**APPLICATION OF EXPERT SYSTEMS**

**IN LANDSCAPE ARCHITECTURE**

by

NITIN Y. KULKARNI

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master in Landscape Architecture

in

College of Architecture and Urban studies

APPROVED:

Professor Benjamin C. Johnson, Chairman

Dr. Terry R. Rakes

Dr. Jane T. Nutter

July 20, 1989

Blacksburg, Virginia

**APPLICATION OF EXPERT SYSTEMS**

**IN LANDSCAPE ARCHITECTURE**

by

NITIN Y. KULKARNI

Professor Benjamin C. Johnson, Chairman

College of Architecture and Urban studies

(ABSTRACT)

Application of artificial intelligence (AI) has been a topic of interest among researchers for the past decade or more. Years of research in the commercial application of AI, availability of hardware support for AI application and affordability of software and hardware has generated a lot of interest in this field and brought this technology within the reach of micro-computer based users. The commercial impact of AI is due to expert systems (ESs). ES technology is a collection of methods and techniques for constructing human-machine systems with specialized problem solving expertise.

This project explores the application of ESs in landscape architecture by developing a prototype ES and testing implications of its use with designers while working on a hypothetical problem in a studio environment. The development process helps identify the typical difficulties of such an application, to uncover technical problems, and to identify areas needing further research.

The project aims at building an ES that provides very limited preliminary data and design guidelines to initialize the design process and keeps track of the most fundamental issues necessary for planning, thus acting as an expert and assistant simultaneously. The idea is to explore the possibility of applying ESs to facilitate the design process so that designers may concentrate on other important aspects of design which include intuitive judgement about qualitative aspects.

The prototype ES is developed for designing a parking lot for a commercial complex. A rule base is developed, which infers information to respond to the user input data and provides the user with preliminary guidelines to initiate the design process. The rules and heuristics are elicited from on-campus experts of the College of Architecture and Urban Studies at Virginia Polytechnic Institute and State University.

# Acknowledgements

The success of a project is not the result of the activities or work of one person alone. It is a teamwork of various thoughts and contribution of different people. This thesis is no exception. I would like to extend my warm appreciation to everyone who has helped me make the study possible.

I express my gratitude to Prof. Benjamin Johnson for his constant encouragement throughout the project and his valuable contribution. Special thanks goes to Dr. Terry Nutter and Dr. Terry Rakes for their valuable input in developing this project.

I am thankful to Dr. Will Shepherd and Dr. Roger Ehrich for making the necessary hardware required for this project available to me.

I am also grateful to my 'knowledge experts', Prof. Benjamin C. Johnson, Dr. Dean Bork, Prof. Larz Anderson and Dr. Will Shepherd, for their valuable time and contribution.

Foremost, I would like to thank Dr. Loren Rees, Dr. Terry Rakes and Dr. Terry Nutter for their valuable guidance during system design and programming stages.

# Table of Contents

# List of Illustrations

# 1.0 INTRODUCTION

## 1.1 GENERAL STATEMENT

With the advent of technology, the field of Landscape Architecture like every other field is becoming more complex. Increased complexity in the requirements (in terms of rules, standards, and codes), issues (like environmental and social issues), and efforts to establish stronger basis for design rationale are the three basic components of this complexity. So much information is generated with research and discovery of new materials that it is becoming difficult for the designer to maintain rapport with all of this information along with the efforts necessary to develop inovative designs. If a landscape architect insists on keeping in touch with all this information, he is left with very little time to involve himself in the intuitive aspects of the design process. Environments must function as completed wholes and because of this their designers tend, when faced with a choice, to base their decisions on knowledge that is broad rather than deep. The depth of knowledge the landscape architect lacks of course, is compensated for in comprehensiveness. The designer on one hand has the impulse for necessary comprehensiveness at a minimal depth of the knowledge; on the other hand the researcher has the drive for the depth at the cost of breadth.

This problem contributes to the increasing distance between designers and researchers. Moreover, it seems that there may be "two cultures" that dominate the incubation of design knowledge.[1] As these perceptions are getting the wide spread consensus,[1] it will be important to organize what designers know about the design process in ways that can be useful to researchers and vice versa.

With the computer technology available until a few years ago, it was not possible to manage complex problems using predicate logic knowledge representation which is important for developing natural language user interfaces. Processors for the computers are available with ever incerasing speeds, storage capabilities are increased considerably. Their capabilities are now also breaking the bounds of linear processing. Is it possible to take advantage of this technology in solving complex design problems ?

Recent research in Artificial intelligence (AI) has made the process of information processing feasible at a much higher complexity, and this may be used as an aid in developing the link between these "two cultures" and also as an aid in both design and research processes. This project explores this feasibility of using computer applications of Artificial Intelligence in developing a more effective two-way thinking of research with design at different possible levels used in the design process by developing a prototype for information processing based on the expert's in depth knowledge to help the designer in the design process.

## 1.2 BACKGROUND

### *1.20 Artificial Intelligence*

In a broader sence, artificial intelligence (AI) is a branch of Computer Science concerned with making computers act more like human beings. During the early days in the ninteen sixties, AI researchers attempted to model the human thinking process by developing general problem solvers and general-purpose programs for solving broad classes of problem. This effort had a limited success even though it produced some interesting research results. In the seventies, AI researchers focussed their attention on representation and search techniques, i.e. how to represent knowledge and to search for a solution in order to solve the problem most efficiently. It was around the mid seventies that AI researchers started to realize that the problem solving ability of human lies basically in their knowledge of a particular domain and not so much in th inferential mechanism they use. [11]

AI can be defined as *"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 pursuing the possibilities that a computer can be made to behave in ways that human reconize as 'intelligent' behavior in each other"* [12]

### *1.21 Developments in Artificial Intelligence*

In seeking to bring quantitative methods into wider use, the scientific community has only recently come to appreciate the heuristic and tacit knowledge that routinely guides professional judgements. This recent appreciation was founded on two new perceptions of decision making.

This is the realization that most of the profound and interesting problems of the physical and social world are mathematically intractable. This mathematical intractability makes it exceedingly difficult to develop computational algorithms that reliably represent logical problem solving procedures.[2] As a result, the computer has remained at arm's length from the process of technical decision-making, barely supplementing and never supplanting human agency.

The second perception came to cognitive scientists and computer specialists attempting to simulate human thinking processes in computing machines. After consulting with introspective professionals in a variety of applied fields and analyzing formal thought protocols generated in experimental settings, researchers in AI and the cognitive process now remark of the professional expert, "He knows more than he is aware of knowing".[3] Practicing experts -- whether engineers, landscape architects, chemist or any others -- could produce a meaning from a jumble of cues, but found it difficult to articulate into forms that seem intelligible to researchers their own methods of heuristic problem formulation, their computational procedures or even their overall schemes for reasoning.

In these types of case studies, AI researchers were able to infer the implicit structures of these schemes by careful, after-the-fact examination of the input and output of hundreds of professional judgments in realistic problem settings combined with questioning the domain experts closely over and over again to verify their inferences. This process of deriving these implicit structures gave rise to the specialized branch in AI called 'Knowledge Engineering'. In the context of this discussion, AI may be defined as "The computer based solution of complex problems through the application of processes that are analogous to the human reasoning process."[4]

**Direct approach in problem solving and AI approach**

A direct approach can be very loosely described as "Implementation of a program that executes in a prepackaged form, the results of a developer's reasoning regarding a specific problem". In simple language many people have described the policies and procedures for highly bureaucratic organizations as plans designed by geniuses to be executed by idiots. Taken in the extreme, a direct computer implementation is analogous to this type of human plan.

A direct implementation is efficient in terms of both execution speed and required storage space. This advantage results from the fact that the program has been streamlined to address a localized problem. In contrast, an AI implementation is relatively flexible and so it is less fragile. Because the system is applying flexible reasoning dynamically, it is often able to perform some portions of the task when faced with partially incomplete or inaccurate data and it can often adapt to modified problem statements. Problems with verifying requirements and partially incomplete or inaccurate data are relatively easy for a human, but they are relatively difficult for a computer using direct implementation methods. In contrast to the direct approach, the AI approach is based on the following principle. 'AI techniques explicitly attempt to move the reasoning process into the program'.

AI techniques, because of the power of their internal reasoning process, can be used to solve very complex problems; specifically, the problems that, because of their complexity, are the most difficult to solve through more direct methods.

## 1.3 PURPOSE

The purpose of this research is to explore the feasibility of using AI techniques in the design process. The design process involves an uneven flow of information inputs that are filtered and packaged by the designer's knowledge. This flow is then output again as information which initiates another series of inputs to a builder or a citizens' group, etc. The information ranges from visceral impressions to highly structured automated data bank. A designer's knowledge originates in information that has been pared, shaped, interpreted, selected and transformed.[5]

If designers could replicate or emulate their common knowledge, if they could understand and capture it sufficiently to equip a machine for knowledgeable information processing, then computers could do much more than the present variety of tasks. They could actively participate in the generative design process. With this futuristic vision in mind, this research address the following issues.

### *1.31 Issues*

- How it is possible to organize, manage and process the information with an expert's skill, so that it does not preempt the creativity of a landscape architect ?
- How can a landscape architect organize and use the ever expanding repertoire of relevant information ?
- How could we relieve the landscape architect from tracking and verifying quantitative aspects of design, to pursue qualitative aspects ?

If AI technology can solve these problems, then pursuing the exploration of its application in landscape architecture is worth the effort.

### *1.32 Expert systems*

Experts, with their uncanny abilities, intuitive judgement and years of experience, can quickly arrive at a precise solution to a given problem. This process requires data transformation and processing and complicated quantitative analysis with the help of heuristics that the expert has developed over the years. Through the application of AI techniques, Expert systems (ESs) capture the basic knowledge that allows humans to act as experts when dealing with complicated problems.

An ES can be defined as "A computer program application that solves complicated problems that would otherwise require extensive human expertise."[4] They provide high level intellectual support for human experts and non-experts by representing and applying knowledge electronically.[6] Frederick Hayes-Roth, a leader in the field of commercial application of ESs, describes their functioning :

> *"Knowledge systems interact with humans through the same modalities as other computer programs, but they do different things and use different methods. They carry on meaningful dialogues with users, explain their own reasoning and generally work to understand a user's problem and then help solve it."* [7]

Knowledge systems translate their problem solving goals into natural language, transform their current lines of reasoning into structures the user can understand, and map the user's description of a problem into a familiar form that they can solve. These capabilities make knowledge systems appear as relevant, intelligent and helpful. And to the extent a knowledge system gives insightful and creditable explanations of it's behavior, it will also appear trustworthy to the user. Although AI encompasses works in the fields such as natural language processing, vision systems, speech recognition and synthesis, due to the present constraints of the technology that limit exploration of most of these aspects (such as vision and speech processing), the commercial impact of AI is mainly due to Expert systems.[8]

## 1.4 GOALS

The aim of this project is to explore the application of Expert Systems (ESs) in landscape architecture by developing a prototype ES and testing implications of its use with designers while working on a hypothetical problem in a studio environment. The development process helps to identify the typical difficulties of such an application, to uncover technical problems, and to identify areas needing further research.

The project aims at building an ES that provides very limited preliminary data and design guidelines to initialize the design process and keeps track of the most fundamental issues necessary for planning, thus acting as an expert and assistant simultaneously. The idea is to explore the possibility of applying ESs to facilitate the design process so that designers may concentrate on other important aspects of design which include intuitive judgement about qualitative aspects.

### *1.41 Specific goals of the project*

1. To develop a prototype expert system on a small scale which provides background information and simple design guidelines necessary for sketch planning of a parking lot for a commercial building.
2. To document the development process.
3. To identify the problems observed in development of such a system.
4. To test application of its use with designers while working on a hypothetical problem in a specific domain for which the expert system is designed. By doing so, to demonstrate the potential application of the comprehensive ESs in the design process.
5. To refine either the aims or the system, based on the outcome of this pilot project.

## 1.5 SCOPE

The scope of this project is limited to a small scale prototype Expert System (ES). Figure 1 shows the phases of the design process and possible application of ES to it. Figure 7 shows the components of the work. The area identified for the prototype is preliminary guidelines used in the design of parking lots for commercial buildings, in particular shopping complexes (See Figure 7). For a detailed description of the problem identification process, please refer to chapter 3.44: problem identification.

Designing a parking lot for a commercial building should involve the following issues :

- Site topography
- Soil characteristics
- Carpet area of the building
- Storm water management
- Building code and safety requirements
- Climatic aspects
- Special features of the site
- Paving materials
- Landscape design criteria
- Circulation, access points
- Illumination standards
- Site furniture selection
- Designer's heuristics about any or all of these aspects

A series of 'If-Then' rules are made for these different aspects. Knowledge elicited from experts is stored in the form of frames and rules.

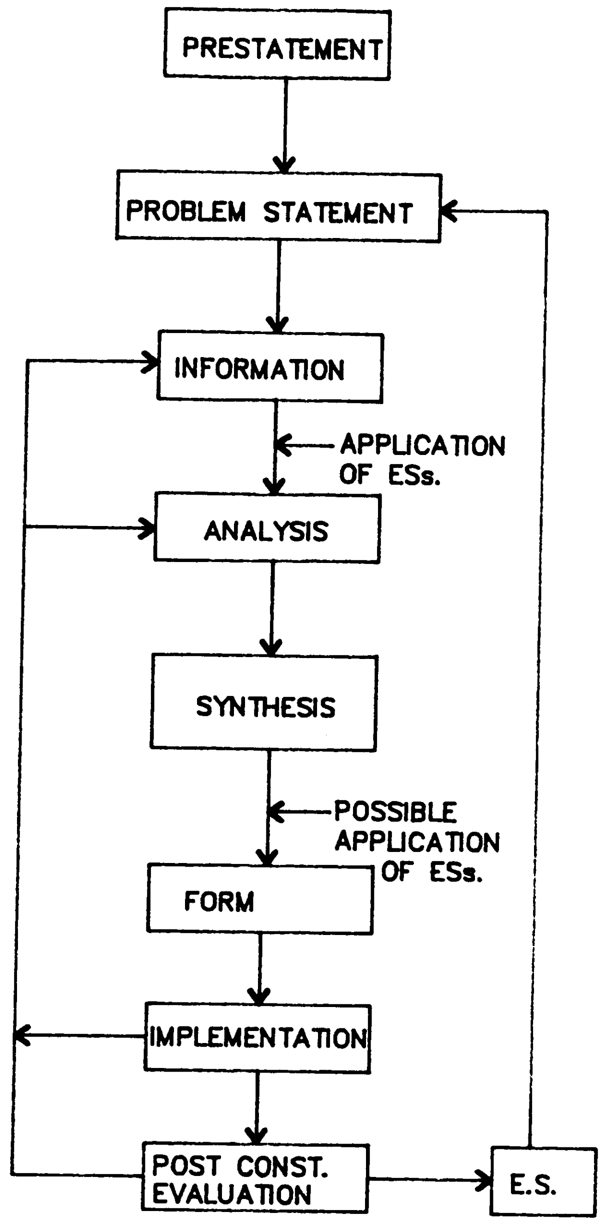



**Figure 1. APPLICATION OF ESs IN DESIGN PROCESS**

**EXAMPLES.**

```
IF
  BUILDING CARPET AREA IS X SFT.
THEN
  Y NUMBER OF CARS NEED TO BE PARKED


IF
  Y NUMBER OF CARS NEED TO BE PARKED
AND
  SLOPE OF THE SITE IS Z %
THEN
  M CFT. OF GRADING CHANGES WILL BE NEEDED


IF
  SLOPE OF THE SITE IS Z %
AND
  SOIL HAS ABSORPTIVE CAPACITY K
THEN
  ?F TYPE OF STORM WATER DRAINAGE IS ADVISABLE


IF
  SLOPE OF THE SITE IS Z%
AND
  SOIL TYPE IS ?H
AND
  PRECIPITATION RATE IS ?Y
THEN
  ?Q TYPE OF GROUND COVER IS ADVISED
```

These rules can be designed to extract information from databases including climatic data or information on materials for paving. These rules interact with the knowledge represented in frame structures. The system asks the user a series of questions about different aspects of site interactively, and analyzes information in the framework of the rules and knowledge represented in frames and databases. It then provides the user with recommendations for the various aspects to be considered in relation to that information. (See Figure 2)

The geographical location is restricted to and encompasses Blacksburg, Virginia and its immediate vicinity to maintain the size of the prototype in the manageable form.

