Using Expert Systems To Select Traffic Analysis Software

EDMOND CHIN-PING CHANG

An experimental expert system was developed by the Texas Transportation Institute to assist users in selecting computerized software packages currently being supported by the FHWA. This system was designed to investigate potential expert systems applications in transportation engineering. This study was performed to serve three basic purposes: to test the feasibility of developing a small-scale traffic engineer knowledge-based expert system using a simple knowledge engineering tool, to develop an alternative method of recommending computer programs for user-specified applications, and to investigate a possible approach for implementing advisory expert systems to be operated in the IBM PC/XT/AT microcomputer environment. The development of a prototype expert system using a commercially available knowledge engineering tool developed by Level Five Research Incorporated is described. INSIGHT 2+ was used to experiment with expert system programming in the inexpensive microcomputer environment. This system reviews and analyzes user-input information, evaluates it with various reasoning paths, and offers a conclusion. With the proper combination of knowledge programming tools and preidentified decision-making processes, individual users can develop their applications faster than if they had to learn complex artificial intelligence programming languages. It is recommended that the expert advisory system design concept of this prototype model be extended to assist practicing traffic engineers in selecting software packages to optimize traffic control strategies. With proper improvements, this type of expert system design can assist the user as a stand-alone expert advice system.

Several traffic engineering programs are currently being supported by the FHWA *(1–5)*. A prototype expert system was developed at the Texas Transportation Institute to assist in the selection of these microcomputer software packages. The intent was to apply the expert systems concept to assist individual users in selecting computer software for optimum traffic analysis *(6–10)*. This system was also used to *(a)* investigate the potential feasibility of using expert systems in traffic engineering and *(b)* computerize the expert system's advice through artificial intelligence techniques *(11–16)*.

The development of a prototype expert system is described. This study was performed to serve three basic purposes: to test the feasibility of developing a small-scale traffic engineer knowledge-based expert system using a simple knowledge engineering tool, to develop an alternative method of recommending computer programs for user-specified applications, and to investigate a possible approach for implementing advisory expert systems in the IBM PC/XT/AT microcomputer environment. INSIGHT 2+, a commercially available knowledge engineering tool developed by Level Five Research, Inc., was selected, because of the simplicity of its implementation, to develop experimental expert systems in the IBM PC/XT/AT microcomputer environment *(17, 18)*. Normally, LISP- or PROLOG-based expert systems are adequate for customized problem solving, but knowledge engineering tools such as the INSIGHT 2+ system can allow users to quickly represent specialized knowledge by following predetermined guidelines.

This prototype TTI-FHWA expert system reviews and analyzes the information given by the user, evaluates it with various paths of reasoning, then offers a conclusion for a particular application *(19–21)*. The technical information was collected from various traffic engineering computer programs available from or being developed by the FHWA. For each program package, information is processed on the basis of the user's inquiry, expected performance, development status, hardware requirements, and the available information source. This study is intended to provide an advice system for recommending suitable traffic engineering software. It is a stand-alone expert advice system that uses a microcomputer.

TRAFFIC ENGINEERING SOFTWARE

As urban traffic demands increase, the most efficient coordination is required between existing traffic control devices and proper signal-timing settings. A large number of computer software packages have been developed to provide better traffic analysis. Microcomputers are increasingly available, and numerous programs are available for traffic engineering applications. Without having to access a mainframe computer, a traffic engineer at virtually any location can routinely and efficiently analyze traffic engineering problems and thereby have more time for innovative engineering analysis. As more traffic professionals begin to apply these traffic engineering–oriented computer packages, programming development is being emphasized at the federal, state, and research community levels to enhance the problem-solving capabilities of these packages. However, because of the numerous enhancements being made in each program, it is hard to keep track of developments in each of these traffic analysis packages. Therefore there is an increasing need for an expert advice system to assist the end user in selecting proper tools for specific kinds of analysis.

The traffic engineering software packages analyzed are supported and maintained by the Systems and Software Support Team of the FHWA. The technical information used is based on

Texas Transportation Institute, Texas A&M University System, College Station, Tex. 77843.

```
MAXBAND - MAXimum BANDwidth

LATEST AVAILABLE VERSION:  A time-space diagram was recently added to the
  original (and, to date, only) version of the program.  The program is also
  available in FORTRAN-77.

PROGRAM DEVELOPERS: Dr. John Little and Mr. Mark Kelson, Operations Research
  Center, Massachusetts Institute of Technology.

FUNCTION:  Develops signal timing plans for arterial streets by
  maximizing the sum of the green bands (in both directions).  Will optimize
  cycle length, phase sequence and offsets.  Uses a linear programming approach
  which guarantees that the best possible mathematical solution is found.

COMPUTER REQUIREMENTS:  Available from FHWA only on magnetic tape for 32-bit
  systems with double precision arithmetic.  Requires 400K memory when
  overlaid.  A microcomputer version is commercially available.

CONTACT FOR TECHNICAL SUPPORT:  Dr. Stephen Cohen, FHWA, Office of Safety and
  Traffic Operations R&D (HSR-10), 6300 Georgetown Pike, McLean, VA. 22101
  (703) 285-2091.

TRAINING AVAILABILITY:  None.

FUTURE PLANS:  A research study is being completed that examined ways of
  determining the proper weighting of opposing bands.  A new version of the
  program with improved output formats will be released in 1986.  A new User's
  Manual will be available with the new version of the program.
```

FIGURE 1 Example of material used for MAXBAND model *(1)*.

the manual released by the FHWA in 1986. Figure 1 is an example of the material used in the MAXBAND program. Basically, these computer software packages include traffic signal–timing optimization programs, traffic flow simulation models, and other traffic engineering computer software. These computer programs and their particular areas of application fall under the following three categories *(1)*:

