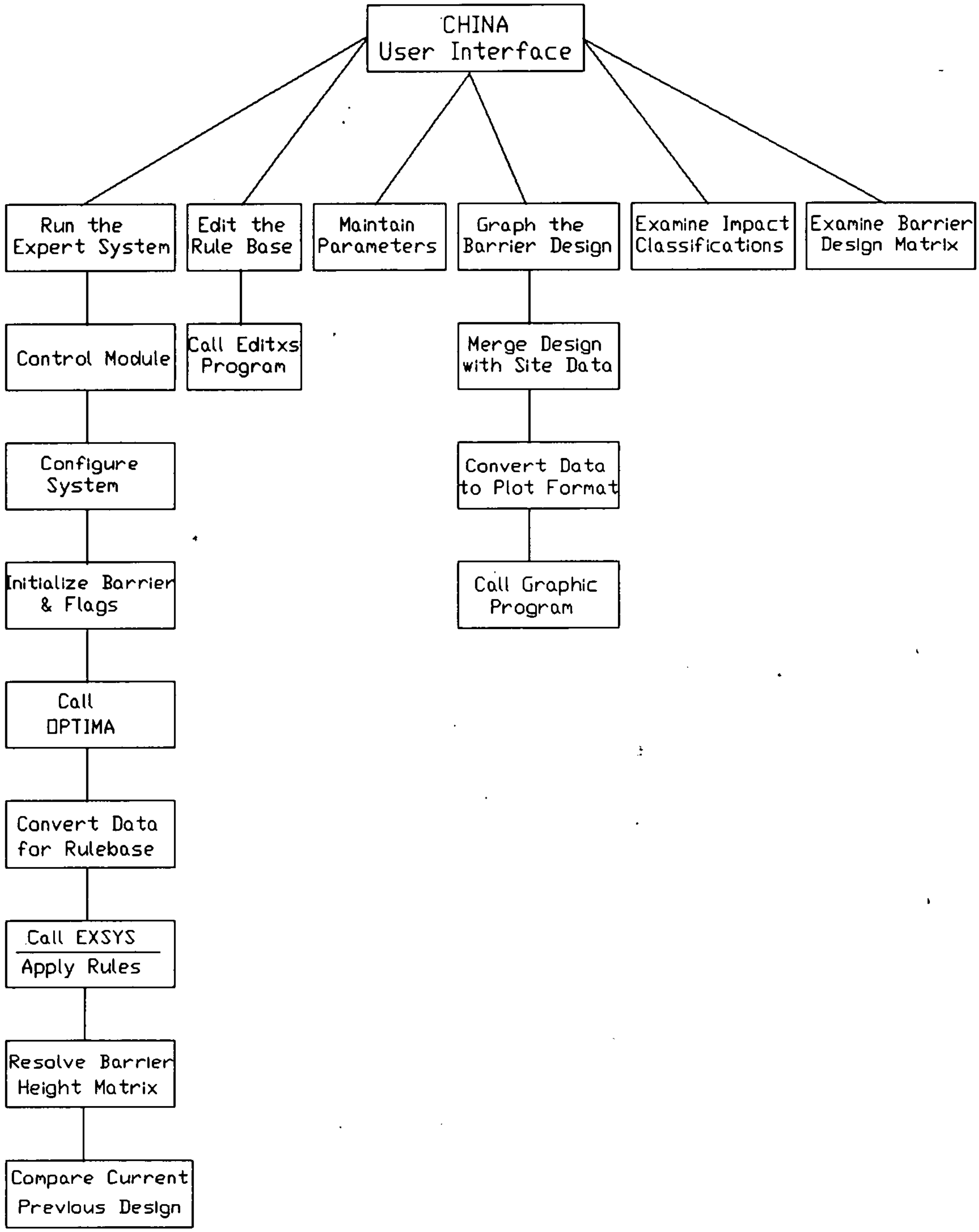


FIGURE B-4 CHINA module hierarchy diagram.

Configuring the Knowledge Base to Read and Write Data

Once the rules were converted and entered, EXSYS was configured to read the variable data from an external source and to write data to an output file. This was a simple procedure which required creating a configuration file and an output specification file.