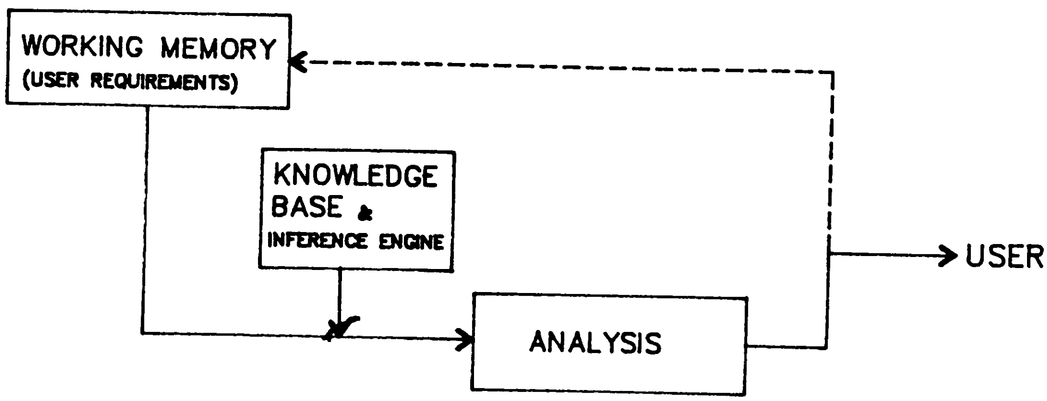



**Figure 2. BASIC COMPONENTS OF AN EXPERT SYSTEM**

## 1.6 LIMITATIONS

Due to the complex nature of the design process and the time available to develop this application software, the prototype addresses only a small portion of the whole process. The work addressed is limited to a segment of in the site planning process in detail rather than attempting to address the comprehensive process with less depth. The emphasis on depth was considered necessary because the anticipated advantages of an ES for designers lay precisely in keeping track of large numbers of detailed design criteria. These advantaged could not be realized without limiting to a narrow subject and investigating it deeply.

The scope was limited in depth and breadth in order to focus on the system architecture. It was also understood that due to the limited number of issues in the problem domain incorporated in the knowledge base, prototyping must demonstrate the potential for the overall system while only focussing on a specific domain. The process of designing such a prototype and testing it with designers in a studio environment is itself a step forward in the direction of the system design. In the long run, to test this kind of system conclusively with designers, the system has to be much more comprehensive in nature. As this effort is first of its kind in this problem domain, the intention is to develop a foundation for this process which further research can build upon.

## 1.7 METHODS AND INSTRUMENTATION

The project has produced a knowledge base (K-base) within the framework of rules and an explanation facility. It consists of fundamental principles of site planning used in the preliminary design phase of site development for a commercial building. A rule base has been developed which will access this knowledge to respond to user input data and provide the user with preliminary design guidelines to initiate the design process. The rules and heuristics have been elicited from on campus experts of the College of Architecture and Urban Studies at Virginia Polytechnic and State University, Blacksburg , VA 24060.

An expert system shell, Goldworks, designed and developed by Gold Hills Computers, Cambridge, MA 02139, is used for developing the ES. The software uses the Golden Common LISP programming language, a modified version of Common LISP, running on an IBM AT-compatible with an 80387 co processor, 20Mhz of processing power and above, and a minimum 6Mbyte of RAM. The ES was tested with four groups of designers. All groups were provided with specific data about the site. (Refer to Appendix B for the problem statement and figure 8 for list of criteria used in design of prototype ES). Two groups used the ES in analyzing this data and the others did not. The designs prepared by all groups were evaluated using criteria similar to those in the rule base and the issues covered in the recommendations. (i.e. the solutions were searched for the methods used in analyzing the site specific data provided along with the problem statement.) The results are stated at the end of this document.

## 1.8 GLOSSARY

***Artificial Intelligence*** - "A sub-field 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 a 'intelligent' behavior in each other".[5]
The foundations of AI are divided into representation, problem-solving methods, architecture and knowledge.

***Blackboard*** - The blackboard in a Blackboard system is a global memory structure that is used to contain the emerging partial solutions from different modules of the system. (See Figure 5)

***Blackboard Model, The*** - The Blackboard model was developed as an abstraction of the techniques used in the Hearsay [9] and Hasp [10] systems, and is a system architecture that is used to structure reasoning in complex domains.

***Common LISP*** - The standard dialect of LISP that is used in commercial AI.

***Domain*** - A subject matter area or problem solving task.

***Expert System*** - "An ES is a computer application that solves complicated problems that would otherwise require extensive human expertise".[4]
To do so, it simulates the human reasoning process by applying specific knowledge and inferences. Internally, an ideal ES can be characterized as including the following:

1. Extensive specific knowledge from the domain of interest
2. Application of search techniques

3. Support for heuristic analysis
4. Capacity to infer new knowledge from existing knowledge
5. Symbolic processing
6. An ability to explain its own reasoning

***Expertise*** - The skill and knowledge that some humans posses that result in performance that is above the norm. Expertise often consists of massive amounts of factual information coupled with rules-of-thumb, simplifications, rare facts and wise procedures all compiled in a way that allows the expert to analyze the specific types of problems in an efficient manner.

***Explanation*** - In ESs explanation normally 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 telling the user something about its assumptions or its inner reasoning.

***Frame*** - A knowledge representation scheme that associates an object with a collection of features. (e.g., facts, defaults, 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.

***Heuristic*** - A rule-of-thumb or other device or simplification that allows its user to draw conclusions without being certain. Unlike algorithms, heuristic do not guarantee correct solutions.

***Heuristic Rules*** - Rules written to capture the heuristics an expert uses to solve a problem.

***Inference*** - The process by which new facts are derived from established facts.

***Inference Engine*** - That portion of an expert system that contains the inference and control strategies.

***Knowledge*** - "An integrated collection of facts and relationships which, when exercised, produces competent performance." (Harmon, 1988).

***Knowledge Acquisition*** - The process of locating, collecting and refining knowledge. This may require interviews with experts, research in the 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 is derived from current sources, especially from experts.

***Knowledge Base*** - The portion of an expert system that consists of the facts and heuristic about a domain. The knowledge may be in the form of examples, facts, rules or objects.

***LISP*** - A programming language based on List Processing. Currently one of the most popular languages for AI, was developed by John McCarthy in 1958.

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

***Rule (If-Then Rule)*** - 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.

***Symbolic Inference*** - The process by which lines of reasoning are formed, for example, syllogisms and other common ways of reasoning step by step form premises. Some inference procedures can use degrees of uncertainty in their inference making.

## 1.9 REFERENCES


1 Villecco, M., ***Architectural Research Round Table.*** Aug 1975.

2 ***An Introduction to knowledge based Expert Systems.*** The Australian Computer Journal, Vol. 12, No. 2, May 1980, pp. 56.

3 Feigenbaum, et al., ***Themes and case studies in Donald Michie.*** Expert systems in micro-electronic age. Edinburg University Press, 1979.

4 Rolston, W. D., ***Principles of Artificial Intelligence and Expert Systems development.*** McGraw Hill Inc. 1988, pp. 1-2.

5 Feigenbaum, et al., ***The fifth generation, Artificial Intelligence and Japan's computer challenge to the world.*** Reading, MA. Addison- Wasley, 1983.

6 Partich, C., ***Expert Systems- Forcasting powerful support for the designer.*** Landscape Architecture, Vol. 76, May/June 1986, pp. 70-74.

7 Roth, F. H., ***The knowledge based expert systems- A tutorial.*** Computer, Sept. 1984.

8 Davis, D. B., ***Artificial Intelligence goes to work.*** High Technology, April 1987, pp. 16.

9 Erman, L., ***"The HEARSAY-II Speech-Understanding System : Integrating Knowledge to Resolve Uncertainities,"*** Computing Surveys, vol. 2, June 1980, pp. 213-253.

10 Nii, P. et al., ***Signal to signal transformation: HASP / SIAP case study*** AI Magazine, Spring 1982, pp. 23-35.

11 Feigenbaum, E. A. and McCorduck, P., ***The Fifth Generation*** Addison-Wesley, Reading, MA, 1983, pp. 258.

12 Feigenbaum, E. A. and McCorduck, P., ***The Fifth Generation*** Addison-Wesley, Reading, MA, 1983, pp. 4-5.

# 2.0 REVIEW OF EXPERT SYSTEMS' LITERATURE

## 2.1 INTRODUCTION TO EXPERT SYSTEMS

Expert systems (ESs) are used to perform a variety of extremely complicated tasks that in the past could be performed by only a limited number of highly trained human experts. Through the application of AI techniques, ESs capture the basic knowledge that allows a human to act as an expert when dealing with complicated problems. Perhaps the most intriguing and powerful characteristic of ESs, and the one which distinguishes them from more traditional computer applications, is their capability to deal with challenging real-world problems through the application of processes that reflect human judgement and intuition.

***Definition of an ES.*** An ES is a computer application that solves complicated problems that would otherwise require extensive human expertise. To do so, it simulates the human reasoning process by applying specific knowledge and inferences.[1] The heuristics used in the development of such ESs are usually accumulated by a human expert over a number of years. Using heuristics, an ES can make educated guesses, recognize promising approaches and avoid blind search and consequently it can narrow down the search process in a solution space.

Various other interpretations and definitions of ESs can be found in the AI literature. To list a few examples,

> An intelligent computer program that uses knowledge and inference procedure to solve problems that are difficult enough to require significant human expertise for their solution. [14]

> An interactive computer program incorporating judgement, experience, rules of thumb, intuition and other expertise to provide knowledgeable advise about a variety of tasks. [15]

> An ES solves real-world, complex problems using a complex model of expert human reasoning, reaching the same conclusions that the human expert would reach if faced with a comparable problem. [16]

Internally an ideal ES has following components- [2] (See Figure 3)

***Knowledge base*** - This is the repository of information available in a particular domain. The Knowledge Base (KB) may include well established and documented definitions, facts and rules, as well as judgmental information, rules of thumb and heuristics.

***Inference mechanism*** - (Also known as inference engine or reasoning mechanism). This controls the reasoning strategy of the ES by making assertions, and drawing conclusions from the hypotheses. In rule-based systems, the inference mechanism determines the order in which rules should be invoked and resolves any conflict among the rules when several rules are satisfied.

***Working memory*** - (Also known as context or global database). This is a temporary storage area for information relating to the current state of problem being solved. Its content changes dynamically and includes both information provided by the user about the problem and information derived by the system.

***Explanation facility*** - This provides answers to questions (e.g. why a certain question is being asked) and justifies answers (e.g. why a specific conclusion or recommendation is made).

***Knowledge acquisition*** - This facilitates the structuring and developing of the knowledge base.

**UTILITIES**

***Debugging facility*** - This helps and guides the developer to debug parts of the program during programming sessions.

***Help facility*** - This helps and guides the user to use the system effectively and easily.

***Intelligent interfaces*** - The user interface allows the user to interact with the ES and query the ES. It may include natural language processors, menus, multiple windows, icons or graphics.

***Knowledge based editors*** - This helps and guides both the user and the developers while updating rules or the knowledge base in a system.

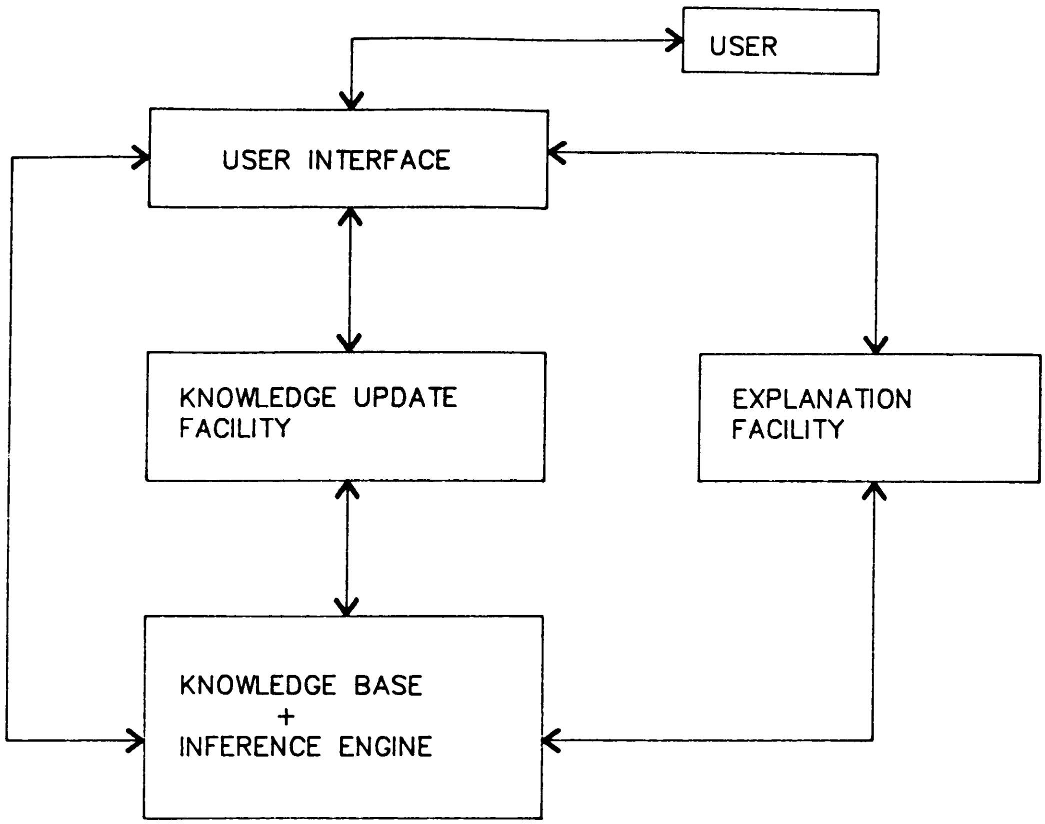



**Figure 3. ARCHITECTURE OF ADVANCED EXPERT SYSTEM**

## *2.11 Early expert systems in the market place*

A few examples of successful early ESs are

1. MYCIN - developed at Stanford university in mid 70's, makes judgement on the diagnosis of bacterial infection in patients and proposes courses of therapy with antibiotics. [17]

2. DENDRAL - developed in early and mid 70's at Stanford university. Used for identifying the molecular structure of organic compounds from mass spectral and nuclear magnetic response data. [18]

3. PROSPECTOR - developed at Stanford research institute in late 1970's used for diagnostics in mineral explorations. It imitates the reasoning process of an experienced exploration geologists for finding an ore deposit in a particular region. It discovered a molybdenum deposit whose ultimate value will probably exceed $100 million. [19]

4. CADUCEUS - It is used to diagnose knowledge of pulmonary function disease for diagnostic consultation and now provides expert analysis at a California Medical Center. [20]

Expert systems are being used in disparate fields from business to science, and their application is limited only by imagination.[3]

## 2.12 *Role of knowledge in Expert Systems*

An expert's power derives from extensive domain specific knowledge rather than from an understanding of generic expert behavior. Three major components of the knowledge that is the source of an expert's ability to perform are listed as facts, procedural rules and heuristic rules.

***Facts*** are statements that express truths within the subject domain. ***Procedural rules*** are well-defined invariant rules that describe fundamental sequences of events and relations among facts relative to the domain. ***Heuristic rules*** are general rules in the form of hunches or rules of thumb that suggest procedures to be followed when invariant procedural rules are not available. The presence of heuristics contributes greatly to the power and flexibility of ESs and tends to distinguish ESs from the traditional software.[1]

## 2.13 *Programming Languages for Expert Systems*

In general, ES programming focuses on issues of inference and heuristic search and depends heavily on the manipulation of symbols -- strings of characters that are freely used to represent any possible element in the problem domain.

The programming languages LISP and PROLOG are by far the most common languages used in ES development. More conventional languages like C are also coming into use. Symbolic processing is important in ESs because the knowledge primitives in a knowledge base and relationships between the knowledge primitives are stored using symbolic representations.

## *2.14 Expert Systems development process*

For a problem domain, a knowledge based system is appropriate if,

1. The designers have access to a definitive knowledge about the problem (often in the form of a bonafide expert);
2. the designers (or the expert) know how to solve the problem and have done so in the past, and further can explain how to solve the problem;
3. the domain of the problem is known and its bounds are clearly defined;
4. the designers can express the data, the decision making process and the decisions themselves in a straightforward manner; and
5. the application, once designed and tested can be used repeatedly, spreading out the costs (time, money and so on) of developing it.