1. Signal-timing optimization programs *(2, 3)*
   - SOAP—Signal Operation Analysis Package
   - MAXBAND—Maximum Bandwidth Optimization
   - AAP—Arterial Analysis Package
   - TRANSYT-7F—Traffic Network Study Tool
   - SIGOP-III—Signal Optimization Model
2. Traffic flow simulation models *(4)*
   - NETSIM—Network Simulation Model
   - TRAFLO—Macroscopic Urban Network Model
   - FRESIM—Freeway Simulation Model
   - ROADSIM—Two-lane, Two-way Rural Road Model
3. Other traffic engineering software *(10)*
   - ITDS—Integrated Traffic Data System
   - HIGHWAY CAPACITY—1985 Highway Capacity Manual Software
   - PPD—Platoon Progression Diagram
   - COUNTS-PC—Signal Warrants Analysis
   - LINKOD—Origin-Destination Table Synthesis

EXPERT SYSTEMS DESIGN

An expert system is a collection of computer programs or systems that applies in a specialized domain. The expert system combines both problem-solving and knowledge-support components for specific applications. The expert systems support environment helps the user interact with the program to specify the requirements and the specialized expertise of an expert in the field *(11–13, 19, 20)*.

There are five major components of artificial intelligence (AI) applications in expert system (ES) designs: *(a)* expert system, *(b)* domain expert, *(c)* knowledge engineer, *(d)* expert systems building tool, and *(e)* end user. Figure 2 shows the basic AI/ES components and their relationships to each other *(13)*. The domain or area expert is an articulate, knowledgeable person with a reputation for producing good solutions to problems in a particular field. The knowledge engineer is usually a person with a background in computer science and AI technology who knows how to build expert systems. The knowledge engineer interviews the domain experts, organizes the knowledge, decides how it should be represented in the expert system, and may assist in development of a specific program. The expert systems building tool is the computer-programming environment and language used by the knowledge engineer or computer programmer to build the expert system. The user or the end user is the person for whom the expert system is developed.

As indicated in Figure 2, the user may be a traffic engineer debugging the expert systems building tool or language, a knowledge engineer refining the existing knowledge in the system, a domain expert adding new knowledge to the system, the end user relying on the system for advice, or clerical personnel adding more information to the knowledge engineering data base. A knowledge engineer converts a domain expert's specialized knowledge into sets of IF-AND-THEN-ELSE rules using instructions that a computer understands. However, no matter what software or hardware the expert system has to use, the knowledge-based expert system ultimately has to be implemented on computer hardware in a

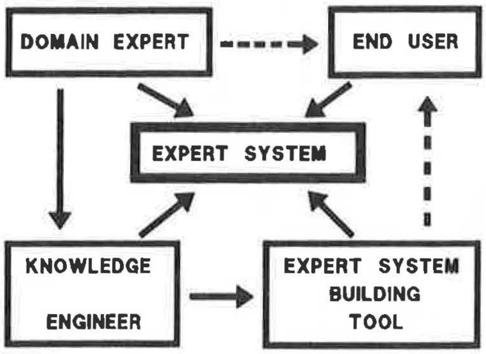

FIGURE 2 Main components of expert systems design.

most primitive machine language format. Most conventional programming is done in high-level languages, such as BASIC, COBOL, FORTRAN, PASCAL, LISP, or C. AI languages are used in ES designs for processing user-input information to derive conclusions and recommendations. Problem-solving AI languages such as LISP and PROLOG are often used.

The evolution of AI applications is shown in Figure 3 *(12)*. As indicated, a more application-oriented research trend has recently become evident. AI/ES programming development can be separated into three areas: expert systems tools, natural language queries, and AI languages. Normally, AI programming is done in LISP and PROLOG. LISP (LISt Processing) is particularly suited for symbolic and numeric processing for decision analysis. LISP is most suitable for manipulating lists of symbols (i.e., strings of numbers or words, or both). For years, LISP has been preferred by AI engineers in the United States. On the other hand, PROLOG (PROgramming in LOGic) is preferred in Europe and Japan. PROLOG contains structures more suitable for writing programs that evaluate logical expressions, whereas LISP contains operators that facilitate the creation of programs that manipulate lists for representing specific expert knowledge.

KNOWLEDGE ENGINEERING TOOLS

Knowledge engineering shells or tools are often used to build expert systems. These tools provide all of the features needed in an expert system, such as help functions, windowing capabilities, graphics support, and other functions, to help the

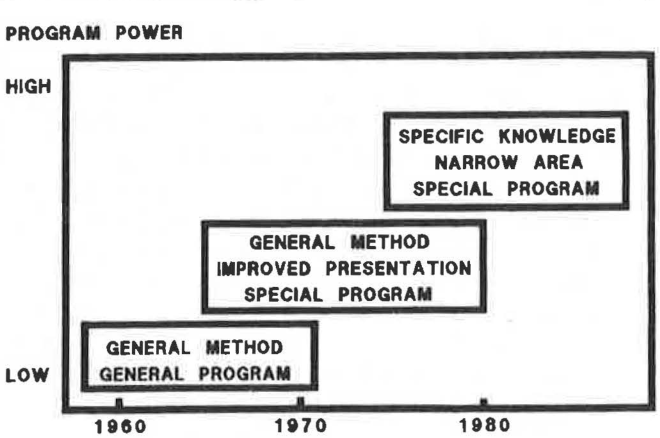

FIGURE 3 Evolution of AI research trend.

knowledge engineer add the information from the domain expert. The knowledge shell usually includes an explanation subsystem that describes the logical steps needed to reach a conclusion. The natural language interface can further help explain these programming development steps in ordinary English to enhance understanding of the decision-support process. Symbolic ES operations using LISP and PROLOG usually take up a lot of computing memory and thus may be executed slowly, particularly on microcomputers. The symbolic operations performed in LISP or PROLOG are usually implemented in more efficient computers using LISP as the operating system.