The configuration file was written to contain the EXSYS datalist command. The syntax of this command is simply

datalist = filename

where filename is the name of a file that contains EXSYS variable names and values. This file is expected to contain a list of variable names, each followed by a value. Each set of data is to be separated by the END marker. Figure B-5 lists a portion of an example datalist file. The example shows values for two receiver/barrier-segment pairs.

When EXSYS encounters the datalist command, it begins reading the file specified until it encounters an END marker or the end of file. All of the variables are read before EXSYS begins to apply rules.

```
[CRITERION 1]              1.5
[CRITERION 2]              6.5
[INSERTION LOSS]   3.8
[SEGMENT NOISE]    50.0
[MAXIM]                    58.0
[HEIGHT]                   5
END
[CRITERION 1]              1.5
[CRITERION 2]              6.5
[INSERTION LOSS]   4.2
[SEGMENT NOISE]    54.5
[MAXIM]                    58.0
[HEIGHT]                   4
END
```

FIGURE B-5 Example datalist file.

The output specification file is created to direct EXSYS to write certain data to an output file at the completion of a pass through the rule base. EXSYS expects this file to contain a filename, which may include flags, and information describing the data to be written. Figure B-6 shows the contents of this file. The first line tells EXSYS to write output data to the file called BARRIER.DAT. The /A flag specifies that the program should append data to the file rather than rewrite it. The second line of the file causes EXSYS to write the value of the variable [HEIGHT] to the file, and the /V flag specifies that the variable is to be written without any associated text. The last line tells EXSYS to run the rule base again after writing the data.

When used in conjunction with a multiple record datalist file, the RESTART command will cause one pass through the rule base for each record of data in the datalist file. Because the append flag (/A) is used, the output data for each record are added to the output file rather than overwriting the file at the end of each pass.

As a result of specifying a multiple record datalist file in the configuration file, and an output file to which EXSYS is to append data in the output specification file, CHINA will read all records in the rulebase file and write the resulting value [HEIGHT] to the file named in the output configuration file (in this case BARRIER.DAT).

Creating the Rule Base Data File

OPTIMA was modified to write important design data to a separate output file. A Pascal routine is used to convert these design data to the rule base format. This conversion utility writes the data to another file, and the CHINA rule base configuration file contains the line

datalist = RULEBASE.DAT

```
FILE barrier.dat /A
[HEIGHT] /V
RESTART
```

FIGURE B-6 Output specification file.

which instructs EXSYS to read variables from this file. The conversion routine also provides two secondary functions.

One of these functions is to write the values of the horizontal and vertical axes of the impact classification guide to a file. This impact file is provided solely as a reference for the user and can be viewed or printed through an option in the CHINA user interface.

The other function writes a file that will replace the standard user input (keyboard) as input to the rule base execution in EXSYS, provided the user has configured the system parameters to do so. This is desirable because EXSYS prompts the user for options at the end of each pass through the rule base. EXSYS does this as part of an explanation module. This allows the user to chain through the rules that fired, thus checking the integrity of the rule base. This EXSYS feature, however, can present a problem. A system with R receivers and BS barrier-segments will make R x BS passes through the rule base, per pass of the design cycle. Thus, if R = 10 and BS = 10, the user would have to respond to the EXSYS prompt 100 times per pass. It is therefore desirable to redirect the standard input to a file that contains 100 Ds, where R x BS = 10 x 10 = 100. The "D" response simply informs EXSYS that the current pass is "Done", and to proceed with the next pass; thus 100 keystrokes by the user can be avoided. The user may specify the name of this file in the parameters option of the CHINA user interface.

In order to understand this conversion algorithm, it is important to understand that the rule base is intended to design a barrier for each receiver, resulting in R barrier designs for a problem with R receivers. Thus, if there are BS barrier- segments, an R x BS matrix is produced by the rule base. This matrix is later resolved into a single barrier design, but the important observation at this point is to note that the rule base expects to operate on R x BS sets of data. That is, it expects to read data for each receiver/barrier-segment pair. It is therefore the task of the design file conversion utility to produce a set of data for each receiver/barrier-segment pair.