The ES development process consists of basic stages that are similar to standard software engineering life cycle segments. These stages consist of problem identification, prototype construction, formalization, implementation, evaluation and long term evolution. It is a reiterative process (see Figure 4). These stages are discussed with reference to this project in chapter 3.0.

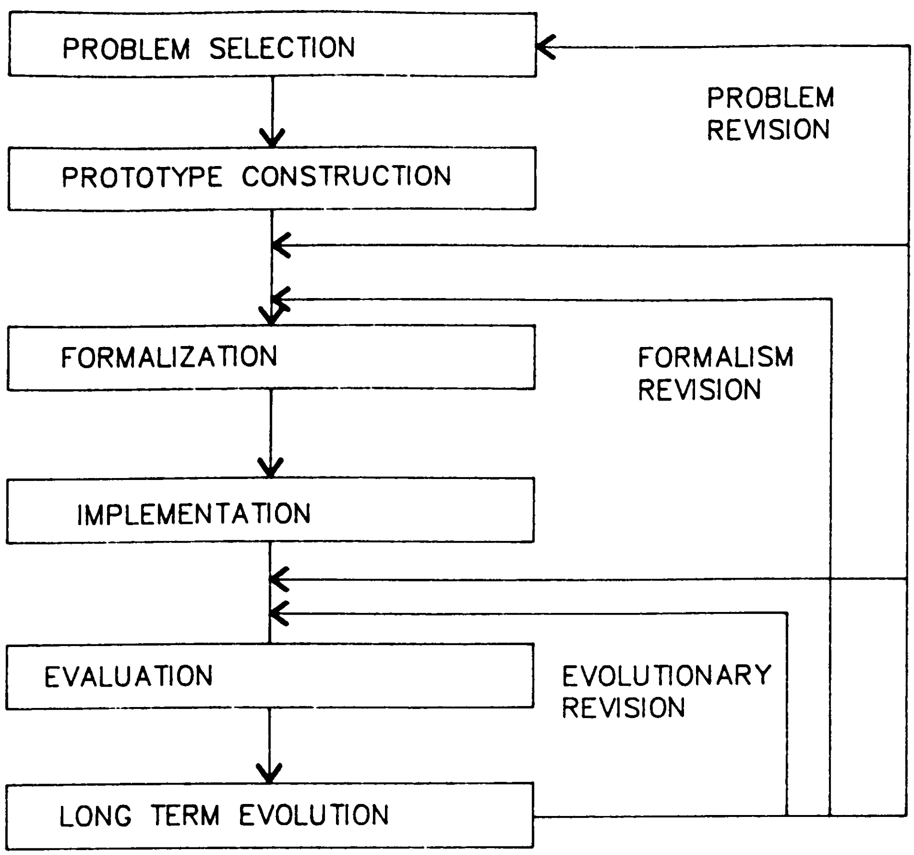



**Figure 4. EXPERT SYSTEM DEVELOPMENT PROCESS**

## 2.2 APPLICATION OF EXPERT SYSTEMS IN ARCHITECTURE

Architecture is one field where such systems are not as simple for application due to the extreme complexity of the nature of the issues involved in design decision processes. This is evident from the limited literature available in the field, and understandably so, since artificial intelligence (AI) is only now beginning to reveal its versatility in application. Very few articles listed in the bibliography relate directly to the proposed work of this project.

The literature review has shown the documentation of the difficulties of applying ESs due to the complexities in design process, [4,5] and the areas within architecture in which there is a need for development of ESs. [6,7,8] Very few, such as Jung/Brannen Associates, an architectural firm from Boston have undertaken the development of ESs for architectural application.[9] Fisher reports that the R&D branch of this firm identified some 300 possible ES applications of which 27 'modules' have been developed and tested for such building elements as fire stairs, elevator cores, toilet rooms and HVAC systems. Extensive work in exploring the area of application of ESs for details and specification has been done by Lansdown. [10]

There are other ESs in architecture ranging from construction applications, structural engineering, construction-specification and detailing, etc. (Refer to appendix B for ESs in civil engineering). The designer would prefer to stay away from almost all these areas and the knowledge is quantifiable. Fisher in his article reports that the potentiality of ESs lies in the non- aesthetic aspects of architectural practice.* The recognized experts in, say, acoustics or management, have knowledge capable of being explicitly described. Also in these areas, most architects have the least expertise.

---

*Although new trends in research in AI show that it is not absolutely true, with the available technology, our first step will be to explore these non-aesthetic aspects before stepping into visual aspects of representation in design.

Among the researchers in architectural field, there is a general agreement that building design is a knowledge-based activity.[8, 11, 12] It is composed of both qualitative and quantitative aspects of knowledge. In a design process, the more information that is available for review and consideration, the better is the solution. This is a widely accepted belief and universal to problem solving processes.[7] According to Forbes, Architect and head of the R&D Corp., set-up by Jung/Brannen Associates, "Architects must become information brokers controlling where and how information is used". This allows the architect to stay involved with the building long after it has been completed. The first article in the issue of AJ on detailing and specification (AJ 1.6.83 pp. 59) showed that many failures or performance shortfalls in buildings arise from known principles and rules not applied to the design and construction problems at hand. At the first glance this seems purely like a management problem. In fact this is related to the issue of accessibility of knowledge to the designer during the design process which is vital in today's world of ever-modifying technology. Expert systems demonstrate a great potential to carry over this task.

Success of any design and architectural firm therefore lies in the way information is organized and how it is processed. Petrich, [8] Carrol, [7] and others go a step further by suggesting an ESs -CAD/CAM integrated system which would act intelligently to manipulate and process the images, specific to the domain in relation to information and data, and vice versa. Though it is possible theoretically and technically to produce one, it will be a long time before a complex system of this nature is developed, and this would require developers from within the profession of architecture, preferably designers. These designers should be trained to transform the designers logic and heuristics used while manipulating this information. To test the feasibility of such a system, a component of the integrated large scale system is developed on a smaller scale in this project and is tested with designers for its utility.

Almost no literature was found siting any applications of AI techniques in the field of landscape architecture except Petrich's article. [8] This shows the void in our profession which needs to be filled in with research.

## 2.3 REFERENCES


[1] Rolston, W. D.,***Principles of Artificial Intelligence and expert systems development***, McGraw-Hill, Inc 1988, p 1-2.

[2] Adeli, H. ***Expert systems in construction and structural engineering.*** Chapman and Hall, 1988.

[3] Ehurkin, N J., ***ES practical face of AI***, Technology review, Nov/Dec 1983, Vol. No 6., pp 73-78.

[4] Cathain, O C., ***Expert systems and Design***, Sept. 1987, pp58-61.

[5] Schon, D. A., ***The reflective practitioner: How professionals think in action***, New York: Basic books, 1983.

[6] Gero, S. J., et. al., ***Knowledge based building design***, Advanced building technology, V 1/11, CIB. 86.

[7] Carrol, C. R., ***Computer-aided Creativity -Have we found it?*** The construction specifier, March 1986, pp 21+

[8] Petrich, C., ***Expert Systems: Forecasting powerful support for the designer***, Landscape Architecture, May/June, Vol. 76, No. 3, 1986, pp 70- .

[9] Fisher, T., ***Techniques: Expert Systems***, Progressive architecture, June 1986, pp 106-107.

[10] Lansdown, J., ***Expert systems: 1 knowledge for designers & Expert systems 2 a prototype for designers*** Architects journal, Feb 1984, pp 75+.

[11] Gero, J S., ***Editorial***, Architectural Science review, Sept 1987.

[12] Kinoglu,F., ***Expert systems:model of the design process***, Design news, March 1986, pp 149.

[13] Davis, D. B.,***Artificial Intelligence goes to work***, High technology,April 1987, p16.

[14] Feigenbaum, E. ***Expert Systems in the 80's*** Machine Intelligence (ed. A. Bond), 1981.

[15] Gasching J. et al., ***Development of a Knowledge-Based System for Water Resources Problems*** SRI Project 1619< Stanford Research Inst., 1981.

[16] Weiss, S. M. et al., ***A practical duide to designing expert systems*** Rowman and Allanheld Publishers, Totowa, N. J., 1984.

[17] Shortliffe, E. ***Computer-based medical consultation*** Elsevier, N.Y., 1976.

[18] Lindsay, R. K. et al., ***Applications of artificial intelligence for Organic Chemistry*** The DENDRAL project, McGraw Hill, N. Y., 1980.

[19] Duda, R. O. et al., ***Development of the PROSPECTOR system for mineral exploration*** Final report, Stanford Research Institute, Menlo Park, Calif., 1979.

[20] People, H. E., Jr., ***Heuristic methods for imposing structure on ill-structured problems : the structure of medical diagnosis*** Artificial Intelligince in Medicine, Westview Press, Boulder, Colorado, 1982.

# 3.0 LARGE SYSTEM ARCHITECTURE

## 3.1 INTRODUCTION

The architecture of large Expert System (ES) which is mentioned as a comprehensive ES for the design process can be conceived as a distributed blackboard based model. The blackboard in a blackboard system is used to contain the emerging partial solutions derived from different modules of the sytem. The blackboard manager, which is the central link in this model connects all the logically independent knowledge sources (See Figure 5). The comprehensive *Architectural Design ES* will consist of many of such logically independent knowledge sources (See Figure 6).

A few examples of these logically independent modules are the soil classification module, site furniture selection module, stormwater drainage management module, product information database, plant selection module, and building bylaws filing system. Many of these modules may not have to be ESs by themselves. Some are available in the market as established softwares with their own interfaces. To incorporate these modules in the comprehensive ES, their user interfaces may need to be modified. Such an integrated system will also include a tutorial manager, which will be linked by n number of tutorial expert modules. All these modules will be related to each other through the blackboard.

To follow the functioning of such an *Architectural design ES*; if the user wants to prescribe the use of a product for a particular use, after his input for the query about availability of such a product, the database module will first check the availability of such a product and post it on the blackboard. The file system module will retrieve the available products and pin up the components and characteristics of these products on the blackboard. The building code model now will retrieve this information to check the compatibility of these characteristics with the building code and regulations. The criteria selected by the user for the economic aspects will be posted on the blackboard from the user interface at this time and those along with the suitable product passed by the building code module will be retrieved by the economic analysis module and select the product which will fit the economic criteria and pass it to the user interface. This kind of system will also be able to handle the conflicting data by either prompting user to check the data or inferencing from the conflicting data by priority resolution.

With this kind of facilities, such a system will be able to handle almost all of the computational, database related and repetitive functions involved in the design process and relieve the designer from the burden of information management to be able to concentrate on the subjective and qualitative aspects of design. A small component of one such module of this system is developed in this project as a prototype ES. This module *'Site criteria analysis assistant'* is discussed late in this document.

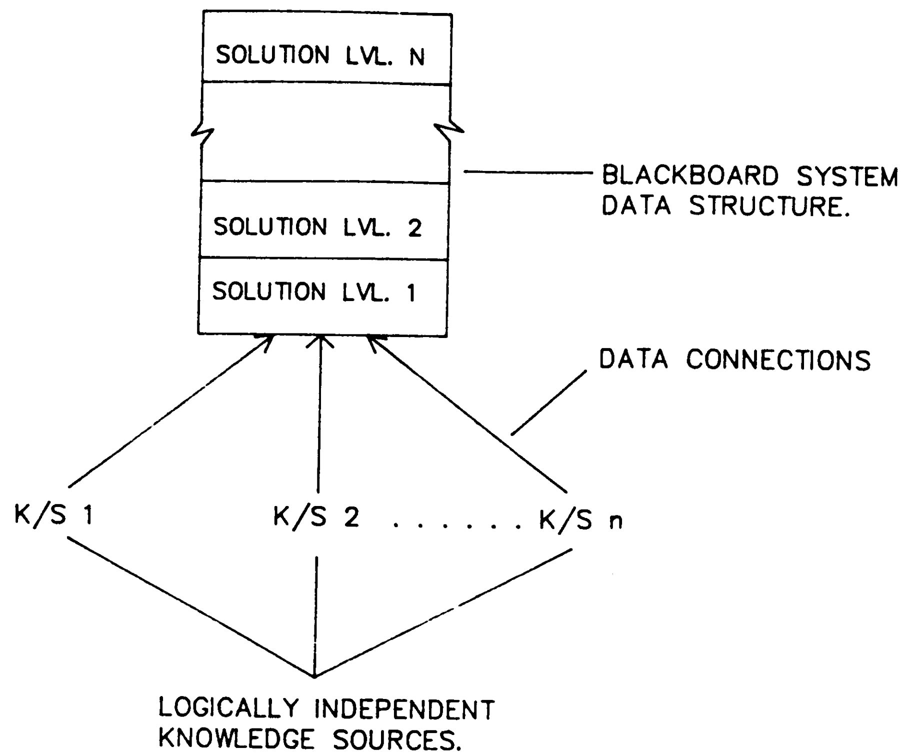



**Figure 5. BLACKBOARD MODEL**

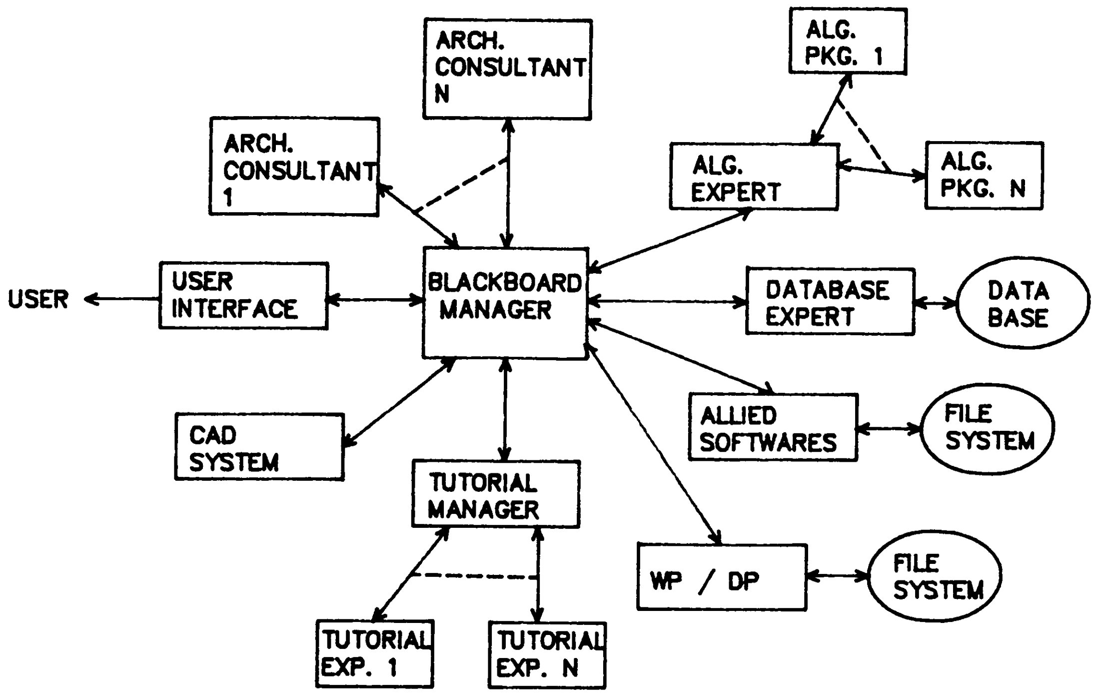



**Figure 6. DISTRIBUTED BLACKBOARD-BASED MODEL FOR ARCH. DESIGN ES**

## 3.2 KNOWLEDGE ENGINEERING AND PROTOTYPE DESIGN

Knowledge engineering (KE) is the process of codifying or formulating the knowledge used for making decisions. Specifically, we must transform the knowledge held by humans into a knowledge base whose operation, treated as a "black box", parallels that of the humans. That is, given the same input, we want to get substantially the same results. Ultimately, this knowledge base together with the other components of a knowledge based system are used to help solve problems. The process of KE is interlinked with the system structure as the knowledge representation in a knowledge base is directly related to the system architecture (e.g. lattice, rules, frames, etc.)