Knowledge engineering tools allow users to develop a prototype of a defined problem quickly and develop their own customized applications in less time than it takes an AI programming language. However, PC-based expert system development tools are not suitable for large-scale ES application. The common practice for developing expert systems is to obtain a commercially available LISP machine to use fast-executing knowledge engineering tools, such as ART, KEE, Knowledge Craft, or EMYCIN, on a VAX-type super-minicomputer for experimental program development *(13, 14)*. Then the developed expert system can be transferred to generate ES programs for practical applications that may eventually be run on a personal computer in a microcomputer environment.

INSIGHT 2+, developed by Level Five Research, Inc., is a microcomputer-based tool that allows prototype ES programming *(17, 18)*. It is used to apply knowledge, form conclusions from facts, and solve problems in small-scale applications. Unlike ordinary data base systems that merely store, organize, and recall information, the INSIGHT 2+ expert system reviews and analyzes the information given, evaluates it using various paths of reasoning, and offers recommendations. INSIGHT 2+ provides a programming environment for the design, creation, and use of knowledge systems. INSIGHT 2+ permits the use of natural language to develop knowledge data bases and to interact in an IBM PC/XT/AT microcomputer environment.

INSIGHT 2+ can be used to implement user expertise for developing problem-solving techniques. Because it can make inquiries and maintain a knowledge base, expert systems developed using INSIGHT 2+ may enhance the end user's ability to analyze a problem and achieve suitable solutions. For example, by accumulating the answers and probability of success from the user's trial-and-error process, the expert system can summarize the modified solutions. During program execution, INSIGHT 2+ automatically questions the user for information for better conclusions. Using the expertise obtained from the user and programmed in its knowledge base, INSIGHT 2+'s inference engine is able to reason, even from incomplete or uncertain information. By asking the user to specify a confidence value, INSIGHT 2+ can further evaluate the viability of a path of reasoning or a chain of rules depending on the probability of a certain line of reasoning. At any point during a consultation, the user can request an explanation of the current reasoning status or, optionally, wait until the conclusion of a session and obtain a complete trace report. The functional structure of the INSIGHT 2+ system is shown in Figure 4.

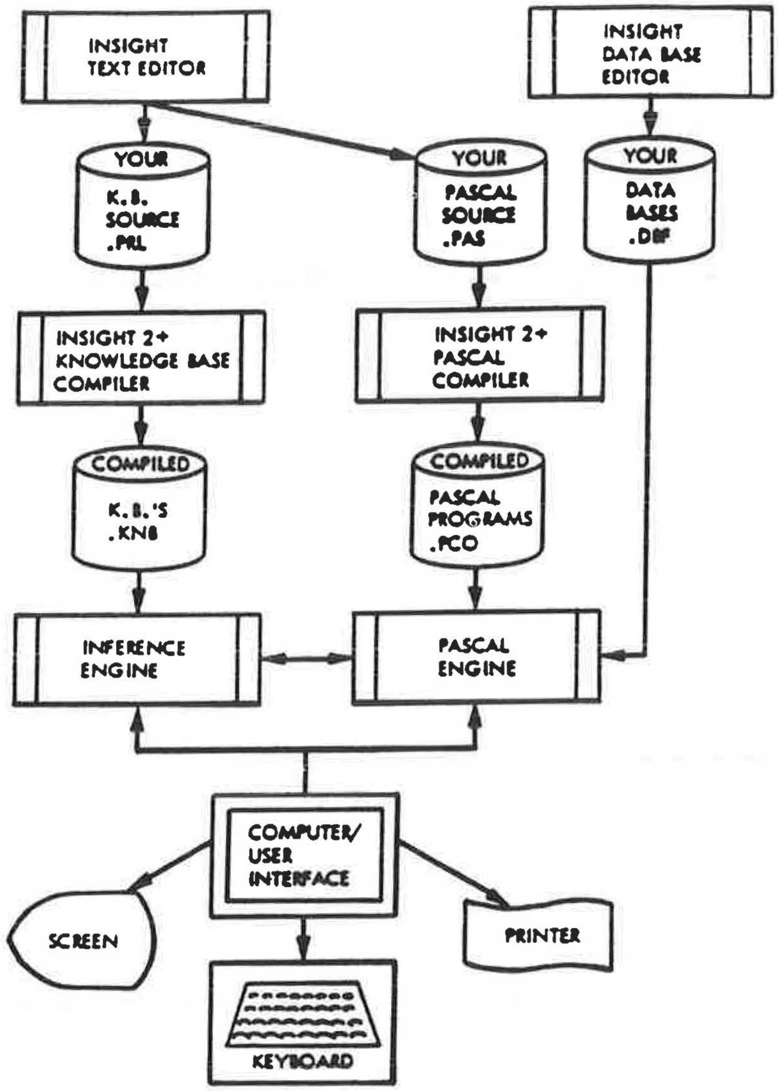

FIGURE 4 Functional structure of the INSIGHT 2+ system *(17, 18).*

BASIC DESIGN PROCESS

A prototype expert system was implemented to provide computerized assistance in selecting traffic engineering software. In this automated selection guide, three main goals or categories of each computer software package are evaluated. These packages include traffic signal–timing optimization, traffic flow simulation, and other traffic engineering analyses. Each computer package is treated as a subgoal. The knowledge engineering tool summarizes the final recommendations of the traffic engineering software from the user-input requirements in this experimental system. The relevant selective information, such as software functions and computer requirements, was extracted from the information source to identify the conclusion of the production rule. It was then used to program the INSIGHT 2+ production rule language (PRL). The displayed information and the proper conclusions were further explained in this expert system.

In this section is described the basic design process used in preparing this prototype expert system for the selection of computerized traffic analysis software packages. The development process used in this ES design can be illustrated by the seven steps shown in Figure 5: extracting basic information, designing a decision table, setting up evaluation goals, selecting evaluation constraints, developing a PRL, debugging the PRL, and completing the program recommendations.