Each set of data is separated by the "END" marker. This marker tells EXSYS to quit reading the data and to proceed with applying rules. When it has completed applying the rules, EXSYS will show the results and provide a set of options and a prompt. When the user (or redirected input file) has entered "D" (DONE) at the prompt, EXSYS will read the next set of data.

Development of the Control System

A prototype system was developed to test the fundamentals of the CHINA design process. This system consisted of a data file initialization utility, the revised OPTIMA program, the data conversion utility, the rule base, the height matrix resolution utility, and the utility that compared the current and previous designs. A simple batch file was used as the control structure for executing these modules. Once it was determined that the system was functioning as intended, design of a more sophisticated control structure was paramount. The role of this control module was three-fold: (1) to invoke the programs and maintain data files regardless of their location in the DOS directory; (2) to provide redirected input where desirable; and, (3) to maintain a virtual disk system on those systems that can support it.

During development of CHINA, all programs and data were located in the same DOS directory. However, this may not be desirable for the end user. One example of such a case may be one in which the user chooses to place the EXSYS programs in a unique directory for use with other expert systems. Another case may occur if CHINA operates on data files that are stored on a floppy disk. For such configurations, a control system has to access programs and data wherever they are stored.

In addition to issuing a prompt with each pass through the data, the rule base must read each set of data from disk and write the output to disk. For a large problem, this generates significant disk input/output (I/O), which slows the design process considerably. This deficiency can be reduced by copying the rule base and the rule base variable file to a virtual random access memory (RAM) disk. As a result, all input and output is read from and written to RAM, a process which is significantly faster than disk I/O. Virtual disk maintenance then, is another important task of an effective control system.

Two modules of this control system are discussed below:

Copying the Rule Base To the Virtual Disk

A CHINA parameter, accessible through the user option menu, contains the name of the virtual disk to be used. If the parameter contains the null string, then a virtual disk is not to be used. The control module's first task is to check the value of this parameter. If it is not null, then a function is invoked which performs a byte-by-byte copy of the rule base to virtual disk, provided it is not already there.

Although a byte-by-byte copy is slow, it is used here rather than the Turbo Pascal blockread and blockwrite functions. These functions operate on 128-byte blocks causing some extra bytes to be written to the virtual disk copy of the rule base files. EXSYS interprets these extra bytes as additional command line parameters which it cannot evaluate, causing an execution error. It is, therefore, necessary to use the byte-by-byte copy even though it is slower. Current conditions require approximately 7,000 bytes of data to be copied. Thus, the byte-by-byte copy requires approximately 7,000 read cycles, while a block copy would require only 55. Although this is about 127 times less efficient, read cycles generally only take a few milliseconds. Thus, the delay is not visible to the user.

Calling External Programs

A primary function of the control module is to provide a routine that could load any external program (such as OPTIMA), pass it command line parameters, and redirect the standard input if necessary, without destroying the control program residing in memory. In order to maintain compatibility among various systems, most compilers and libraries do not provide such a function. Thus, one of the first steps in developing the control module was to create a program to carry out this task.

A Pascal procedure was written to accommodate this task. The program receives two parameters from the calling routine. The first is the name of the external function to execute, the second is the command line parameters including redirection arrows. The address of the program to be called is loaded into a variable representing the DS:DX register pair (Intel 8088, 8086 family of microprocessors) (*7*), and the MS-DOS load function (Hex 4B) is loaded into a variable representing the AX register. The command line parameter passed to the routine is examined for redirection arrows. If found, the characters following the arrows are examined for a file name, and the standard input is redirected to come from this file through a process that duplicates the standard input file handle (*8*), and assigns the user specified file to the standard input handle. Once this has been completed, the Turbo Pascal MS-DOS function is invoked to load and execute the external program. This module can be used to load and execute any external program, providing sufficient memory is available.