A few important factors to be considered before beginning the KE are to look for the source of knowledge, availability of experts, and possibility of the conflict resolution in expert opinions about the problem domain. Also, it is important that the kind of problem the system is to address has been solved successfully in the past. In different domains, KE could be a very difficult art. In real life problems, it may be necessary to elicit knowledge from an expert who is handling the problem domain during his work hours and so may not be available to discuss the issues exclusively. Hence the knowledge engineer may have to develop the logic behind his heuristics by observation. The prototype cycle for the design is presented in Figure 4.

## 3.3 KNOWLEDGE ACQUISITION

Having a knowledge engineer (KE) who is also a designer considerably reduced the amount of time other experts had to spend with the knowledge engineer by eliminating the need for the KE to familiarize himself with different concepts in the design profession. This project drew experts from the faculty of the School of Landscape Architecture and the School of Planning. The KE had an average three to four meetings with every expert and a few meetings with town council authorities. The system also included knowledge drawn from secondary sources including books, articles, and standards. Generally, the first meeting with each expert ended by finding more appropriate reference materials for this project.

A sort of reluctance was found among the experts in expressing the heuristics in generalized terms in the problem domain. Two experts pointed out that this reluctance is due to the extensive number of factors involved in the site analysis process, many of which vary from site to site. Experts in the design profession tended not to quantify (e.g. experts would comment on high or low penetration rates, but when asked for numbers, they referred to the standard books). There was a clear reluctance in providing information on design process heuristics although it was agreed that site analysis heuristics were the integral part of the design process.

These difficulties led to modification of the objective of design of a prototype expert system to instead address a smaller part of the problem domain (See Figure 1). In particular, post construction analysis was not implemented in the prototype. The revised prototype ES addresses the objective site analysis issues which related more to hard knowledge rather than soft knowledge like objective design decisions. A few contradictions were found in experts' opinions, but almost all of them were resolved except a few subjective opinions like the maximum distance to be walked by a person from his car to the door of the shopping complex. Standards were followed to resolve such contradictions.

## 3.4 PROTOTYPE FORMULATION

### *3.41 Types of systems and problems*

Rule based systems can be thought of as a system forming a continuum with a simple forward chaining checklist system at the left end and an integrated ES at the right. A system's functionality varies along this continuum, with the more functional (powerful) ones being found closer to the right. However, a useful system can be built anywhere along this continuum. This hierarchy can be listed as follows

1. Forward or Backward chaining system.
2. Hybrid Forward and Backward chaining system.
3. Hybrid Frame based system.
4. Hybrid Frame based system with integration to external software.
5. Integrated ES.

The system developed in this project can be identified in the third category. The architecture of the system is discussed later in this chapter.

### *3.42 Areas of application*

Expert systems are developed for many application areas, including information filtering, real time monitoring, consultation, maintenance, diagnosis, configuration, decision support and analysis, classification, and simulation and modelling. The system developed in this project can be classified under the category of consultation. Consultation systems are also known as advisory systems. These are on of the most common types of knowledge based systems. They give 'expert advise' or specific consultative information on one problem in response to questions or requests. Such systems often incorporate a diagnostic ability.

### *3.43 Prototype design*

The prototype is designed as a forward chaining hybrid rule and frame based system. The original objective was to handle two aspects of the design process. One was to work as an aid in the pre-analysis stage and the second was to work as an aid in the design evaluation stage. Due to time constraints, the limited nature of the prototype and difficulties in eliciting knowledge in the problem domain for evaluation, efforts were cincentrated on the pre-analysis stage. (See Figure 1).

### *3.44 Problem identification*

Selecting the problem domain to develop a prototype was a two-way process. Figure 7 shows the division of design criteria in its smallest components in the context of this project. This division tree can be expanded further to very minute details, but for simplification, depth is limited to the extent shown in the figure. The idea was to identify the role of this problem in the bigger picture. The specific problem domain was chosen in such a way that it is a modular part of the whole system, but also incorporates many of the design decision criteria in its definition.

Out of performance standards, site components which is a sub part of planning factors was selected. Out of the five identified problem domains in site components, parking spaces was selected as a target problem domain due to its relative simplicity as compared to other problem domains. The issues involved in design decisions of parking spaces are broad in range.

These selection criteria also indicate that an integrated ES could incorporate all the remaining problem domains and that they could be added in the form of modules to the parent ES shell in later phases as desired and as practical. Hence, modularization is a key factor in adopting these selection criteria. Due to the complex nature of the design process, all the modules identified in this process are interconnected at some point or other; but this aspect should not pose problems in developing the integrated ES due to the versatility and flexibility of available ES shells in the market.

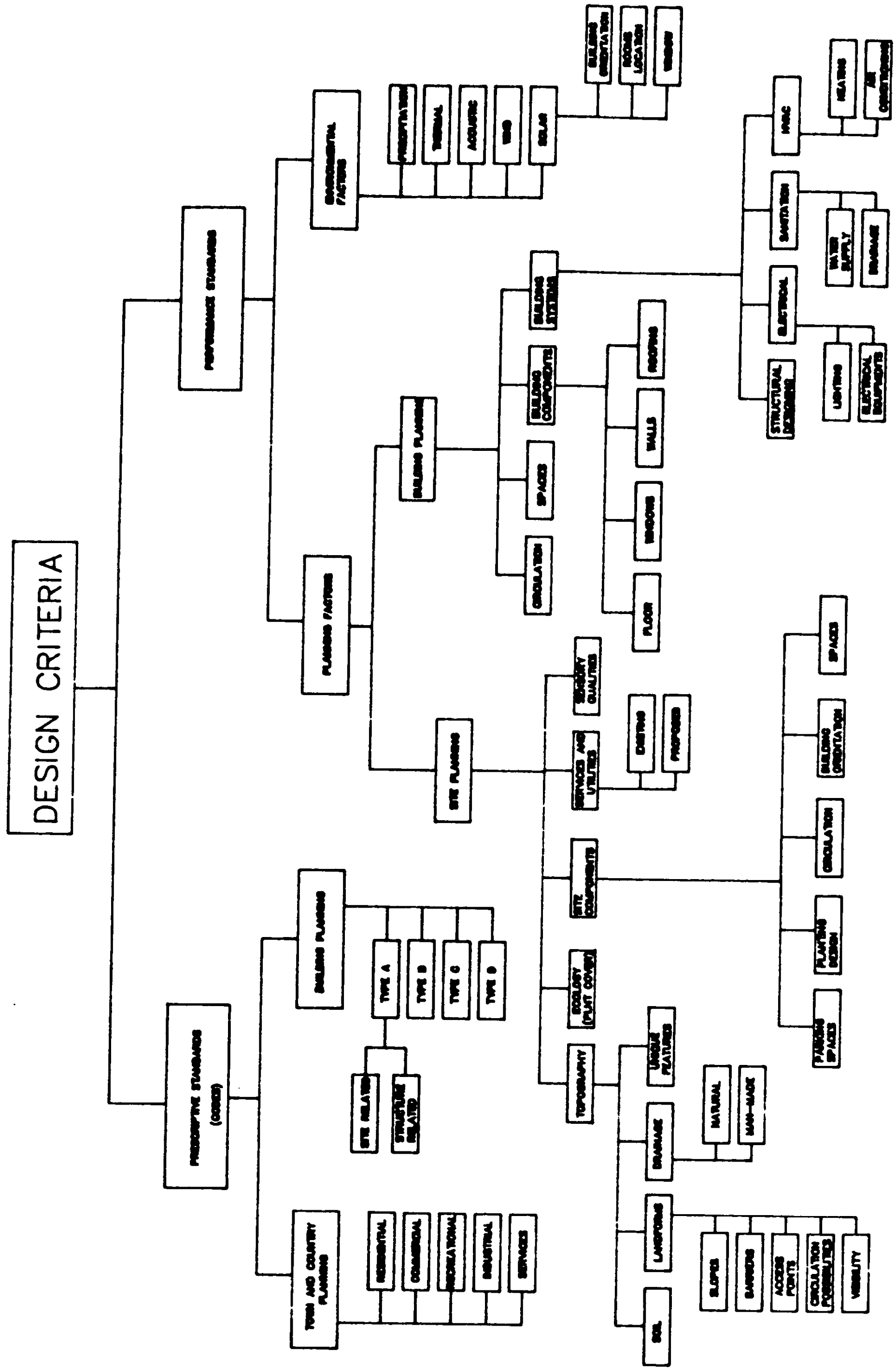



**Figure 7. CLASSIFICATION OF DESIGN CRITERIA**

### *3.45 Architecture of prototype*

The prototype is a hybrid frame and rule based forward chaining system. The different components of the prototype are discussed in the following paragraphs.

***1. USER INTERFACE*** - When initiated, the system displays a short introductory screen about its functioning and purpose. Next, the user is asked to press return to start the session. When started, series of questions are asked about the site in a simple menu-driven manner. As an additional facility in the program during the questionnaire, the user is prompted if he is interested in getting any help in different standards like spatial standards, illumination standards, etc.

All the information collected through the session is displayed in an abbreviated form at the end of the session on a work screen. The user can make any needed changes on this screen before starting the inference mechanism. At the end of this phase, the user is asked to confirm that all the information input is correct.

When this is done, the inference engine receives all the information and forms and displays on the screen its recommendations based on the site-specific data. The user can also get a hard copy of these recommendations. To receive a formatted copy of these recommendations, different LISP routines are designed in the program. If the user choses to see information on standards on the data collection phase, that information also becomes part of the final output. The intent is to demonstrate the utility of such a system to act as a information access facility in addition to acting as an advisor. To control the sequence of the output, rules in the inference engine are designated with priority rather than controlling their sequential firing with sponsors.* As the size of the system increases, using sponsors may be a better solution due to their modular nature.

***2. FORMATION OF RULES*** - The following is an example of how a particular rule was derived from the heuristic knowledge of one of the experts.

Heuristics :

> *"To develop a site with an effective economic standpoint, 1% to 3% slope is best. Flatter land may pose drainage problems and a site over 8% slope may need extensive cut and fill making it economically not viable unless there are strong economic forces. If the slope is more than 3%, some cut and fill may be necessary to develop the site for designing a parking lot"* (Dr. William Shepherd, Knowledge Expert).

Because the ES requires that knowledge in the knowledge base be stored in accordance with the system's knowledge representation convention, the knowledge engineer must transform the representation of the knowledge. In this heuristic knowledge, four types of landforms are suggested to be treated differently. They are, slope less than 1%, 1% to 3%, 3% to 8%, and more than 8%. As this knowledge needs to be transformed in cause-effect format due to system requirements, each type will be *"If the type of slope is X, Then Y type of action is suggested."* The four rules developed from this paragraph in simple English can be stated as-

If slope is very flat (less than 1%).
Then reconsider the site for its impact for development cost.

If slope of the site is in the range of 1% to 3%.

---

* Sponsor is a special structure in Goldworks to control th resources allocated to the firing of rules. Sponsors are organized in a tree hierarchy and control the sequential firing of related rules in that hierarchy.

Then this site can be developed without intention of cut and fill and may be the best possible site economically for development cost.

If slope of the site is in the range of 3% to 8%.

Then this part of the site may need to be developed with terracing.

If slope of the site is more than 8%.

Then reconsider this part of the site for its impact for development cost.

These rules are named in the program as 'slope-very-flat, slope-flat-1, slope-flat-2 and slope-rolling'. Following is the example of the complete rule 'slope-very-flat.

**(define-rule SLOPE-VERY-FLAT**

**:print-name "slope-very-flat"**

**:certainty 1.0**

In case of this rule, certainty factor is not used in inferencing and hence default is 1.0

**:doc-string "if the slope is less than 1%, then reconsider the site for its economic impact for development cost."**

This type of explanation string is useful when the user needs to make changes in the program through the developer's interface. They act as the explanatory notes in the program.

**:dependency nil**

Dependency factor is not used in the inferencing mechanism.

**:direction :forward**

This rule will fire in the forward direction during inferencing. i.e. when all the conditions in the IF part are satisfied, the Then part will get executed.

**:priority 0**

Priorities are used to control sequential firing.

**:sponsor TOP-SPONSOR**

Sponsors are also not used to define the hierarchy in the inferencing process.

**(start-inference-engine)**

This assertion is made by firing the rule *intermediate assertion* to activate the inference engine.

**(instance input-data is input-values**

**with slope-val very-flat)**

This is the second condition to be satisfied for this rule to fire. i.e. if the user selects the value *very-flat* for the slope of the site.

**THEN**

**(evaluate**

**(recon-line :write-line**

**"Reconsider this part of the site for its economic impact for development cost as the slope is less than 1% here")))**

Then this recommendation is passed on to the inference engine.

***3. KNOWLEDGE BASE*** - The knowledge is stored in the form of rules and frames. The entire user interface is controlled by the set of rules along with a few predefined frames of Goldworks used for display purpose. The inference engine consists of frames which are used in the beginning of the session for primary storage of the information (global memory) provided by the user for a specific site. This information is then retrieved and processed by the rules during the inference process. The copy of the program in appendix A reflects an effort to minimize complexity by reducing the number of assertions made during the inference process. On the next few pages is an example of how the system derives one design recommendation from the user provided data.

The following list gives the knowledge in plain English before transforming into the knowledge base into the system.

***RULES IN PLAIN ENGLISH***

1) IF
    Slope is too flat (< 1%)
  THEN
    Reconsider the site for its economic impact for development cost.

2) IF
    Slope is flat (1%-3%)
  THEN
    Site could be developed without intention of cut & fill.

3) IF
    Slope is flat (3%-8%)
  THEN
    Site needs to be developed with terracing

4) IF
    Slope is rolling (> 8%)
  THEN
    Reconsider the site for its economic impact for development cost.

5) IF
    Slope is flat or rolling
  THEN
    Use asphalt as a pavement material

6) IF
    Slope is too flat (< 0.5%)
  THEN

Use Portland cement as a paving material

7) IF
Existing vegetation consists of abundant endangered species
or
Site is a habitat for endangered wild life species
THEN
Reconsider the site for its economic impact for development cost in terms of conservation criteria.

8) IF
Part of the site consists of wetland
and
Wetland area is over an acre
THEN
Site is to be replaced by two acres of wetland for every acre

9) IF
Existing vegetation is healthy
and
Age of vegetation is less than 10 years
THEN
Try to preserve trees more than 4" in dia.

10)IF
Existing vegetation is not healthy
and
Age of vegetation is more than 10 years
THEN
Omit deceased or old vegetation from preservation

11)IF
Site has a natural feature existing in it
and
Natural feature is stream
THEN
Try to preserve the woodland along stream to keep it natural.
(It helps in storm water management to avoid increased runoff)

12)IF
Soil is well drained
and
Bedrock and water table is low
and
Topography is flat (1%-8%)
THEN
1 - 10 type of drainage techniques can be used

13)IF
Soil is well drained
and
Bedrock and water table is low
and
Topography is rolling (8%-15%)
THEN
1,4-10 type of drainage techniques can be used

14)IF
Soil is well drained
and
Bedrock and water table is high
and
Topography is flat
THEN
2,8-10 type of drainage techniques can be used

15)IF
Soil is well drained
and
Bedrock and water table is high
and
Topography is rolling
THEN
8-10 type of drainage techniques can be used

16)IF
Soil is moderately well drained
and
Bedrock and water table is low
and
Topography is flat
THEN
2-4,6-10 type of drainage techniques can be used

17)IF
Soil is moderately well drained
and
Bedrock and water table is low
and
Topography is rolling
THEN
4,6-10 type of drainage techniques can be used

18)IF
Soil is moderately well drained
and
Bedrock and water table is high
and
Topography is flat
THEN
2,8-10 type of drainage techniques can be used

19)IF
Soil is moderately well drained
and
Bedrock and water table is high
and
Topography is rolling
THEN
8-10 type of drainage techniques can be used

20)IF
Soil is poorly drained
and
Bedrock and water table is low
THEN
7-10 type of drainage techniques can be used

21)IF
Soil is poorly drained
and
bedrock and water table is high
THEN
8-10 type of drainage techniques can be used

22)IF
Soil type is moderate-shrink-swell
THEN
Remove / bury 8-10" below top soil
and
Add 1-2" asphalt paving to strengthen surface

23)IF
Soil type is high-shrink-swell
THEN
Bring n top soil
and
Add 8-10" gravel sub base in the ground

24)IF
Soil type is less shrink swell
THEN
Use 1.5-.75" crusher run for sub base

25)IF
Gross leasable area of shopping complex is x sft
THEN
.0055 x number of cars need to be parked

26)IF
Gross leasable area of shopping complex is x sft.
THEN
2.0625 x sft. of area is needed for parking

27)IF
Objective is space saving in parking area design
THEN
Use 90 degrees parking bays

28)IF
Objective is circulation ease in parking area design
THEN
Use angled parking at less than 90 degrees

***QUESTIONS***

1) What is slope of the site ?
   very flat (< .5%)
   flat (1 - 3%)
   flat (3 - 8%)
   rolling (> 8%)

2) Are there any endangered plant species present on site ?
   y/n

3) Are there any endangered wild life species present on site ?
   y/n

4) Is part of the site wetland ? If yes is wetland area over an acre ?
   y/n

5) What is the condition of existing vegetation ?
   healthy
   deceased

6) What is the age of existing vegetation ?
   more than 10 years
   less than 10 years

7) Is there any natural feature present on site and what kind ?
   none
   stream

8) What is characteristic of the soil ?
   well drained
   moderately well drained
   poorly drained

9) What is characteristic of the top-soil ?
   moderate-shrink-swell
   high-shrink-swell
   less-shrink-swell

10) What is the level of bedrock and water table ?
   low
   high

11) What is the gross leasable area of shopping complex ?
   .... sft.