Extract Basic Information

Information used to develop this prototype expert system was extracted from three sources: *(a)* the second issue of the Traffic Software Users Awareness Report, *(b)* the BASIC program developed and demonstrated by the FHWA at the 1986 Annual Meeting of the Transportation Research Board, and *(c)* the working experience of the Traffic Operations Program of the Texas Transportation Institute. The Traffic Software Users Awareness Report, issued semiannually to all users who have received any of the FHWA's traffic engineering software, contains the latest information on FHWA software. The Systems and Software Support Team in the Office of Traffic Operations of the FHWA is responsible for distributing all of the transportation– and traffic engineering–related computer software.

To simplify the discussion, the traffic engineering software discussed here is classified in three categories. These FHWA-

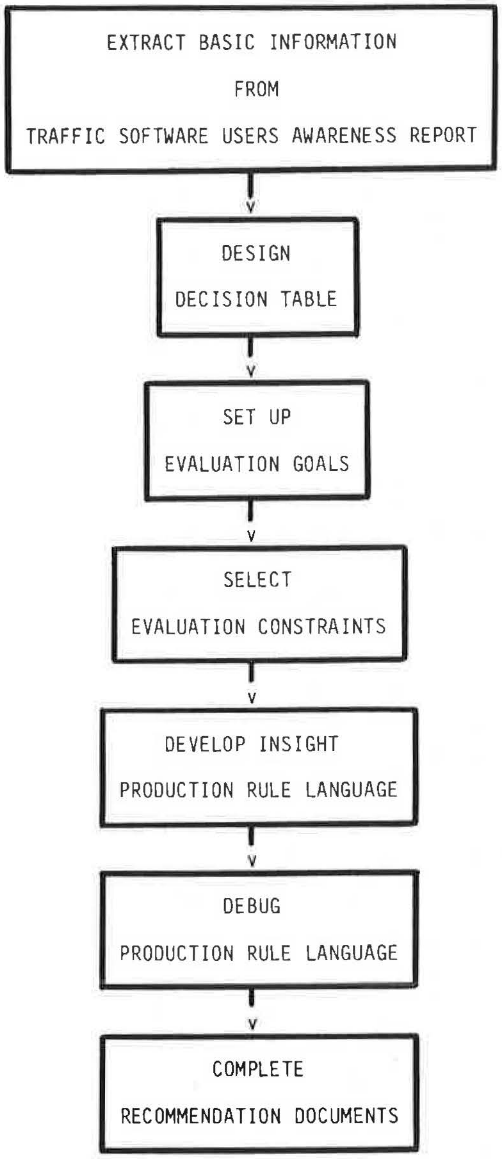

FIGURE 5 Basic design process.

supported packages include the traffic signal–timing optimization programs, traffic flow simulation models, and other traffic engineering analysis software. These traffic analysis packages include some of the popular traffic engineering tools, component models of the TRAF family, step-down versions of the transportation planning tools, and the Highway Capacity Manual software available at present. The relevant information for each of the traffic models was extracted from the Traffic Software Users Awareness Report and summarized. Summaries were based on careful reading of the article, comparing different traffic engineering software, and differentiating special characteristics according to engineering judgment and the criteria stated in the Traffic Software Users Awareness Report. On the basis of the results of the comparisons, the current FHWA-supported traffic engineering software packages were divided into three categories:

- Signal-timing optimization programs: Signal-timing optimization programs optimize the major signal-timing variables, such as cycle length, phase length, sequence, and offset. The transportation facilities analyzed may include isolated intersections, arterial streets, and networks. It should be noted that most of these computer programs, with the exception of MAXBAND, can also perform certain simulation functions for evaluating existing conditions.
- Traffic flow simulation models: Traffic flow simulation models are designed to simulate different traffic control strategies. These models can simulate various types of transportation facilities. Transportation facilities may include isolated intersections, arterial streets, networks, rural highways, and urban highways.
- Other traffic engineering software: Other traffic engineering software, supported by the FHWA, may perform specialized functions, such as computerized data base management, highway capability analysis, time-space analysis, traffic flow profile display, traffic signal warrant analysis, and origin-destination travel analysis.

After separating all of the relevant and irrelevant information from the available technical material, the domain expert traffic engineer modifies the design of the decision-making process according to his past experience with these models. He then determines what information should be emphasized for analysis and identifies factors to be used to evaluate the relative importance of criteria used in the practical design of the prototype expert system.

Design Decision Table

Decision table analysis is a decision-making aid that is used in design and evaluation *(20)*. Table 1 gives a simplified version of the decision table used for studying two basic design elements of this particular expert system. In the decision table, the horizontal components represent the main goals and subgoals defined in the expert system. Horizontal components on the first level in the table are the main goals. Horizontal components on the second level are the subgoals. The vertical components of the decision table represent the facts and rules, such as design constraints and potential application areas, that could be investigated by each of the traffic analysis packages.

Table 1 is an example of how the decision table was applied to analyze the MAXBAND program. The main goal and subgoal of MAXBAND are first identified by applying the backward-chaining concept. Then the major constraints of MAXBAND are separately identified. After the goal, subgoal, and design constraints are summarized, the production rules are specified for actual program development. The major advantages of using this type of decision table in the development of a practical expert system include the ability to

1. Summarize the basic relationships of different constraints.
2. Evaluate the requirements of independent constraints.
3. Study the detailed interrelationships among major variables in a systematic approach. (It should be noted that the original table used in the actual design of this prototype expert system is more complex than the one presented here.)
4. Provide the domain expert's knowledge and skill in completing the background information required in the decision-making process.
5. Set up evaluation goals, subgoals, and design constraints.