Developing the User Interface

With the control module completed, the user could invoke an expert system to design a noise barrier; but to make a truly useful tool, other features are necessary. Many of these features have been discussed in the development of the various modules and are simply questions that must be answered in order to run the design system as intended. For example, CHINA must know the answers to the following questions:

1. Where are the programs and data located?
2. Will a virtual disk be used?
3. Is the standard input to be redirected during the rule base pass?

A powerful user interface should also provide design aids such as a graphics interface, design data printouts, and the ability to change crucial design variables and to revise or view the rule base.

These options and parameters were developed into a user interface module that expands the CHINA expert system into a complete design tool. Six options were developed into the user interface. Figure B-7 shows the user interface menu, which is loaded to begin execution of the CHINA system. All other CHINA programs are included in the compiled version of the user interface via the Turbo Pascal INCLUDES directive.

Configuring the System

When the CHINA user interface program is invoked, it immediately reads the parameter file and sets the default parameters.

```
CHINA USER OPTIONS

1      RUN CHINA

2      EDIT RULE BASE

3      CONFIGURE PARAMETERS

4      CONVERT DATA & PLOT DESIGN

5      VIEW IMPACT CLASSIFICATION

6      VIEW BARRIER DESIGN MATRIX


Q      QUIT PROGRAM
```

FIGURE B-7 CHINA user interface.

Two Pascal modules handle these tasks. Every program module in the CHINA system identifies files and programs solely by variable names which are set in these two modules. The first module simply reads the parameters and assigns them to variable names. The other routine examines the parameters and determines full pathnames for data and programs.

Parameter Maintenance

The parameters defined for the CHINA system provide the user with control over the location of programs and data, virtual disk usage, standard input redirection, and crucial control variables. Figure B-8 shows the parameter maintenance screen which is loaded from the CONFIGURE PARAMETERS option of the user interface.

Impact and Barrier Matrix Report

These reports were designed to help the user understand what occurred during the design run. The impact classification report lists the impact classifications both before and after the barrier design. This information can be useful in adjusting the value of CRITERION or the impact classification rules. The barrier matrix report lists the barrier height index matrix that was developed during the final pass. This information is useful in adjusting the barrier-segment significance ratio.

Graphics Interface

The development of programs to plot the barrier site was accomplished independently of the expert system and then integrated with it to complement the design tool. The graphic system comprises two programs. One is a data conversion program that reads the file of data to be plotted and converts it to a format readable by the second program, which is the actual graphics routine. These programs may be called independently of the CHINA system, or through the CHINA user interface.

The graphics interface has three functions: (1) it merges the new barrier-segment heights designed by CHINA with the original design problem data file that contains the traffic volumes and coordinates of the roadways, baseline barrier, and receivers; (2) it calls the Turbo BASIC program CONVERT, which converts these data to the format required by the graphics program; and, (3) calls the graphics program to plot the final barrier design and site.

The first task is performed through a routine that reads the newly created height data from the appropriate file, and merges it with the original site data file. The actual merger occurs when the program reads the section describing the baseline barrier from the original data file. When it reads the data, it substitutes points determined by the new barrier design for those points contained in the original data file for the baseline barrier.

When the first step is complete, the graphic interface invokes the conversion program. This program reads the input file and prepares the data needed by the graphics program. The last task of the graphic interface is to invoke the graphics routine.

Screen Formatting Tools

Several screen formatting tools were developed to aid the visual representation of the program modules and provide a user-friendly environment. These tools were primarily text positioning and box drawing tools.

SUMMARY

A number of expert systems "shells" are available that assist the user in organizing, building, and implementing their own rule-based expert system for a particular application. However, beyond simply applying the rule base to a given problem, these shells do not provide much help in developing an interface that all potential users of the system would find friendly. A significant feature of CHINA is the use of a flexible and useful interface to an expert system that is integrated with an existing design tool, OPTIMA. This feature results from the development of a structure that controls operation of the expert system and required external programs, as well as a user interface that permits the user full access to the power of the system.