12) What is your objective in parking area design ?
   space saving
   circulation ease
   either

13) INFORMATION AVAILABLE ON
Permissible slope for different areas
Location of paved areas
Tips on storm water management techniques
Illumination standards.

### *3.451 Example of inferencing process*

To obtain a design recommendation about the storm water drainage techniques, following process takes place internally in the prototype. See Figure 8 for knowledge base architecture of the prototype.

After initiation of the software, the introductory screen is displayed and then a series of questions are asked on the screen. This instance is executed by rule 'ASK-DATA-1', which is

```
IF
(instance THE-PLAN is PLANNING with status :in- progress)
```

*i.e. the session is initiated.*

```
(instance START-PLAN is SCREEN-CONTROL
     with status :running
     with CURRENT-SCREEN WORK-SCREEN)
```

*i.e. the input information to the questions is simultaneously stored and displayed on the work sheet which is running simultaneously with the questionnaire session.*

```
 AND
(or (instance INPUT-DATA is INPUT-VALUES
          with-unknown SLOPE-VAL ?)
    (instance INPUT-DATA is INPUT-VALUES
          with-unknown SUB-SOIL-VAL ?)
    (instance INPUT-DATA IS INPUT-VALUES
          with-unknown DRAINAGE-CAP-VAL ?)
```

*i.e. When either of the three questions in the instance 'INPUT-DATA-1' are unanswered or none of them are answered at all.*

```
THEN
  (instance USER-OUTPUT is OUTPUT-WINDOW
       with display "Select OK when done")
  (instance INPUT-DATA-1 is SET-SLOT-VALUES
       with go :yes)
```

*This initiates the instance 'INPUT-DATA-1' and hence the display contents of this instance (i.e. three questions described above) are displayed on the screen. Also at the bottom of the screen, a message is displayed "select OK when done" through another output window, which is defined at the start of the program to display all the introductory messages.*

The instance 'INPUT-DATA-1' has 3 questions in it display contents. The third question is, "What is drainage capacity of sub-soil?" Options for this questions are 'well-drained', 'moderately-well-drained' and 'poorly-drained'. When the user brings the highlighted bar in front of this question, the abbreviated answers for all these three alternatives are displayed in a small window.

These answers are controlled by the slot values defined for the frame 'INPUT-VALUES'. e.g. in case of 'DRAINAGE-CAP-VAL', it has a slot named 'CONSTRAINTS' which is (:one-of (well moderate poor)) directly following the same sequence to that of the options displayed on the screen below the question in instance 'INPUT-DATA- 1'. Hence, when the user selects 'poor' from the screen by bringing highlighted bar on the option and pressing 'return' key, the slot value 'poor' is stored in the slot 'DRAINAGE-CAP-VAL' in the frame 'INPUT-VALUES'. Similarly, the slot value 'low' is stored in the slot 'WATER-TABLE-VAL' in the frame 'INPUT-VALUES' from instance 'INPUT-DATA-2' by answering the question "What is the depth of bedrock and water-table ?" by selecting the option, 'low'. The options for this questions 'high' and 'low' are defined as constraints (one-of (high low)) in slot 'WATER-TABLE-VAL' in frame 'INPUT-DATA'.

This process shows how all the values selected by the user in the questionnaire session are put in the frame 'INPUT-VALUES' in their respective slots. All the values are input in the system at the end when questions in the display of 'INPUT-DATA-3' are answered, and the work sheet appears on the screen with all the abbreviated values. The user should already be familiar with these abbreviations since the respective long-forms are displayed on the screen when questions are asked.

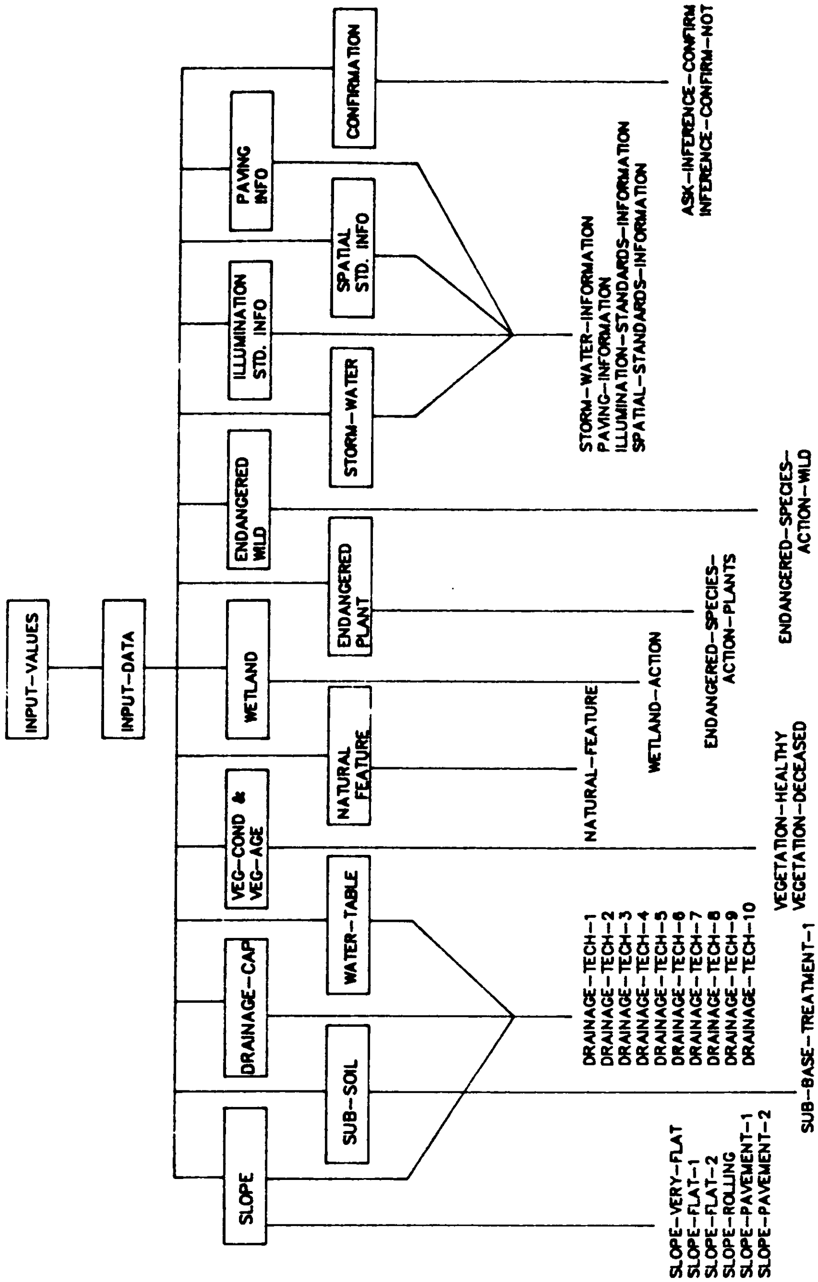



**Figure 8. KNOWLEDGE BASE OF THE PROTOTYPE**

At this stage 'WORK-SCREEN' allows the user to make any on-screen alterations in the input values. Then the user is prompted to answer the question at the bottom of the screen as to whether all these values displayed are correct or not. This question is stated in the instance 'INFERENCE-CONFIRM' with contents "Are all the values you have input in the system correct ?" and this question is displayed by firing of the rule 'ASK- INFERENCE-CONFIRM', which is

**IF**
**(instance THE-PLAN is PLANNING**
**with status :running**
**with current screen WORK-SCREEN)**
**(instance INPUT-DATA is INPUT-VALUES**
**with unknown CONFIRMATION ?)**

*i.e. The slot 'CONFIRMATION' in instance 'INPUT-DATA' in frame 'INPUT-VALUES' is yet unanswered.*

**THEN**
**(instance INFERENCE-CONFIRM is popup-confirm with go :yes)**

*i.e. display the contents of instance 'INFERENCE-CONFIRM' on the screen.*

When user selects the answer 'yes' to this question, the inference engine is invoked. To make the forward chaining mechanism work, an assertion is made with the help of the rule 'INTER-MEDIATE- ASSERTION'

**IF**
**(instance INFERENCE-CONFIRM is popup-confirm with answer :yes)**

*i.e. If the user has answered yes to the question for confirmation of input values.*

**THEN**
**(start-inference-engine)**
*i.e. Then make this assertion in the assertion base.*

When this assertion is made in the working memory, all the inference rules come into the agenda as every inference rule has this assertion as a part of their antecedent. To continue the discussion of following the process of reaching a particular assertion, at this stage the rule 'DRAINAGE-TECH-9' comes in the agenda.

**IF**

**(start-inference-engine)**

;; which is the assertion made by the rule 'INTERMEDIATE ASSERTION'

**(Instance INPUT-DATA Is INPUT-VALUES with DRAINAGE-CAP-VAL poor)**

**(Instance INPUT-DATA Is INPUT-VALUES with WATER-TABLE-VAL low)**

Since all three conditions in the antecedent of this rule are satisfied, this rule fires and its consequent executes, recommending any of 7-10 types of storm-water-drainage management techniques. This recommendation is then displayed on the screen at the end of inference process after firing of the rule 'PRINT-OUT-OUTPUT', which contains several LISP routines in its consequent to format of these recommendations on the screen. This display can then be directed to printer by pressing the 'print- screen' key to obtain a hard copy of the recommendations.

## 3.5 TESTING

### *3.51 Pretesting*

Emphasis on the pretesting of the software was mainly on the user interface rather than using it in the actual studio environment. The experts themselves were asked to evaluate software by asking them to use it a few times.

One of the problems in the prototype was inflexibility of the user interface during the input session. The user was not able to modify the input-values in the system before running the inference engine. To overcome this problem, a new instance 'WORK-SHEET' was introduced, which displays all the input-values simultaneously along with the questionnaire session and has on-screen editing capabilities. In addition to this, information displayed during the session in 'OUTPUT WINDOW' was made more elaborate and useful than before. In this process, the user interface was modified making it more flexible and informative than before. A few rules which were contradictory were pointed out by an expert and were also fixed.

### *3.52 Testing*

The prototype's effectiveness was tested on two groups of graduate design students and two groups of undergraduate design students in the studio environment. Each group consisted of five members assigned randomly. They were given a hypothetical site for developing a parking lot (Refer to appendix B for problem statement). Along with the site plan, factual information about the site (which the prototype was designed to handle) was provided. All groups were advised to document their analysis process on an analysis drawing. Each group was required to prepare one design scheme.

Two groups were given with prototype ES to use in the design process. All groups were allowed the same time to work on the problem (three hours). The designs prepared by all groups were evaluated for how effectively they used the given factual information about the site.

### *3.53 Results*

The software proved to be extremely user friendly. Average time taken by the team members on the system was in the range of 15 to 20 minutes. Groups were advised to divide the site into four subsites according to the criteria of their physical characteristics to produce modules for feeding information into the system. The time spent on the design problem by both the groups did not vary. Also the designs, when evaluated on previously discussed criteria, did not show any significant differences in the information processing in the design process.

Although the designs did not differ in the contents, the time taken for the designers to use the ES did not alter the overall time taken for the design. This shows that even such a limited system can work effectively with it's knowledge base built from designers heuristics. This gives some understanding of advantages of incorporating the more comprehensive *Architectural Design ES* in the design process. With the ES incorporating more in-depth knowledge, it will lead the designer to achieve better quality in design in accordance with the 'Better information relates to better design' theme.

When asked verbally, a common consensus was expressed by the users of the prototype that effectiveness of using this kind of system in the problem would have been enhanced if it covered a broader range of issues with greater depth. Some examples of further issues are soil classification, climatic data interpretation, etc. This suggestion gives some clues for deciding the scope for a prototype to demonstrate the effectiveness of a comprehensive system. Hence with such a limited nature of the expert system in this context, it was not possible to prove the effectiveness of the integrated ES with sufficient strength. The prototype itself needs to be much more comprehensive for testing in the studio environment.

# 4.0 CONCLUSIONS AND RECOMMENDATIONS

## 4.1 CONCLUSIONS

This prototype ES is designed to function as an aid in the pre-analysis stage of the design process. With its knowledge base design and effective user interface, it has been demonstrated that this type of ES can be developed on a larger scale. Results suggest that this study did not establish effectiveness mainly due to the limited information processed by the system. Hence even a prototype for preliminary research needs to be much more comprehensive, covering more in-depth issues used in the decision process in design.

The ES shell 'Goldworks' shows good promise for developing such a system on a much larger scale. Primary problems encountered in the using this shell were inadequate documentation from the manufacturer, unfriendly user manuals, slow inference and stringent hardware requirements. Advantages of using 'Goldworks' were availability of various features which are found in expensive ES shells, interface with LOTUS 123 & DBASE, support of a powerful 'Golden Common Lisp', and availability of such a versatile software in a P.C. environment.

With the process demonstrated in the problem identification, the comprehensive ES may be designed with modularity by different people using similar operating criteria at the top level and considering the interconnection of these modules at necessary levels. These modules may run independently or interdependently according to the extent of their interaction in the design process. For example, an ES module for building codes and an ES module for material specification may run interdependently to specify materials which follow the specifications recommended by building codes in a specific area. These modules can also interface with databases (most ES shells on the market come with this feature). Possibilities for integrating different modules are unlimited due to the complex nature of the design process. Such a comprehensive system can act an intelligent information retrieval facility, an intelligent tutor, an expert analyzer, etc. This type of system can act as a common platform for designers and researchers to exchange their expertise in an effective and meaningful manner. Accessibility to more and comprehensive information means better design solutions as discussed before. (See chapter 2.2)

This particular procedure shows promise to landscape architects and other design professionals in dealing issues of complexity in design associated with requirements (in terms of rules, standards and codes), concerns (like environmental and social), and efforts to establish stronger basis for design rationale to achieve higher goals in design quality.

## 4.2 RECOMMENDATIONS

The following recommendations may prove to be useful for anyone trying to duplicate this kind of research or carry it further.

***Choosing and defining a problem :*** It is important to know whether the decisions you want the system to make are clearly defined. A few questions may be worth answering during the process of problem selection.

1. Can you predict and control the data ?

2. Are there rules, or do the best solutions seem to come by intuition ?

3. Is the necessary knowledge limited enough in scope to be contained in the system ? If not, can you modularize the problem, possible and how effectively you can specify the modules ?

4. Are experts available for the problem domain ? Will they cooperate ? Will they have the time needed ?

5. Is there an easier way to solve this problem ? Does a knowledge based system provide an added value compared to an existing solution ?

6. Is there a commitment to a problem, to the process and to making it work ?

It is very important to start small in the prototype design process. It may not be important to demonstrate the utility of the larger system with the prototype completely in the first shot, but it is very important to judge whether or not the kind of approach you are taking for knowledge base design and user interface is going to work. It is also important that the problem domain you have selected be clearly defined in the context of a bigger problem domain to make sure you are working in a meaningful direction.

In particular, do not attempt the full knowledge engineering at an early stage. For one thing, the problem and knowledge domains may change as you learn more about the problem. One key to successful prototyping is proper division of the problem into modules. When this is well done, each module will make sense by itself as a solution to a small problem or microworld. And last but not least, while dealing with multiple experts with subtle differences, make sure you all agree on a single authoritative source for resolving disputes before you start building the system.