TABLE 1 SIMPLIFIED DECISION TABLE ANALYSIS

	SIGNAL TIMING OPTIMIZATION PROGRMAS					TRAFFIC FLOW SIMULATION MODELS				OTHER TRAFFIC ENGINEERING SOFTWARE				
	SOAP	MAXBAND	AAP	TRANSYT-7	SIGOP-III	NETSIM	TRAFLO	FRESIM	ROADSIM	ITDS	CAPACITY	PPD	COUNTS-PC	LINKOD
I. APPLICATION AREAS														
A. SIGNAL TIMING PLAN	X	X	X	X	X	X	X	X	X					
1. OPTIMIZATION	X	X	X	X	X									
a. ISOLATED	X													
b. ARTERIAL		X	X	X	X									
c. NETWORK				X	X									
2. SIMULATION	X	X	X	X	X	X	X	X	X					
a. ISOLATED	X		X		X	X								
b. ARTERIAL		X	X	X	X	X	X							
c. NETWORK			X	X	X	X	X							
d. FREEWAY							X							
e. URBAN FREEWAY								X						
f. RURAL FREEWAY									X					
B. PROVIDED FUNCTION	X	X	X	X	X	X	X	X	X			X		
C. DATA MANAGEMENT SYSTEM	X		X							X				
D. HIGHWAY CAPACITY ANALYSIS											X			
E. SIGNAL WARRANT ANALYSIS													X	
F. ORIGIN-DESTINATION PLANNING														X
II. COMPUTER REQUIREMENTS														
A. MAINFRAME	X	X	X	X	X	X		X	X					X
B. MICROCOMPUTER	X		X	X	X	X				X	X	X	X	

Set Up Evaluation Goals

The basic evaluation goal for this prototype expert system, as described earlier, is to provide specific recommendations about software packages to meet user requirements. The INSIGHT 2+ PRL program is constructed to meet this objective:

1. Program belongs to signal-timing optimization programs
   1.1 Program SOAP is recommended
   1.2 Program MAXBAND is recommended
   1.3 Program AAP is recommended
   1.4 Program TRANSYT-7F is recommended
   1.5 Program SIGOP-III is recommended
2. Program belongs to traffic flow simulation models
   2.1 Program NETSIM is recommended
   2.2 Program TRAFLO is recommended
   2.3 Program FRESIM is recommended
   2.4 Program ROADSIM is recommended
3. Program belongs to other traffic engineering software
   3.1 Program ITDS is recommended
   3.2 Program HIGHWAY CAPACITY is recommended
   3.3 Program PPD is recommended
   3.4 Program COUNTS-PC is recommended
   3.5 Program LINKOD is recommended

The INSIGHT 2+ program uses a set of outline-type evaluation goals with different degrees of recommended action coded as part of the prototype expert system. In this particular system setup, several things are noted. First, the definition of this goal is identical to the functional classification given in the section on traffic engineering software. Second, the purpose of each program package is identified as part of the goal definition. This arrangement indirectly implies the inclusion of the program categories as part of the decision rule. The other possible programming approach is to not classify these programs under three main goals but treat each program as a separate goal.

Select Evaluation Constraints

Evaluation criteria are made hard to identify by differences in traffic engineering evaluation constraints. The evaluation criteria used for this analysis were that each transportation-related computer analysis software package must be unique, identifiable, and classifiable.

1. Unique: The specified constraints should be adequate to describe the characteristics of the computer software to be analyzed.

2. Identifiable: The selection criteria should provide clearly defined characteristics for decision-making support, such as definable application areas and confident answers.

3. Classifiable: The common features selected in the analysis are the potential program applications, such as optimization, simulation, and other transportation-related features. The major application areas of these FHWA computer software packages can be implemented for isolated intersections, arterial streets, generalized signalized networks, rural highways, urban highways, and freeway corridor systems.

Develop the INSIGHT 2+ Production Rule Language

To build expert systems with INSIGHT 2+, the user must first specify a set of goals for decision making. INSIGHT 2+ uses the production rule language (PRL) to represent knowledge in terms of IF–AND–THEN–ELSE rules that contain factual information in the expert knowledge domain. PRL also allows the end user to specify procedural rules and execute dependent conditions to search for any unsatisfied IF conditions. Knowledge bases have a variable threshold, or a minimum confidence acceptability level, that can be adjusted as the knowledge data base is executed. A numeric confidence level may also be assigned to each conclusion to allow the user to work with specialized knowledge. This applies only with the known simple-facts and question-answer type of query in the evaluation. Knowledge bases created by INSIGHT 2+ can be executed quickly at a microcomputer-based work station.

Basically, the key words of the INSIGHT 2+ PRL are command words for programming the main decision-making and other information-supporting functions. They are

AND	DISPLAY	IF	RULE
ARE	ELSE	IS	THEN
OF	END	OFF	THRESHOLD
CONFIDENCE	EXPAND	ON	TITLE

In this study, the necessary constraints are selected for the basic facts, rules, and application areas for each of the computer packages in the analysis. First, the particular application groups are assembled and grouped as basic constraints. Second, the explicit information for designing the detailed expert system structure is defined with the commands TITLE, THRESHOLD, CONFIDENCE, and GOALS. Third, the production rules are set up according to the nature of the conclusions and recommendations for traffic engineering management. Last, the trace report provided in the INSIGHT 2+ knowledge engineering system makes possible the study of the decision-making process according to the specific production rule defined. An example follows to describe the information selection process in the MAXBAND program using the basic information obtained from the FHWA software awareness report. It gives the rules coded for the MAXBAND program according to the required constraints extracted from Figure 1. As indicated, a set of natural language program statements first defines the prerequisite conditions for determining the function of the MAXBAND program in this prototype expert system.

```
RULE for selecting program MAXBAND
IF Program belongs to Signal Timing Optimization Programs
AND Optimize signal timing      Cycle Length
OR Optimize signal timing      Phase Length
OR Optimize signal timing      Offset
AND Optimized Facility is Arterial Street
AND Program can Not simulate Cycle Length
AND Program should Provide plots of Time-Space
      Diagram ?
AND Computer requirement is      Mainframe
AND Type of microcomputer you use is      None
THEN Program MAXBAND is recommended
AND DISPLAY MAXBAND
```

The INSIGHT 2+ system separates the decision rules into two basic categories: knowledge rules and inference rules. Knowledge rules include the facts about and relationships of a problem that are embedded in the expert's knowledge. For example, if the experience of the traffic engineer suggests that optimization of traffic signal timing is required for traffic analysis, then this element becomes the major deciding factor for choosing a particular signal-timing program. The inference mechanisms, on the other hand, can tell the computer how to use knowledge rules to solve a problem. Inference is a reasoning algorithm, not a rule, that provides the reasoning or problem-solving strategy. In a completed AI program, knowledge rules are usually combined with both the knowledge base and the inference rules in the expert system to provide better application.