One of the primary functions of the control structure is to invoke required programs. This routine is able to load any external program, pass to it command line parameters, and redirect the standard input without destroying the control program residing in memory. It also allows the user to redirect input, if desired,

```
                    CHINA PARAMETERS

 1  CRITERION                          60
 2  OPTIMA ACO FILE                    optima.aco
 3  RULE BASE STANDARD INPUT           rbinput.dat
 4  RULE BASE NAME                     china
 5  RULE BASE HOME DIRECTORY           c:\exsys
 6  VIRTUAL DISK ID                    e:
 7  OPTIMA PROGRAM NAME                opt4.exe
 8  INFERENCE ENGINE                   c:\exsys\exsys.exe
 9  BARRIER SEGMENT SIGNIF. RATIO      0.10
10  IS COLOR MONITOR USED? (Y/N)       Y
11  RULE BASE EDITOR                   c:\exsys\editxs.exe
12  LOCATION OF STAMINA PLOT PRGRM     c:\stam2
```

FIGURE B-8 CHINA parameter maintenance screen.

and provides the capability to use a virtual disk on those systems which can support it.

Other features were designed into a user interface that make the system a truly useful tool. Some of these features are simply questions (contained in a parameter menu) that must be answered in order for the system to run as desired. For example, the user is able to specify the location of programs and data, or whether a virtual disk is available. The interface also provides a graphics routine that aids the user in visualizing the completed barrier design. This routine merges the CHINA barrier design into the original input file, converts that input file into a format required by the plotting program, and then calls the plotting program to present the final barrier design and site.

The KBES described in this section provides a complete, flexible, easily used method for accessing an expert system in a microcomputer environment, as well as a convenient means of managing input/output used by the system. The modules that make up this control structure provide the user with a friendly interface, as well as providing a means to completely control operation of the system. CHINA has been designed to be flexible, so that as the knowledge base grows, or the KBES is otherwise enhanced, these changes can be easily accommodated.

A version of CHINA was developed for the New Jersey DOT, and is operational in that agency. A more generic version is in wide use throughout the country, as part of the *NOISE* software library that is distributed through the University of Louisville.

APPENDIX B REFERENCES

1. Ritchie, S. G., C. Yeh, J. P. Mahoney, and N. C. Jackson, "Development of an Expert System for Pavement Rehabilitation Decision Making," *Transportation Research Record 1070*, Transportation Research Board, National Research Council, Washington, D.C. (1986).
2. Ritchie, S. G., "Expert Systems in Pavement Management," *Transportation Research*, Vol. 21A, No. 2, Pergamon Press, Oxford, U. K. (1987).
3. Harris, R. A., L. F. Cohn, and W. Bowlby, "Designing Noise Barriers Using the Expert System CHINA," *Journal of Transportation Engineering*, American Society of Civil Engineers, New York, NY, Vol. 113, No. 2 (March 1987) pp. 127–138.
4. Asphalt Institute, *Asphalt Overlays for Highway and Street Rehabilitation*, Manual Series No. 17 (MS 17) Lexington, KY (1983).
5. Ritchie, S. G. and J. P. Mahoney, "Development and Performance of an Expert System for Pavement Overlay Design," *Civil Engineering Systems*, Vol. 6, Chapman and Hull, London, U. K. (1989).
6. Wigan, W. R., "Knowledge Based Systems Tools on Microsystems," Research Report AAR No. 134, Australian Road Research Board, Victoria, Australia (1985).
7. Morse, S., *The 8086/8088 Primer*, Hayden Book Company, Hasbrouck Heights, NJ (1982) pp. 10–40.
8. Jump, D. N., *Programmer's Guide to MSDOS for the IBM PC*, Brady Communications Company, Inc., New York, NY (1984) pp. 38–58.

APPENDIX C
RESOURCE GUIDE FOR KBES AND AI SOFTWARE

Knowledge Based Expert Systems and Artificial Intelligence related software is available from a wide variety of vendors. The list included in this appendix contains information on more than forty such vendors; some of the included information was extracted from the January 1991 issue of Byte magazine. Inclusion in the list does not imply any endorsement of the vendor or product; exclusion from the list does not imply anything negative about a vendor or product.