# 5.0 EPILOGUE

## 5.1 FUTURE PROSPECTS OF COMPUTER APPLICATION IN DESIGN

With a dreamer's viewpoint this subject can be discussed in an endless essay. From automated computer aided design to including holographic techniques in design process for spatial studies, many of the possibilities in the near / distant future may be included in this essay. The intention of this research was to explore only a fraction of the dream of using computers in the information processing with human intelligence.

Expert systems seem to play an important role in this direction. The versatility of presently available ES shells in terms of integration with database software, flexibility in knowledge base designs, etc. make it possible to visualize further progress of ESs. A designer's *dream machine* may be visualized in the near future which will consist of a comprehensive software with capabilities of intelligent information processing with supportive intelligent graphic packages.

In Omar Akin's* protocol for representation in the design process, these kinds of intelligent CAD package may play an important role. Libraries of different representations knitted with knowledge bases may be able to work as inspiration guidance machines for designers to help them visualize and strengthen their thinking process. Intelligent information access along with this process may enrich the experience of designing bringing dreams on realism by tracking them towards practical direction. A step forward may be to use these kind of systems in training cycles of Neural Networks.** These systems show potential in achieving the concept of 'informed design', accountibility in a more complexing environment, and giving the landscape architect more freedom in design process.

Affordability of software on microcomputer based work-stations has made it possible to bring these kind of systems in reach of more and more people, making advances in this field easier. A question, last but not least, arises in the reader's mind, as to whether these machines are intended to replace man ? I think that they are not intended to replace man, but help to enrich the processes used by man using the expertise he has developed. They reinforce the power of visualization and logical thinking in the designer's mind, which may make it feasible to derive designs with better quality with their help.

---

* Akin, O. ***Representation and Architecture*** Information dynamics inc, 1982.

** Eckmiller R. ***Neural Computers*** Springer-Verlag, 1988.

# 6.0 BIBLIOGRAPHY


1. Adeli, H., ***Expert systems in construction and structural engineering.*** Chapman and Hall, 1988.

2. Akin, O. ***How do architects design?*** in Artificial Intelligence and Pattern Recognition in Computer Aided Design, Latombe, J., ed. Amsterdam: North-Holland, 1987.

3. Akin, O., ***Representation and Architecture*** Information dynamics inc, 1982.

4. Blacksburg landuse plan: A component of Blacksburg comprehensive plan, 1977.

5. Brandon, P. S. and Sidney N., ***The development of a knowledge based system for setting an economic strategy in building design, inproceedings of the International Council for building Research, Studies and Documentation 1986 Congress.*** Gaithersburg, Maryland: Center for Building Technology, National Bureau of Standards, 1986.

6. Building Research Board, National Research Council. ***The 1984 workshop on advanced technology for building design and engineering*** Washington, D. C. : Building Research Board, National Research council, 1985.

7. Conye, R. D. and Gero, J. S., ***Design Knowledge and sequential plans, working paper. sydney, Australia: Computer Applications Research Unit.*** Department of Architectural Science, University of Sydney, 1984.

8. Eastman, C. M., ***Spatial synthesis in computer aided building design.*** Applice Science Publishers Ltd. 1975.

9. Fenves, S. J., ***Class notes for expert systems in civil engineering, technical report*** Pittsburgh, Pennsylvania: Department of Civil Engineering, Carnegie-Mellon University, 1983.

10. Hayes-Roth, F., D Waterman and D. Lenat, eds. ***Building expert systems*** Reading, Massachusetts: Addison-Wesley, 1983.

11. Ibbs, C. W., ***Proceedings of a workshop for the development of new research direction in computerized applications to construction engineering and management studies***, Construction Research Series Report no. 19., Chicago, Illinois: University of Illinois, July 1985.

12. Latombe J. C. ***Artificial intelligence and pattern recognition in computer aided design.*** North Holland, 1975.

13. Lansdown, J., ***Expert systems: their impact on the construction industry.*** London: Royal Institute of British Architects Conference Fund, 1982.

14. Lynch, K., ***Site planning*** MIT press, 1984.

15. McGartland, M., ***Expert system applications for construction project management (master of Science thesis).,*** Pittsburgh, Pennsylvania: Carnegie-Mellon University 1983.

16. Melkanoff, M. A., ***AI, expert systems and architecture***, (paper presentation, NSF Workshop on the Computational Foundations of Architecture, University of California, LA, 1985). Washington, D. C. : National Science Foundation, 1985.

17. Mitchell, W. J., ***Computer Aided Architectural Design*** Maron / Charter Publishers, inc. 1977.

18. Radford, A. D. and Mitchell, J. R., ***Automated architectureal detailing: a knowledge-based approach***, in Proceedings of the Documentation 1986 Congress. Gaithersburg, Maryland: Center for Building Technology, National Bureau of Standards, 1986.

19. Skitmore, Martin, R., ***Towards an expert conceptual estimating system.*** In Proceedings of the International Council for Building Research, Studies, and Documentation 1986 Congress. Gaithersburg, Maryland: Center for building Technology, National Bureau of Standards, 1986.

20. Sriram, D., Maher M. L., Bielak J. and Fenves S. J., ***Expert systems for civil engineering-a survey*** Technical Report. Pittsburg, Pennsylvania: Carnegie-Mellon University, July 1982.

21. Stone, D. and Wilcox, D., ***Formulation of building regulations using interactive logic programs***, in Proceedings of the International Council for Building Research, Studies, and Documentation 1986 Congress. Gaithersburg, Maryland: Center for building Technology, National Bureau of Standards, 1986.

22. Tourbier J. T., ***Water resource protection technology.*** Urban Land Institute, 1981.

## 6.1 PERIODICALS


### *A/E Systems Report*

1. Kellogg, H. B., ***A/E CADD and artificial intelligence.*** A/E Systems Report, Vol. 11, August 1987, pp. 3-4.

### *Architect's Journal*

2. Lansdown, J., ***Expert systems 1: knowledge for designers*** Architect's Journal, Vol. 179, February 8, 1984, pp. 55-58.

3. Lansdown, J., ***Expert systems 2: a prototype for designers*** Architect's Journal, Vol. 179, February 15, 1984, pp. 75-77.

### *Architecture*

4. Gordon, D. E. and Stubbs, S. M., ***Knowledge-based and expert systems, in A Global look at construction***, Architecture, Vol. 76, September 1987. pp 106-107.

5. Witte, O. R., ***The uses of computers in management of information*** Architecture, June 1987. pp 116-118.

### *Architectural Science Review*

6. Conye, R., ***Knowledge-based planning systems and design: a review.*** Architectural Science Review, Vol. 28, December 1985, pp 95-103.

7. Radford A. D., ***Approaches to knowledge based architectural detailing.*** Architectural science review, pp 88-94.

***Artificial Intelligence***

8. Lowe,D. G., ***Three-dimensional object recognition from single two-dimensional images***, Artificial Intelligence, Vol. 31, March 1987, pp. 355-395.

9. Radford, Antony D., ***Approaches to knowledge-based architectural detailing***, Architectural Science Review, Vol. 28, December 1985, pp. 85-114.

***ASMT Standardization News***

10. Kaufman, J. G., ***Towards standards for computerized material property data and intelligent knowledge systems***, Vol. 15, March 1987, pp. 38-43.

***Building***

11. Hamilton, Gavin., ***Constructuin leads way in expert systems software***, Building, vol. 250, March 14, 1986, pp. 77+.

12. Stewart, Alastair., ***Expert systems--will they work?*** Building, Vol. 251, December 12, 1986.

***Bulletin of Computer Aided Architectural Design***

13. Lansdown, J., ***Expert Systems: a memorandum, on their Scope and nature***, Bulletin of Computer Aided Architectural Design, No. 38, November 1980, pp. 25-36.

***Building Research and Practice***

14. Trimble, G., ***Expert systems in construction***, Building Research and Practice, Vol. 13, July/August 1985, pp. 231-233. .br

***Byte***

15. Thompson, B. and Thompson, W., ***Finding rules in data*** Byte, Vol. 11, November 1986, pp. 149-152+.

***Civil Engineering***

16. Fenves, S. J., ***Expert systems: C. E. potential***, vol. 54, October 1984, pp. 44-47.

***Computer***

17. Bic, L. and Gilbert, J. P., ***Learning from AI: new trends in database technology.***, Computer, Vol. 19, March 1986, pp. 44-45.

18. Brandon, Peter S. and Sydney Newton., ***Improving the forecast: knowledge based systems***, Chartered Quantity Survey, Vol. 8, May 1986, pp. 24-26.

19. Faught, W. S., ***Applications of AI in engineering*** Computer, Vol. 19, July 1986, pp 17-23+.

20. Havens, W. and R. Mackworth., ***Representing knowledge of the visual world***, Computer, Vol. 16, October 1983, pp 90-96.

21. Nau, D. S., ***Expert computer systems***, Computer, Vol. 16, February 1983, pp.63-65.

***Computer Aided-Design***

22. Radford, A. D. and Gero, J. S., ***Towards generative expert systems for architectural detailing***, Computer-Aided Design, Vol. 17, November 1985, pp 428-435.

23. Rechak, D. R. and Howard, H. C., ***Interfacing expert systems with design databases in integrated CAD systems***, Computer-aided design, Vol, November 1985, pp. 443-454.

24. Rosenman, M. A., ***Expert systems applications in computer-aided design***, Computer-Aided Design, Vol. 18, April 1986, pp. 132-138.

### *Computer Design*

25. Stefik, M. J. and Dekleer, J., ***Prospects for expert systems in CAD***, Computer Design, Vol. 22, April 21. 1983, pp. 65-66 + .br

### *Construction Specifier*

26. Carroll, C. R., Jr., ***Computer-aided creativity-- have we found it?***, The construction Specifier, Vol. 39, March 1986, pp. 21-23.

### *Consulting Engineer*

27. Markbury, D. L., ***Artificial intelligence: CAE's outer limits*** Consulting Engineer, Vol. 63, December 1984, pp. 39-41. .br

### *Construction En Systems*

28. Holnapy D., ***Language formal pour les systems construction*** 1981, pp 559-667.

### *Datamation*

29. Michaelsen, R. and Michie, D., ***Expert systems in business*** Datamation, Vol. 29, November 1983, pp.240-244+.

30. Myers, E., ***Making micro experts (knowledge-based systems on pcs)***, Datamation, Vol. 31, September 1, 1985, pp. 50-51+. .br

## *Design news*

31. ***Artificial intelligence-what's holdup?*** Design News, Vol. 43, February 23.1987, p.42.

32. Dietz, P. W., ***Artificial intelligence: building rulė- based expert systems***, Design news, Vol.42, March 3, 1986, pp 136-138.

33. Firdman, H. E., ***Artificial intelligence: understanding the basic concepts***, Design News, Vol. 42, March 3, 1986 pp. 89-92+.

34. Kinoglu, F., ***Artificial intelligence: expert system model of the design process***, Design News, Vol. 42, March 3, 1986, pp. 149-152+.

## *Design Studies*

35. Cathain, C. O., ***Expert systems and design.*** Design studies, Vol. 8, No. 2, April 1987.

36. Fawcell, W., ***A note on the logic of design.*** Design Studies, Vol.8, NO. 2, April 1987.

37. Gregory, S. A., ***Designer Support Environment- Expert systems.*** Design studies, Vol. 8, No. 2, April 1987.

38. Magee, Kevin ***The elicitation of knowledge from designers.*** Design studies, Vol. 8, No. 2, April 1987.

39. Schon, D. A., ***Designing- Rules, type and worlds.*** Design studies, Vol. 9, No. 3, July 1988.

***Electronic Design***

40. Schindler, M., ***Artifiicial Intelligence begins to pay off with expert systems for engineering***, Electronic Design, Vol. 32, August 9, 1984, pp. 106-120+.

***Engineering.***

41. Swift, K. and Matthews, A., ***Expert computer systems in engineering design***, Engineering(London), Vol. 223, September 1983, pp. 673-678.

***Engineering News-Record***

42. Hoffmann, P. ***Concrete experts pool Knowledge***, Engineering News-Record, Vol. 217,October 9, 1986, p 19.

***Environment and Planning***

43. Akin, O., ***A formalism for problem restructuring and resolution in design*** Environment and Planning B. Planning and design, 1986 Vol. 13, pp 223-232.

***Journal of Construction Engineering and Management***

44. Ibbs, C. W., ***Future directions for computerized construction research***, Journal of Construction Engineering and management, Vol. 112, September 1986, pp. 326-345.

***Heating/Piping/Air Conditioning***

45. Wright, V. E., ***Expert systems in HVAC design***, Heating/Piping/Air Conditioning, Vol. 57, January 1985, pp. 125-131.

***High Technology***

46. Davis, D. B., ***Artificial intelligence goes to work.*** High technology, April 1987, pp 16-27.

***Journal of Surveying Engineering***

47. Robinson, V. B., ***Introduction to expert systems for land information systems***, Journal of Surveying Engineering, Vol. 112, October 1986, pp. 109-118.

***Engineering Design News***

48. Mosley, J. D., ***Expert systems as engineering tools will broaden productivity/creativity option***, EDN, Vol. 31, November 13, 1986, pp. 91+.

***Fire Technology***

49. Smith, R. L., ***ASKBUBJr a precursor of an expert system for the evaluation of fire hazard***, Fire Technology, Vol. 23, February 1987, pp. 5-18.

***IEEE Computer Graphics and Applications***

50. Myers, W., ***Artificial intelligence computers applied to computer graphics***, IEEE Computer Graphics and Applications, Vol. 3, July 1983, pp.78.

***Iron and Steel Engineer***

51. Fischer, C. H.,***Expert systems and artificial intelligence***, Iron and Steel Engineer, Vol. 62, November 1985, pp. 58-62.

52. McGartland, Martin R. and Chris T. Hendrickson., ***Expert systems for construction project monitoring***, Journal of Construction Engineering and Management, Vol. 111, September 1985, pp. 293-307.

### *Landscape Architecture*

53. Petrich, C., ***Expert systems: Forecasting powerful support for the designer***, Landscape Architecture, Vol. 76, May/June 1986, pp. 70-74.

### *Optical Engineering*

54. Subramanian, V., ***Document retrieval using a fuzzy knowledge-based system***, Optical Engineering, Vol. 25, March 1986, pp. 445-455.

### *Progressive Architecture*

55. Fisher, T. and Doubilet, S., ***Intelligent computers***, Progressive Architecture, Vol. 67, June 1986, pp. 104-113.

56. Stahl, Fred I., ***Expert systems***, Progressive Architecture, Vol. 65, October 1984, pp. 61-63.

### *Simulation*

57. ***Software tool kit for building expert systems on IBM PC- AT,_ PC-XT, and UNIX-based computers (RuleMaster)***, Simulation, Vol. 45, August 1985, pp. 98.

*Technology Review*

58. Shurkin, J. N., ***Expert systems: the practical face of artificial intelligence***, Technology Review, Vol. 86, November/December 1983, pp. 72-78.