In operation, the INSIGHT 2+ inference engine or knowledge processor of the expert system compares the decision rules in the knowledge base with the facts and information entered by users. If the user-input information is incomplete, the inference engine will ask the user to provide more descriptions for additional analysis. It can also offer conclusions and explain recommended actions in a natural language interface. Usually, the recommendations are based on the reasoning used to reach final conclusions. Moreover, it can provide the user question-and-answer prompts in English not just output computer codes. The reasoning or inferencing process will link related decisions supplied by the user to appropriate actions from the production rules in the knowledge base. These linked rules form knowledge chains in which the THEN statement may become the IF statement that can eventually lead to the most likely conclusion in the evaluation.

Debug the Production Rule Language

Program testing and debugging are essential to successful computer programming and ES applications. The program support environment makes it relatively easy to compile and debug the PRL. The basic procedure for running this particular expert system is described in the following steps. Load the program in response to the MS DOS prompt command. Load the INSIGHT 2+ interpreter by typing 'I2'. Next, specify the knowledge base, FHWAINF0 in this case, to start program execution. After compiling production rules, INSIGHT 2+ will flag error messages until the compiled knowledge base can be obtained. Then, run the program by using the function keys that are defined at the bottom of the display screen. After loading the compiled program in the interpreter mode, press the function key F3 to start the expert system for analysis. The user will then respond to the questions presented and select the desired answers. At the end, recommendations to meet the user's input requirements will be given.

After the trace report has been reviewed, the production rules can be revised using the trial-and-error method. The different levels of trace reports could be used to evaluate the relative effectiveness of the model. Programming efforts should be continued until the computer program works as designed in the production rules. Some working experience obtained from debugging the INSIGHT 2+ knowledge engineering tools includes revising the search sequence for each study goal. It was noted that the order of the goals and subgoals may be used to rank their relative importance and how each goal is analyzed in the system. Because the system separates each goal according to its unique identification number and characters, these items are important in the initial design of the PRL to avoid any potential problems. Logical errors coded in PRL may create a lot of design problems later on. Therefore selection of the proper facts and constraints is important for successful operation of an expert system.

COMPLETE PROGRAM DOCUMENTATION

To complete the necessary program documentation, knowledge engineering tools usually provide various programming-support functions and commands. Program documentation about the system should be done through the use of three command words in the INSIGHT 2+ system—TITLE, EXPAND, and DISPLAY.

1. TITLE: This function is used to summarize the contents of the expert system under design and provide additional information for program documentation and reference. For example, this function was used in this prototype expert system to explain the proper execution steps for instructing users how to execute and implement this expert system before the actual execution of the INSIGHT program.

EXECUTION STEPS

I. USER INPUT INFORMATION.
- PROGRAM CATALOG
- OPTIMIZATION CAPABILITY
- CYCLE LENGTH SIMULATION
- OPTIMIZE FACILITY
- SIMULATE FACILITY
- INPUT DATA PROGRAM
- TIME-SPACE DIAGRAM
- MAINFRAME OR MICROCOMPUTER
- TYPE OF MICROCOMPUTER

II. PRODUCTION RULE ANALYSIS.

III. RECOMMEND SOLUTION ALTERNATIVE.

IV. OPTIONAL TRACE REPORT.

2. EXPAND additional information: The EXPAND function can be used to describe the questions in the production rules and the constraints in the query analysis. This command and other similar key words can also provide explanations and the characteristics of the problem under analysis. This function was used in this system to describe the question to be asked the user.

EXPAND Optimized Facility is isolated intersection

Diagram of an isolated intersection:

```
            ¶
   _________¶_________
            ¶
            ¶
```