Advanced A.I. Systems, Inc.
Mountain View, CA

AICorp, Inc.
Waltham, MA

AI Ware, Inc.
Cleveland, OH

Artificial Intelligence Research Group
Los Angeles, CA

Artificial Intelligence Technologies, Inc.
Hawthorne, NY

Beacon Expert Systems, Inc.
Brookline, MA

Bell Atlantic
Princeton, NJ

California Intelligence
San Francisco, CA

Carnegie Group, Inc.
Pittsburgh, PA

Cognition Technology
Cambridge, MA

Comdale Technologies, Inc.
Toronto, Ontario

Digital Equipment Corp.
Maynard, MA

Emerald Intelligence, Inc.
Ann Arbor, MI

Experience in Software, Inc.
Berkeley, CA

Expertech, Inc.
Nevada City, CA

ExperTelligence, Inc.
Goleta, CA

Expert Systems Design, Inc.
Berkeley, CA

EXSYS, Inc.
Albuquerque, NM

Hyperpress Media Lab
Foster City, CA

IBM
Armonk, NY

ICAD
Cambridge, MA

If/Then Solutions
Palo Alto, CA

Information Builders, Inc.
New York NY

Intellicorp
Mountain View, CA

IntelligenceWare, Inc.
Los Angeles, CA

Intelligent Environments
Tewksbury, MA

Lucid, Inc.
Menlo Park, CA

Millenium Software
Laguna Beach, CA

Nestor, Inc.
Providence, RI

Neuron Data
Palo Alto, CA

Orphic Systems
Philadelphia, PA

OXKO Corp.
Annapolis, MD

Paperback Software International
Berkeley, CA

Perceptics Corp.
Knoxville, TN

ROSH Intelligent Systems
Needham, MA

Software Architecture and Engineering, Inc.
Arlington, VA

Symbolics, Inc.
Burlington, MA

Symbologic Corp.
Redmond, WA

Togai Infra-Logic, Inc.
Irvine, CA

APPENDIX D
SURVEY QUESTIONNAIRES

NATIONAL RESEARCH COUNCIL

Transportation Research Board

2101 Constitution Avenue Washington, D.C. 20418

NCHRP Synthesis Topic 22-09
Knowledge Based Expert Systems In Transportation
Authors:
Louis F. Cohn and Roswell A. Harris
University of Louisville

Address Reply to:
Louis F. Cohn
University of Louisville
Department of Civil Engineering
Louisville, KY 40292
ph. (502) 588-6276/fax 588-7033

QUESTIONNAIRE TO:
STATE DEPARTMENTS OF TRANSPORTATION

1. Does your department currently utilize Knowledge Based Expert Systems (KBES) in any way? If yes, what does each application do? We would appreciate your sending any documentation (reports, papers, etc.) that may be available.

   NOTE: This synthesis will emphasize transportation planning, design, analysis, and engineering in the areas of roadways and structures. Less emphasis is to be given to financial and management KBES applications.

2. Has your department sponsored any research related to KBES development? If so, please briefly describe each project. If not, do you contemplate sponsoring any such research? We would also appreciate receiving any documentation that may be available.

3. For any project mentioned in question (2), assess its progress towards implementation based on the following:

   1 - Conceptual
   2 - Prototype under development
   3 - Prototype developed and under testing
   4 - Detailed KBES under development
   5 - Finished KBES in use
   6 - Project terminated (give reasons)

4. List and describe the hardware and software currently housed within your department that is used in KBES activity, if any.

5. List up to five problems in transportation that you think have exceptional potential for KBES applications. Be as specific as possible.

We appreciate your willingness to participate in this survey and NCHRP synthesis project. Please send your response to Professor Cohn at the above address. If you have any questions or comments, feel free to call Professor Cohn or Professor Harris.

NATIONAL RESEARCH COUNCIL

Transportation Research Board

2101 Constitution Avenue Washington, D.C. 20418

NCHRP Synthesis Topic 22-09
Knowledge Based Expert Systems In Transportation
Authors:
Louis F. Cohn and Roswell A. Harris
University of Louisville

Address Reply to:
Louis F. Cohn
University of Louisville
Department of Civil Engineering
Louisville, KY 40292
ph. (502) 588-6276/fax 588-7033

Questionnaire to:
TRB and ASCE Committees on Expert Systems

1. Do you currently utilize Knowledge Based Expert Systems (KBES) in transportation in any way? If so, what does each application do? We would appreciate your sending any documentation (reports, papers, etc.) that may be available.