## 6.2 NEWSLETTERS

1. Harmon, P. ***Expert systems strategies***, Cutter information corp, Vol. 4 No. 4-10, 1988.

# APPENDIX A

## PROTOTYPE EXPERT SYSTEM.

```
;;; -*-Mode:LISP;
Package:GW; -*-
;;;
;;; GoldWorks Knowledge Base
;;;
;;; Dumped on  9:08am  22-Mar-89
;;; For registered user: DR. T. W. BONHAM - 3 COLLEGE OF BUSINESS, VIRGINIA TECH ;;;
;;; Portions of Knowledge Base saved:
;;; (FRAME RELATION INSTANCE ASSERTION RULE)
;;;
(in-package 'gw)
```

***LISP FUNCTIONS-***

```
(defparameter recom-list ())

(defun recon-line (operation &optional args)

  (case operation

    (:write-line

      (setf recom-list (append recom-list (list args))) args)      (:which-operations

       '(:write-line :which-operations))

    (stream-default-handler recom-line operation :write-line)))




(defparameter *start-plan-day* 1)

(defparameter *start-plan-month* 1)

(defparameter *start-plan-year* 1989)

;;GET DAY, MONTH AND YEAR FROM SYSTEM AND SET DATE VARIABLES

(defun update-date (&aux second minute hour date month year)

  (multiple-value-setq (second minute hour date month year)

  (get-decoded-time))

  (setf *start-plan-day* date)

  (setf *start-plan-month* month)

  (setf *start-plan-year* year))



;;RETRACT-ALL-SLOTS finds all the slots of an instance and ;;retracts all values from the

slot


(defun retract-all-slots (inst)

   (dolist (slot (instance-all-slots inst))

     (slot-retract-value inst slot)))
```

```
(defparameter RECOM-LIST ())

(defparameter FINAL-LIST ())

(defparameter COUNT-LIST-ELM 0)

(defparameter COUNTER-R 0)



(defun final-print (args)
   (cond
      ( (null args)
        (SEND INSTRUCTION-WINDOW :CLEAR-SCREEN)
        (send instruction-window :write-string
"PRESS <PRINT-SCREEN KEY> TO PRINT AND THEN HIT <ENTER>"))
      ((equal counter-r 5)
       (setf instruction-window
            (make-window-stream :left 0
                                :attribute 4
                                :top 24
                                :height 4
                                :width 80))
        (send instruction-window :write-string
"PRESS <PRINT-SCREEN KEY> TO PRINT AND THEN HIT <ENTER>")
      (read-line)
        (setf counter-r 0)
        (send *terminal-io* :clear-screen)
        (final-print new-args))
        ((listp args)
```

```
(setf new-args (rest args))
(setf counter-r (incf counter-r))
(setf count-list-elm (incf count-list-elm))
(format t "¬%¬%¬A¬A" count-list-elm (first args))
(final-print (rest args)))
(t nil)))
```

***FRAMES-***

```
(DEFINE-FRAME INPUT-VALUES
  (:print-name "INPUT-VALUES"
   :doc-string "This frame is made to hold the slots for the input-values from the user."
   :is TOP-FRAME)
 (SLOPE-VAL
   :doc-string "existing slope of the site"
   :constraints (:one-of (very-flat flat1 flat2 rolling)))

 (SUB-SOIL-VAL
   :doc-string "shrink-swell capacity of sub-soil"
   :constraints (:one-of (high moderate less)))
 (DRAINAGE-CAP-VAL
   :doc-string "drainage capacity of sub-soil"
   :constraints (:one-of (well moderate poor)))
 (WATER-TABLE-VAL
   :doc-string "depth of bedrock and water table"
   :constraints (:one-of (low high)))
```

```
(VEG-COND-VAL
   :doc-string "condition of existing vegetation"
   :constraints (:one-of (healthy deseased none)))
(VEG-AGE-VAL
   :doc-string "age of existing vegetation"
   :constraints (:one-of ( > 10yr  < 10yr)))
(NATURAL-FEATURE-VAL
   :doc-string "natural feature present on site"
   :constraints (:one-of (none stream)))
(WETLAND-VAL
   :doc-string "whether part of the site is wetland"
   :constraints (:one-of (yes no)))
(ENDANGERED-PLANT-VAL
   :doc-string "endangered plant species"
   :constraints (:one-of (yes no)))
(ENDANGERED-WILD-VAL
   :doc-string "endangered wild life habitat"
   :constraints (:one-of (yes no)))
(STORM-WATER-INFO
   :doc-string "slot to confirm output window for storm water management information"
   :constraints (:one-of (yes no)))
(ILLUMINATION-STD-INFO
   :doc-string "slot to confirm information screen for illumination standards"
   :constraints (:one-of (yes no)))
(SPATIAL-INFO
   :doc-string "slot to confirm information for spatial std. information"
   :constraints (:one-of (yes no)))
(PAVING-INFO
```

```
    :doc-string "slot to confirm information screen for paving information"
    :constraints (:one-of (yes no)))
  (CONFIRMATION
    :doc-string "slot to confirm the begining of inference engine"
    :constraints (:one-of (:yes :no)))
  (DIRECT-RUN
    :doc-string "slot to confirm the running of inference engine from work-screen"
    :constraints (:one-of (:yes :no))))


(DEFINE-FRAME PROJECT
  (:print-name "PROJECT"
   :doc-string "An instance of this frame represents the project information."
   :is TOP-FRAME)
  (PROJECT-NAME
     :constraints (:LISP-TYPE STRING))
  (DESIGNER-NAME
     :constraints (:LISP-TYPE STRING))
  (PROJECT-NUMBER
     :default-values  (1)))


(DEFINE-FRAME PLANNING
  (:print-name "PLANNING"
   :doc-string "Instance of this frame is used to represent the status of the consultation
session."
   :is TOP-FRAME)
  (END-SESSION)
```

```
(STATUS)
(CLEAR)
(MONTH-NUMBER)
(DAY-NUMBER)
(YEAR)
(DATE-STRING
  :default-values ("Unknown Date")))
```

***INSTANCES-***

```
(DEFINE-INSTANCE INPUT-DATA-3
  (:print-name "INPUT-DATA-3"
   :doc-string "THIRD SET OF QUESTIONS TO BE ASKED"
   :is  SET-SLOT-VALUES)
  (CENTER :X-AND-Y)
  (BORDER-COLOR :BLUE)
  (REVERT-BUTTON :YES)
  (DEFAULT-BUTTON :NO)
  (INSTRUCTIONS
" PLEASE ANSWER FOLLOWING QUESTIONS
 SELECT THE CHOICES PRESENTED WITH ARROW KEYS THEN HIT ENTER KEY ")
  (CONTENTS
     ( (INPUT-DATA NATURAL-FEATURE-VAL
" * IS THERE ANY NATURAL FEATURE PRESENT ON SITE & WHAT KIND ?  -->
   1. None
   2. Stream")
```

```
        (INPUT-DATA WETLAND-VAL
" * IS PART OF THE SITE WETLAND ? DOES IT EXCEEDS OVER AN ACRE ?  -->
    1. Yes
    2. No")
        (INPUT-DATA ENDANGERED-PLANT-VAL
" * ARE THERE ANY ENDANGERED PLANT SPECIES PRESENT ON SITE ?  -->
    1. Yes
    2. No")
        (INPUT-DATA ENDANGERED-WILD-VAL
" * ARE THERE ANY ENDANGERED WILD-LIFE SPECIES PRESENT ON SITE ?  -->
    1. Yes
    2. No"))))


(DEFINE-INSTANCE INPUT-DATA-2
   (:print-name "INPUT-DATA-2"
    :doc-string "SECOND SET OF QUESTIONS TO BE ASKED"
    :is  SET-SLOT-VALUES)
  (CENTER :X-AND-Y)
  (BORDER-COLOR :BLUE)
  (REVERT-BUTTON :YES)
  (DEFAULT-BUTTON :NO)
  (INSTRUCTIONS
" PLEASE ANSWER FOLLOWING QUESTIONS
  SELECT THE CHOICES PRESENTED WITH ARROW KEYS THEN HIT ENTER KEY ")
  (CONTENTS
      ( (INPUT-DATA WATER-TABLE-VAL
" * WHAT IS THE DEPTH OF BEDROCK AND WATER TABLE ?  -->
```

```
1. Low
2. High" )
  (INPUT-DATA VEG-COND-VAL
" * WHAT IS THE CONDITION OF EXISTING VEGETATION ?  -->
1. Healthy
2. Deseased
3. None existing" )
  (INPUT-DATA VEG-AGE-VAL
" * WHAT IS THE AVERAGE AGE OF EXISTING VEGETATION ?  -->
1. More than 10 years
2. Less than 10 years
3. Not applicable" ))))


(DEFINE-INSTANCE INPUT-DATA-1
  (:print-name "INPUT-DATA-1"
   :doc-string "FIRST SET OF QUESTIONS TO BE ASKED"
   :is  SET-SLOT-VALUES)
 (CENTER :X-AND-Y)
 (BORDER-COLOR :BLUE)
 (REVERT-BUTTON :YES)
 (INSTRUCTIONS
" PLEASE ANSWER FOLLOWING QUESTIONS
 SELECT THE CHOICES PRESENTD WITH ARROW KEYS THEN HIT ENTER KEY ")
 (DEFAULT-BUTTON :NO)
 (CONTENTS
   ( (INPUT-DATA SLOPE-VAL
" * WHAT IS SLOPE OF THE SITE ?  -->
```

```
      1. Very flat (< .5%)

      2. Flat1     (1 -3%)

      3. Flat2     (3 -8%)

      4. Rolling   (>  8%)" )

        (INPUT-DATA SUB-SOIL-VAL

" * WHAT IS THE CHARACTERISTICS OF SUB-SOIL ?  ->

      1. High-shrink-swell

      2. Moderate-shrink-swell

      3. Less-shrink-swell" )

        (INPUT-DATA DRAINAGE-CAP-VAL

" * WHAT IS DRAINAGE CAPACITY OF SUB-SOIL ?  ->

      1. Well drained

      2. Moderately well drained

      3. Poorly drained" ))))



(DEFINE-INSTANCE INFORMATION

  (:print-name  "INFORMATION"

   :doc-string  "information screens"

   :is SET-SLOT-VALUES)

  (CENTER :X-AND-Y)

  (BORDER-COLOR :BLUE)

  (REVERT-BUTTON :YES)

  (DEFAULT-BUTTON :NO)

  (INSTRUCTIONS

" PLEASE ANSWER FOLLOWING QUESTIONS

  SELECT THE CHOICES PRESENTED WITH ARROW KEYS THEN HIT ENTER KEY ")

  (CONTENTS
```

```
( (INPUT-DATA STORM-WATER-INFO
" * WOULD YOU LIKE TO HAVE INFO ON STORM WATER MANAGEMENT ?  -->
  1. Yes
  2. No")
   (INPUT-DATA PAVING-INFO
" * WOULD YOU LIKE TO HAVE INFO ON PAVING & PERMISSIBLE SLOPES ?  -->
  1. Yes
  2. No")
   (INPUT-DATA ILLUMINATION-STD-INFO
" * WOULD YOU LIKE TO HAVE INFO ON ILLUMINATION STANDARDS ?  -->
  1. Yes
  2. No")
   (INPUT-DATA SPATIAL-INFO
" * WOULD YOU LIKE TO HAVE INFO ON SPATIAL STANDARDS ?  -->
  1. Yes
  2. No"))))


(DEFINE-INSTANCE GO-MENU
  (:print-name "GO-MENU"
   :doc-string "Menu to start the investment session"
   :is POPUP-CHOOSE)
  (TARGET-SLOT STATUS)
  (TARGET-INSTANCE THE-PLAN)
  (BORDER-COLOR :MAGENTA)
  (CENTER :X-AND-Y)
  (CONTENTS
    ( ("     *START*     " :START)
```

```
("*CLEAR, THEN START*" :CLEAR-THEN-START))))


(DEFINE-INSTANCE WORK-SCREEN
  (:print-name "WORK-SCREEN"
   :doc-string "Work screen for obtaining input data from the user."
   :is  SCREEN-LAYOUT)
  (SYSTEM-MENU :YES)
  (SCREENS-MENU :YES)
  (MENU-BAR-ITEMS
     ( ("   QUESTIONS"
      ( (:MENU PROJECT-DATA
          "PROJECT NAME, NUMBER AND DESIGNER'S NAME")
        (:MENU INPUT-DATA-1
         "TOPOGRAPHY AND SOIL INFORMATION")
        (:MENU INPUT-DATA-2
         "WATER TABLE AND VEGETATION")
        (:MENU INPUT-DATA-3 "CONSERVATION ISSUES")
        (:MENU INFORMATION "INFORMATION SCREENS")))
      ("   RUN"
      ( (:MENU RUN-MENU "RUN INFERENCE ENGINE")))
      ("   END-SESSION"
      ( (:MENU CLEAR-CONFIRM-MENU "END CONSULTATION")))))
  (MENU-BAR-BORDER-COLOR :LIGHT-GRAY)
  (MENU-BAR-TEXT-COLOR :LIGHT-GRAY)
  (SCREEN-TEMPLATES
     ( (PROJECT-TITLE :LEFT 5 :TOP 2 :WIDTH 70 :HEIGHT 6)
  (INPUT-VALUES-1 :LEFT 0 :TOP 8 :WIDTH 40 :HEIGHT 12)
```

```
(INPUT-VALUES-2 :LEFT 40 :TOP 8 :WIDTH 40 :HEIGHT 12))))


(DEFINE-INSTANCE PROJECT-DATA
  (:print-name "PROJECT-INFORMATION"
   :doc-string "Pull down menu to ask the project information."
   :is SET-SLOT-VALUES)
  (CENTER :X-AND-Y)
  (BORDER-COLOR :BLUE)
  (REVERT-BUTTON :YES)
  (INSTRUCTIONS
" PLEASE INPUT THE FOLLOWING INFORMATION ABOUT CURRENT PROJECT ")
  (DEFAULT-BUTTON :NO)
  (CONTENTS
    ((THE-PROJECT DESIGNER-NAME
      "PLEASE TYPE IN YOUR NAME HERE        ")
    (THE-PROJECT PROJECT-NAME
      "PLEASE TYPE IN PROJECT NAME HERE     ")
    (THE-PROJECT PROJECT-NUMBER
      "PLEASE TYPE IN PROJECT NUMBER HERE   "))))


(DEFINE-INSTANCE WELCOME-SCREEN
  (:print-name "WELCOME-SCREEN"
   :doc-string "Initial screen of the application site-planner."
   :is  SCREEN-LAYOUT)
  (SYSTEM-MENU :YES)
  (SCREENS-MENU :YES)
```

```
(MENU-BAR-BORDER-COLOR :LIGHT-GRAY)
(MENU-BAR-TEXT-COLOR :LIGHT-GRAY)
(SCREEN-TEMPLATES
  ( (WELCOME-FORM :LEFT 5 :TOP 5 :WIDTH 70 :HEIGHT 15))))


(DEFINE-INSTANCE START-PLAN
 (:print-name "START-PLAN"
  :doc-string "Site-planner is a system to help designer in the planning process."
  :is SCREEN-CONTROL)
 (SCREEN-LAYOUTS (WELCOME-SCREEN WORK-SCREEN))
 (TEMPLATE-AREA-HEIGHT 19)
 (STATUS :INITIAL-STATUS))


(DEFINE-INSTANCE THE-PROJECT
 (:print-name "THE-PROJECT"
  :doc-string "This instance represents the project in this application."
  :is PROJECT)
 (PROJECT-NUMBER 1)
 (PROJECT-NAME)
 (DESIGNER-NAME))


(DEFINE-INSTANCE INPUT-DATA
 (:print-name "INPUT-DATA"
  :doc-string "To store the values obtained from work screen."
  :is INPUT-VALUES))
```

```
(DEFINE-INSTANCE THE-PLAN
  (:print-name "THE-PLAN"
   :doc-string "This instance represent the status of consultation session."
   :is  PLANNING))



(DEFINE-INSTANCE PROJECT-TITLE
  (:print-name "PROJECT-TITLE"
   :doc-string ""
   :is  SCREEN-TEMPLATE)
  (OBJECTS
       ( (DESIGNER-NAME :SLOT THE-PROJECT DESIGNER-NAME :WIDTH 50 :COLOR
:WHITE)
          (PROJECT-NAME :SLOT THE-PROJECT PROJECT-NAME :WIDTH 35 :COLOR
:CYAN)
       (PROJECT-NUMBER :SLOT THE-PROJECT PROJECT- NUMBER :WIDTH 20 :COLOR
:WHITE)))
  (CONTENTS
     ( ("DESIGNER'S NAME    " DESIGNER-NAME)
       ("PROJECT NAME       " PROJECT-NAME)
       ("PROJECT-NUMBER     " PROJECT-NUMBER)))
  (TEXT-COLOR :LIGHT-GRAY)
  (BORDER-COLOR :LIGHT-GRAY)
  (BORDER :SINGLE)
  (ORIENTATION :ROW)
  (LAYOUT :LINEAR)
```

```
(SPACES-BETWEEN-COLUMNS 0)
(SPACES-BETWEEN-ROWS 0))


(DEFINE-INSTANCE RUN-MENU
  (:print-name  "RUN-MENU"
   :doc-string  "to run the inference engine directly from the work-screen"
   :is SET-SLOT-VALUES)
  (center :x-and-y)
  (border-color :RED)
  (instructions "TO START THE INFERENCE ENGINE ")
  (contents
    ( (INPUT-DATA DIRECT-RUN
" ARE ALL THE VALUES YOU HAVE CHANGED IN WORK-SCREEN CORRECT ? --> "))))


(DEFINE-INSTANCE INPUT-VALUES-1
  (:print-name "INPUT-VALUES-1"
   :doc-string "WILL DISPLAY INPUT VALUES"
   :is  SCREEN-TEMPLATE)
  (OBJECTS
    ( (SLOPE :SLOT INPUT-DATA SLOPE-VAL :WIDTH 1)
      (SHRINK-SWELL :SLOT INPUT-DATA SUB-SOIL-VAL :WIDTH 1)
      (SOIL-DRAINAGE :SLOT INPUT-DATA DRAINAGE-CAP- VAL :WIDTH 1)
      (WATER-TABLE :SLOT INPUT-DATA WATER-TABLE-VAL :WIDTH 1)
      (VEG-CONDITION :SLOT INPUT-DATA VEG-COND-VAL :WIDTH 1)))
  (CONTENTS
    ( ("SLOPE OF THE SITE    " SLOPE)
```

```
        ("")
        ("SHRINK SWELL        " SHRINK-SWELL)
        ("")
        ("SOIL DRAINAGE       " SOIL-DRAINAGE)
        ("")
        ("WATER TABLE         " WATER-TABLE)
        ("")
        ("VEG CONDITION       " VEG-CONDITION)))
    (TEXT-COLOR :LIGHT-GRAY)
    (BORDER-COLOR :LIGHT-GRAY)
    (BORDER :SINGLE)
    (ORIENTATION :ROW)
    (LAYOUT :LINEAR)
    (SPACES-BETWEEN-COLUMNS 0)
    (SPACES-BETWEEN-ROWS 0))


(DEFINE-INSTANCE CLEAR-CONFIRM-MENU
    (:print-name "CLEAR-CONFIRM-MENU"
      :doc-string "Asks if the user really wants to clear all input so far and return to the
welcome screen."
     :is POPUP-CONFIRM)
    (TARGET-SLOT END-SESSION)
    (TARGET-INSTANCE THE-PLAN)
    (INSTRUCTIONS "END OF CONSULTING SESSION:")
    (BORDER-COLOR :BLUE)
    (CENTER :X-AND-Y)
    (DEFAULT-ANSWER :YES)
```

```
(CONTENTS
  "DO YOU WANT TO CLEAR ALL INPUTS AND RETURN TO THE INITIAL SCREEN?"))


(DEFINE-INSTANCE INPUT-VALUES-2
  (:print-name "INPUT-VALUES-2"
   :doc-string ""
   :is  SCREEN-TEMPLATE)
  (TEXT-COLOR :LIGHT-GRAY)
  (BORDER-COLOR :LIGHT-GRAY)
  (BORDER :SINGLE)
  (ORIENTATION :ROW)
  (LAYOUT :LINEAR)
  (SPACES-BETWEEN-COLUMNS 0)
  (SPACES-BETWEEN-ROWS 0)
  (OBJECTS
    ( (VEG-AGE :SLOT INPUT-DATA VEG-AGE-VAL :WIDTH 1)
      (NATURAL :SLOT INPUT-DATA NATURAL-FEATURE-VAL :WIDTH 1)
      (WETLAND :SLOT INPUT-DATA WETLAND-VAL :WIDTH 1)
      (PLANT-SPECIES :SLOT INPUT-DATA ENDANGERED- PLANT-VAL :WIDTH 1)
      (WILD-SPECIES :SLOT INPUT-DATA ENDANGERED-WILD- VAL :WIDTH 1)))
  (CONTENTS
    ( ("VEG AGE            " VEG-AGE)
      ("")
      ("NATURAL FEATURE     " NATURAL)
      ("")
      ("WETLAND            " WETLAND)
      ("")
```

```
        ("ENDANGERED PLANTS    " PLANT-SPECIES)
        ("")
        ("ENDANGERED WILD LIFE " WILD-SPECIES))))



(DEFINE-INSTANCE WELCOME-FORM
  (:print-name "WELCOME-FORM"
   :doc-string ""
   :is  SCREEN-TEMPLATE)
  (OBJECTS ((GO-BUTTON :MENU GO-MENU "*" :COLOR :WHITE)
         (DATE-STRING :SLOT THE-PLAN DATE-STRING :COLOR :BLUE )))
  (CONTENTS
      ( ("       SITE PLANNING ADVISOR           ")
        ("") ("")
        ("  THIS SYSTEM WILL HELP YOU IN THE SITE PLANNING PROCESS. ")
        ("  THE PRESENT STATE OF APPLICATION IS RESTRICTED TO THE  ")
        ("  DESIGN OF THE PARKING LOT FOR A COMMERCIAL BUILDING.  ")
        ("  SYSTEM WILL ASK YOU SITE SPECIFIC DATA IN THE FOLLOWING ")
        ("  SCREENS. MAKE SURE YOU ANSWER ALL THE QUESTIONS ASKED. ")
        ("  REFER HANDOUT FOR SPECIFIC INFORMATION OF CODES USED. ")
        ("")
        ("  SELECT HERE TO START => " GO-BUTTON " <=")
        ("")
        ("          TODAY'S DATE: " (:NO- SELECT DATE-STRING))))
  (TEXT-COLOR :LIGHT-GRAY)
  (BORDER-COLOR :LIGHT-GRAY)
  (BORDER :DOUBLE)
  (ORIENTATION :ROW)
```

```
(LAYOUT :LINEAR)
(SPACES-BETWEEN-COLUMNS 0)
(SPACES-BETWEEN-ROWS 0))


(DEFINE-INSTANCE USER-OUTPUT
  (:print-name "USER-OUTPUT"
   :doc-string "Informative and explanatory messages to the user are displayed via this
window"
   :is  OUTPUT-WINDOW)
  (LEFT 0)
  (TOP 22)
  (WIDTH 80)
  (HEIGHT 3)
  (FOREGROUND-COLOR :RED)
  (BACKGROUND-COLOR :WHITE)
  (AUTO-NEWLINE :YES)
  (SCROLLING :SCROLL))


(DEFINE-INSTANCE OUTPUT-STANDARDS
  (:print-name  "OUTPUT-STANDARDS"
   :doc-string  "informative windows on different subjects"
   :is OUTPUT-WINDOW)
  (LEFT 0)
  (TOP 0)
  (WIDTH 80)
  (HEIGHT 22)
```

```
(AUTO-NEWLINE :YES)
(CLEAR-BEFORE-NEW-DISPLAY :YES))

(DEFINE-INSTANCE INFERENCE-CONFIRM
  (:print-name "INFERENCE-CONFIRM"
   :doc-string "to confirm begining of the inference engine"
   :is POPUP-CONFIRM)
  (TARGET-SLOT CONFIRMATION)
  (TARGET-INSTANCE INPUT-DATA)
  (INSTRUCTIONS " TO START THE INFERENCE ENGINE " )
  (BORDER-COLOR :MAGENTA)
  (TOP 18)
  (LEFT 2)
  (DEFAULT-ANSWER :YES)
  (CONTENTS
" * ARE ALL THE VALUES YOU HAVE INPUT IN THE SYSTEM CORRECT ?  * "))
```