3. DISPLAY supportive information: This functional command is used to expand the basic and relevant information for

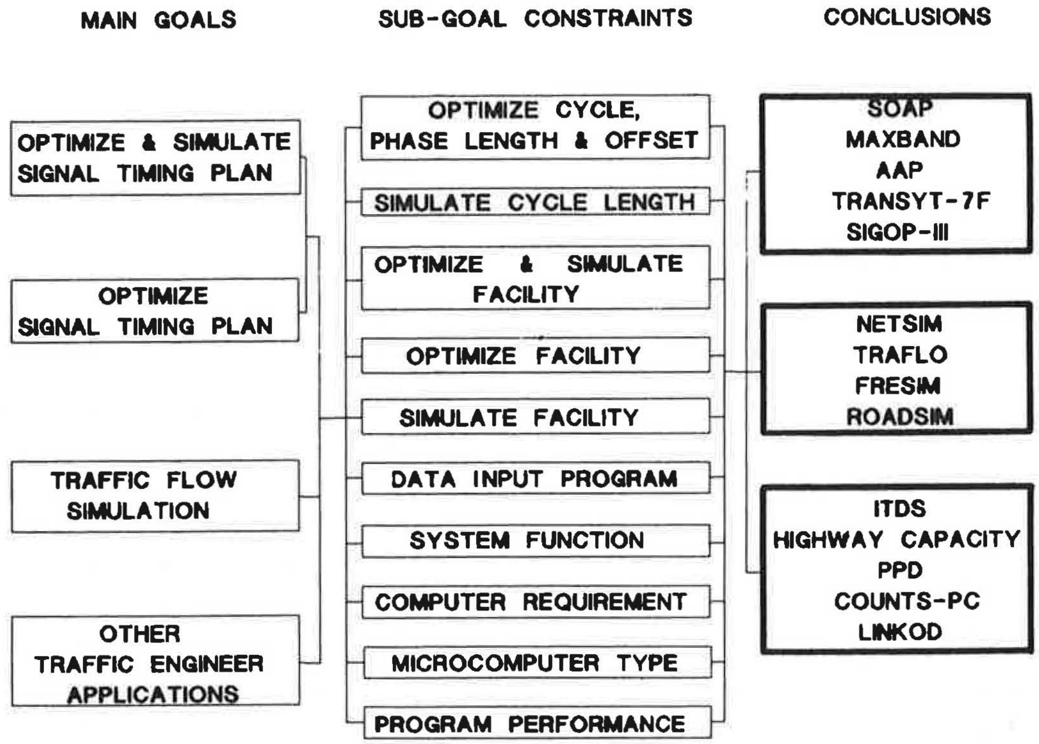

FIGURE 6 Basic program structure.

providing supportive suggestions about the recommended design alternatives and the question itself during the query process. The DISPLAY function was used in this expert system to provide additional suggestions to obtain directions for searching more information when conclusions have been reached through the knowledge inference process.

DISPLAY MAXBAND (MAXimum BANDwidth)
MAXBAND — MAXimum BANDwidth
LATEST AVAILABLE VERSION: A time-space diagram was recently added to the original (and, to date, only) version of the program. The program is also available in FORTRAN-77.

PROGRAM STRUCTURE

The structure of the expert system, as described in this paper, is shown in Figure 6. The functional structure of the program clarifies the interrelationships among the six components: *(a)* main goal, *(b)* subgoal, *(c)* selection criteria, *(d)* constraints, *(e)* rules, and *(f)* recommendations. The prototype expert system program was developed to explain the main goals, relate main goals and subgoals using production rules, explain subgoal constraints, demonstrate selection criteria for the production rules, and recommend subgoal conclusions to main goals in the AI/ES analysis.

The INSIGHT 2+ knowledge system consists of an inference system and a knowledge base compiler. The inference mechanism executes the knowledge data base. After the user has selected a knowledge base on a particular topic, INSIGHT 2+ searches for all possible recommendations. INSIGHT 2+ presents the user with questions to answer and goals to select. The knowledge system formulates the goal choices, the questions, and the conclusions from information obtained from the domain expert and the end user. The knowledge base compiler works with the knowledge engineer's input to create the compiled knowledge base that INSIGHT 2+ runs. The knowledge engineer creates a knowledge base using PRL and a standard text editor processor. INSIGHT 2+ takes the knowledge base, translates it, and then streamlines it so the INSIGHT 2+ knowledge system can run faster in execution. The INSIGHT 2+ tools can be best used in areas that require routine professional judgment. They can assist engineers and managers in designing procedures for implementation. They are helpful when many people at different locations need expert advice to do a job.

USING INSIGHT 2+

The advantages of using a knowledge engineering tool such as INSIGHT 2+ are its easily understood programming structure and well-equipped support functions in a user-friendly environment. Other advantages of using a knowledge engineering tool such as the INSIGHT 2+ system follow. Sequencing in the production rule is important only for the definition of goals and subgoals. Although the order of the constraints in the production rules is not important in query input, the interpreter will seek to optimize the execution sequence and the operation of the expert system. The order of evaluation constraints within the production rule for defining goals and subgoals will not influence execution of the expert system.

The knowledge engineering tool is also easy to use. The information that needs to be defined is the specific constraints required to determine each individual goal and subgoal using decision table analysis. The coding of PRL is efficient within the knowledge engineering programming environment. Both forward- and backward-reasoning processes can be performed in this expert system without additional computer-programming efforts. This PRL program can be maintained easily using the built-in editing function or regular word-processing facilities.

With knowledge engineering tools like INSIGHT 2+, a person can develop expert systems, which accumulate knowledge on a subject or knowledge base, to analyze, reason, and provide solutions to problems that would normally require human expertise. INSIGHT 2+ uses a backward- and forward-chaining inference mechanism. In a forward-chaining application, INSIGHT 2+ can be used to acquire user input and try to recommend a software package according to the information it contains on a particular application or a pattern described by the knowledge rules. In a backward-chaining application, INSIGHT 2+ begins with a specific software package and determines whether or not the preconditions justify using that package.

INSIGHT 2+ does have some disadvantages. Because of the interconnected cause-and-effect relationships, errors in program logic are difficult to identify. The INSIGHT 2+ system also limits the type of data that can be analyzed in the PRL system. Moreover, there are limitations on the length of a line to be coded in the PRL program. But perhaps one of the most important improvements between the INSIGHT 1 and INSIGHT 2+ systems is the addition of explicit OR functions for eliminating duplicate definitions of each individual condition. This will provide the benefit of not having to specify every possible decision tree by using duplicate production rules.

CONCLUSIONS AND RECOMMENDATIONS

This study evaluates the feasibility of using expert system designs to aid in the selection of programs. Texas Transportation Institute researchers believe that it is cost-effective to develop computer software using AI techniques to assist the end user in optimizing traffic management strategies. The expert systems design can assist practicing traffic engineers in selecting proper traffic-related software for developing better traffic control strategies in both urban and rural areas. Furthermore, the production rules of the proposed expert systems design, developed with either AI languages or knowledge engineering tools, can provide an alternative means for representing traffic engineering expertise in the decision-making process.