    NOTE: This synthesis will emphasize transportation planning, design, analysis, and engineering in the areas of roadways and structures. Less emphasis is to be given to financial and management KBES applications.

2. Have you or your organization sponsored any research related to KBES development in transportation ? If so, please briefly describe each project. If not, do you contemplate sponsoring any such research?

3. For any project mentioned in question (2), assess its progress towards implementation based on the following:

   1 - Conceptual
   2 - Prototype under development
   3 - Prototype developed and under testing
   4 - Detailed KBES under development
   5 - Finished KBES in use
   6 - Project terminated (give reasons)

4. List and describe the hardware and software you currently use in KBES activity, if any.

5. List up to five problems in transportation that you think have exceptional potential for KBES applications. Be as specific as possible.

We appreciate your willingness to participate in this survey and NCHRP synthesis project. Please send your response to Professor Cohn at the above address. If you have any questions or comments, feel free to call Professor Cohn or Professor Harris.

THE TRANSPORTATION RESEARCH BOARD is a unit of the National Research Council, which serves the National Academy of Sciences and the National Academy of Engineering. It evolved in 1974 from the Highway Research Board, which was established in 1920. The TRB incorporates all former HRB activities and also performs additional functions under a broader scope involving all modes of transportation and the interactions of transportation with society. The Board's purpose is to stimulate research concerning the nature and performance of transportation systems, to disseminate information that the research produces, and to encourage the application of appropriate research findings. The Board's program is carried out by more than 270 committees, task forces, and panels composed of more than 3,300 administrators, engineers, social scientists, attorneys, educators, and others concerned with transportation; they serve without compensation. The program is supported by state transportation and highway departments, the modal administrations of the U.S. Department of Transportation, the Association of American Railroads, the National Highway Traffic Safety Administration, and other organizations and individuals interested in the development of transportation.

The National Academy of Sciences is a private, nonprofit, self-perpetuating society of distinguished scholars engaged in scientific and engineering research, dedicated to the furtherance of science and technology and to their use for the general welfare. Upon the authority of the charter granted to it by the Congress in 1863, the Academy has a mandate that requires it to advise the federal government on scientific and technical matters. Dr. Frank Press is president of the National Academy of Sciences.

The National Academy of Engineering was established in 1964, under the charter of the National Academy of Sciences, as a parallel organization of outstanding engineers. It is autonomous in its administration and in the selection of its members, sharing with the National Academy of Sciences the responsibility for advising the federal government. The National Academy of Engineering also sponsors engineering programs aimed at meeting national needs, encourages education and research, and recognizes the superior achievements of engineers. Dr. Robert M. White is president of the National Academy of Engineering.

The Institute of Medicine was established in 1970 by the National Academy of Sciences to secure the services of eminent members of appropriate professions in the examination of policy matters pertaining to the health of the public. The Institute acts under the responsibility given to the National Academy of Sciences by its congressional charter to be an adviser to the federal government and, upon its own initiative, to identify issues of medical care, research, and education. Dr. Kenneth I. Shine is president of the Institute of Medicine.

The National Research Council was organized by the National Academy of Sciences in 1916 to associate the broad community of science and technology with the Academy's purposes of furthering knowledge and advising the federal government. Functioning in accordance with general policies determined by the Academy, the Council has become the principal operating agency of both the National Academy of Sciences and the National Academy of Engineering in providing services to the government, the public, and the scientific and engineering communities. The Council is administered jointly by both Academies and the Institute of Medicine. Dr. Frank Press and Dr. Robert M. White are chairman and vice chairman, respectively, of the National Research Council.

Transportation Research Board
National Research Council
2101 Constitution Avenue, N.W.
Washington, D.C. 20418

ADDRESS CORRECTION REQUESTED