AI languages and tools are generally more flexible for developing expert systems yet more difficult for programming than is a conventional computer language. Because of the complexity involved in AI/ES programming, only well-trained programmers can comfortably use the LISP and PROLOG languages to build expert systems. Knowledge engineering design can be done with a range of knowledge engineering tools for developing specialized applications. Knowledge engineers often have to make decisions about the programming languages to be used. If portability is the primary concern, they will probably choose to translate their codes into conventional programming languages that can later be run on conventional operational systems. On the other hand, if more complex or sophisticated expert systems are to be developed for future applications, the tools may be coded in LISP or PROLOG and designed to run on LISP- or PROLOG-based machines. Usually, AI languages do not have user-friendly programming support for ES development as do knowledge engineering tools, which can be easily used on conventional computer systems.

To enhance this prototype expert system, it is recommended that the information be stored outside the expert system to optimize program execution and compilation of the knowledge data base. To restructure this expert system, additional investigations are also needed in the following three areas *(12–14)*:

1. From a knowledge engineering programming standpoint:
   - Modify the goals, subgoals, rules, and constraints;
   - Add a debug error message in the trace report;
   - Include a logic table or logic tree in the trace report; and
   - Enhance the program through the INSIGHT 2+ system.
2. From a domain expert applications standpoint:
   - Modify design using manual procedures,
   - Work with other knowledge engineering systems,
   - Obtain experience from teaching end users, and
   - Use other computer programs.
3. From an end user applications standpoint:
   - Develop the expert system to interface with other software, such as DBASE, PASCAL, and LOTUS programs;
   - Expand the knowledge bases to help the end users; and
   - Provide determinable information, such as threshold settings.

It is further recommended that expansions be enhanced to provide a computerized expert system for advising end users in the selection of proper computer programs for effective traffic engineering analysis. This application will be especially useful in the future for helping users select suitable computer software packages in the TRAF family as supported by the FHWA. It is also believed that modification of this expert system could be best achieved by improving AI/ES program efficiency and restructuring the formulation of existing programs for object-oriented problem-solving applications.

ACKNOWLEDGMENTS

The author appreciates the research support of the Texas Department of Highways and Public Transportation in cooperation with the U.S. Department of Transportation, Federal Highway Administration, and the Texas Transportation Institute. In addition, the constructive comments and suggestions provided by the TRB reviewers were helpful in the completion of this paper.

REFERENCES

1. *Use of Traffic Simulation Models in Project Planning and Evaluating.* Office of Safety and Traffic Operations Research and Development, Traffic Systems Division, FHWA, U.S. Department of Transportation, Sept. 1985.
2. A. S. Byrne et al. *Handbook of Computer Models for Traffic Operations Analysis—Technical Appendix: Summary of Models and References.* Report TS-82-214. FHWA, U.S. Department of Transportation, Dec. 1982.

3. A. S. Byrne et al. *Handbook of Computer Models for Traffic Operations Analysis*. Report TS-82-2134. FHWA, U.S. Department of Transportation, 1982.
4. *Special Report 194: The Application of Traffic Simulation Models*. TRB, National Research Council, Washington, D.C., 1981.
5. Y. S. Hsu and P. K. Munnjal. Freeway Digital Simulation Models. In *Transportation Research Record 509*, TRB, National Research Council, Washington, D.C., 1974, pp. 24–41.
6. W. D. Labrum and R. M. Farr. *Traffic Network Analysis with NETSIM in a Small Urban Grid*. Report TS-80-230. FHWA, U.S. Department of Transportation, June 1980.
7. E. B. Lieberman and R. D. Worrall. *Traffic Network Analysis with NETSIM—A User's Guide*. Report IP-80-3. FHWA, U.S. Department of Transportation, Jan. 1980.
8. D. A. Wicks and B. J. Andrews. *Development and Testing of INTRAS, A Microscopic Freeway Simulation Model*. 4 Vols. Reports RD-80-106, RD-80-107, RD-80-108, and RD-80-109. FHWA, U.S. Department of Transportation, Oct. 1980.
9. E. Lieberman. *Microscopic Simulation for Urban Traffic Management*. 5 Vols. Reports RD-80-113, RD-80-114, RD-80-115, RD-80-116, and RD-80-117. FHWA, U.S. Department of Transportation, Jan. 1982.
10. *An Introduction to Urban Travel Demand Forecasting—A Self-Instructional Text*. User-Oriented Materials for UTPS. FHWA and UMTA, U.S. Department of Transportation, 1977.
11. P. H. Winston. *Artificial Intelligence*, 2nd ed. Addison-Wesley, Inc., Reading, Mass., July 1984.
12. P. Harmon and D. King. *Expert Systems—Artificial Intelligence in Business*. John Wiley and Sons, Inc., New York, 1985.
13. D. A. Waterman. *A Guide to Expert Systems*. Addison-Wesley, Inc., Reading, Mass., 1986.
14. B. G. Buchanan and E. H. Shortlife. *Rule-Based Expert Systems—The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Inc., Reading, Mass., 1985.
15. *PD PROLOG—User's Manual*. PROLOG documentation for the educational and public domain system. Robert Morein and Automata Design Associates, Dresher, Pa., 1986.
16. *TURBO PROLOG—The Natural Language of Artificial Intelligence—Owner's Manual*. Borland International, Inc., Scotts Valley, Calif., May 1986.
17. *INSIGHT 1 Reference Manual*. Level Five Research, Inc., Indialantic, Fla., 1986.
18. *INSIGHT 2+ Reference Manual*. Level Five Research, Inc., Indialantic, Fla., 1986.
19. C. T. Hendrickson, D. R. Rehak, and S. J. Fenves. Expert Systems in Transportation Systems Engineering. *Transportation Research*, Jan. 1986.
20. C. Zozaya-Gorostiza and C. T. Hendrickson. An Expert System for Traffic Signal Setting Assistance. Civil Engineering Department, Carnegie-Mellon University. Submitted to *Journal of Transportation Engineering*, ASCE, Jan. 1986.
21. W. F. Clocksin and C. S. Mellish. *Programming in PROLOG*, 2nd ed. Springer-Verlag, New York, 1984.