***RULES-***

```
(DEFINE-RULE ASK-INFERENCE-CONFIRM
  (:print-name "ASK-INFERENCE-CONFIRM"
   :certainty 1.0
   :doc-string "to ask user whether all the values in system are correct so that inference
engine could be started"
   :dependency nil
   :direction  :forward
   :explanation-string ""
```

```
  :priority 300
  :sponsor TOP-SPONSOR)
 (INSTANCE THE-PLAN IS PLANNING
    WITH STATUS :IN-PROGRESS)
 (INSTANCE START-PLAN IS SCREEN-CONTROL
    WITH STATUS :RUNNING
    WITH CURRENT-SCREEN WORK-SCREEN)
 (INSTANCE INPUT-DATA IS INPUT-VALUES
    WITH-UNKNOWN CONFIRMATION ?)
THEN
 (INSTANCE INFERENCE-CONFIRM IS POPUP-CONFIRM
    WITH GO :YES) )


(DEFINE-RULE INFERENCE-CONFIRM-NOT
 (:print-name "INFERENCE-CONFIRM-NOT"
  :certainty 1.0
  :doc-string ""
  :dependency nil
  :direction  :forward
  :explanation-string ""
  :priority 290
  :sponsor TOP-SPONSOR)
 (INSTANCE INFERENCE-CONFIRM IS POPUP-CONFIRM
    WITH ANSWER :NO)
THEN
 (INSTANCE START-PLAN IS SCREEN-CONTROL
    WITH STATUS :INITIAL-STATUS)
```

```
  (INSTANCE THE-PLAN IS PLANNING
      WITH CLEAR :YES))


(DEFINE-RULE ASK-PROJECT-DATA
  (:print-name "ASK-PROJECT-DATA"
   :certainty 1.0
   :doc-string "first screen in automation asking project data"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 550
   :sponsor TOP-SPONSOR)
  (INSTANCE THE-PLAN IS PLANNING WITH STATUS :START)
  (INSTANCE START-PLAN IS SCREEN-CONTROL
      WITH STATUS :RUNNING
      WITH CURRENT-SCREEN WORK-SCREEN)
  (OR (INSTANCE THE-PROJECT IS PROJECT
         WITH-UNKNOWN PROJECT-NAME ?)
     (INSTANCE THE-PROJECT IS PROJECT
         WITH-UNKNOWN PROJECT-NUMBER ?)
     (INSTANCE THE-PROJECT IS PROJECT
         WITH-UNKNOWN DESIGNER-NAME ?))
 THEN
  (INSTANCE THE-PLAN IS PLANNING
      WITH STATUS :IN-PROGRESS)
  (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
      WITH DISPLAY " ** SELECT OK WHEN DONE ** ")
```

```
(INSTANCE PROJECT-DATA IS SET-SLOT-VALUES
    WITH GO :YES))


(DEFINE-RULE ASK-DATA-1
  (:print-name "ASK-DATA-1"
   :certainty 1.0
   :doc-string "second screen in automation asking first set of questions"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 525
   :sponsor TOP-SPONSOR)
  (INSTANCE THE-PLAN IS PLANNING
     WITH STATUS :IN-PROGRESS)
  (INSTANCE START-PLAN IS SCREEN-CONTROL
     WITH STATUS :RUNNING
     WITH CURRENT-SCREEN WORK-SCREEN)
  (OR (INSTANCE INPUT-DATA IS INPUT-VALUES
        WITH-UNKNOWN SLOPE-VAL ?)
     (INSTANCE INPUT-DATA IS INPUT-VALUES
        WITH-UNKNOWN SUB-SOIL-VAL ?)
     (INSTANCE INPUT-DATA IS INPUT-VALUES
        WITH-UNKNOWN DRAINAGE-CAP-VAL ?))
 THEN
  (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
     WITH DISPLAY " ** SELECT OK WHEN DONE ** ")
  (INSTANCE INPUT-DATA-1 IS SET-SLOT-VALUES
```

```
    WITH GO :YES))

(DEFINE-RULE ASK-DATA-2
  (:print-name "ASK-DATA-2"
   :certainty 1.0
   :doc-string "third screen in automation asking second set of questions"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 500
   :sponsor TOP-SPONSOR)
  (INSTANCE THE-PLAN IS PLANNING
      WITH STATUS :IN-PROGRESS)
  (INSTANCE START-PLAN IS SCREEN-CONTROL
      WITH STATUS :RUNNING
      WITH CURRENT-SCREEN WORK-SCREEN)
  (OR (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH-UNKNOWN WATER-TABLE-VAL ?)
      (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH-UNKNOWN VEG-COND-VAL ?)
      (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH-UNKNOWN VEG-AGE-VAL ?))
 THEN
   (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
      WITH DISPLAY " ** SELECT OK WHEN DONE ** ")
   (INSTANCE INPUT-DATA-2 IS SET-SLOT-VALUES
      WITH GO :YES))
```

```
(DEFINE-RULE ASK-DATA-3
  (:print-name "ASK-DATA-3"
   :certainty 1.0
   :doc-string "fourth screen in automation asking third set of questions"
   :dependency nil
   :direction :forward
   :explanation-string ""
   :priority 475
   :sponsor TOP-SPONSOR)
  (INSTANCE THE-PLAN IS PLANNING
      WITH STATUS :IN-PROGRESS)
  (INSTANCE START-PLAN IS SCREEN-CONTROL
      WITH STATUS :RUNNING
      WITH CURRENT-SCREEN WORK-SCREEN)
  (OR (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH-UNKNOWN NATURAL-FEATURE-VAL ?)
      (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH-UNKNOWN WETLAND-VAL ?)
      (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH-UNKNOWN ENDANGERED-PLANT-VAL ?)
      (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH-UNKNOWN ENDANGERED-WILD-VAL ?))
 THEN
   (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
       WITH DISPLAY " ** SELECT OK WHEN DONE ** ")
   (INSTANCE INPUT-DATA-3 IS SET-SLOT-VALUES
```

```
        WITH GO :YES))


(DEFINE-RULE ASK-INFORMATION
  (:print-name "ASK-INFORMATION"
   :certainty 1.0
   :doc-string "fifth scrren in automation asking about information screens"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 450
   :sponsor TOP-SPONSOR)
  (INSTANCE THE-PLAN IS PLANNING
      WITH STATUS :IN-PROGRESS)
  (INSTANCE START-PLAN IS SCREEN-CONTROL
      WITH STATUS :RUNNING
      WITH CURRENT-SCREEN WORK-SCREEN)
 THEN
   (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
       WITH DISPLAY " ** SELECT OK WHEN DONE ** ")
   (INSTANCE INFORMATION IS SET-SLOT-VALUES
       WITH GO :YES))


(DEFINE-RULE START-WITH-WELCOME-SCREEN
  (:print-name "START-WITH-WELCOME-SCREEN"
    :doc-string "this causes welcome-screen to be displayed when the application is
started."
```

```
    :dependency NIL
    :direction  :FORWARD
    :certainty  1.0
    :explanation-string ""
    :priority 600
    :sponsor TOP-SPONSOR)
   (INSTANCE START-PLAN IS SCREEN-CONTROL
      WITH STATUS :STARTED)
  THEN
   (INSTANCE THE-PLAN IS PLANNING
       WITH STATUS :NO)
   (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
       WITH CLEAR :YES)
   (INSTANCE START-PLAN IS SCREEN-CONTROL
       WITH STATUS :RUNNING
       WITH NEW-SCREEN WELCOME-SCREEN))


(DEFINE-RULE CHANGE-STATUS-OF-STARTPLAN
   (:print-name "CHANGE-STATUS-OF-STARTPLAN"
    :doc-string ""
    :dependency NIL
    :direction  :FORWARD
    :certainty  1.0
    :explanation-string ""
    :priority 0
    :sponsor TOP-SPONSOR)
   (INSTANCE START-PLAN IS SCREEN-CONTROL
```

```
      WITH STATUS :INITIAL-STATUS)
  THEN
    (EVALUATE (START-SCREEN (QUOTE START-PLAN))))


(DEFINE-RULE START-THE-CONSULTATION-SESSION
   (:print-name "START-THE-CONSULTATION-SESSION"
    :doc-string ""
    :dependency NIL
    :direction  :FORWARD
    :certainty  1.0
    :explanation-string ""
    :priority 800
    :sponsor TOP-SPONSOR)
   (INSTANCE THE-PLAN IS PLANNING
      WITH STATUS :START)
   (INSTANCE START-PLAN IS SCREEN-CONTROL
      WITH STATUS :RUNNING
      WITH CURRENT-SCREEN WELCOME-SCREEN)
  THEN
    (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
       WITH DISPLAY "STARTING THE CONSULTATION SESSION ")
    (INSTANCE START-PLAN IS SCREEN-CONTROL
       WITH GET-USER-INPUT-WHEN-NEW-SCREEN :NO
       WITH NEW-SCREEN WORK-SCREEN))


(DEFINE-RULE INITIALIZE-DATE-STRING
```

```
(:print-name "INITIALIZE-DATE-STRING"
 :certainty 1.0
 :doc-string ""
 :dependency nil
 :direction  :forward
 :explanation-string ""
 :priority 0
 :sponsor TOP-SPONSOR)
(INSTANCE THE-PLAN IS PLANNING
    WITH MONTH-NUMBER ?MONTH
    WITH DAY-NUMBER ?DAY
    WITH YEAR ?YEAR)
THEN
 (INSTANCE THE-PLAN IS PLANNING
     WITH DATE-STRING
 (EVALUATE (FORMAT NIL "¬S/¬S/¬S" ?MONTH ?DAY ?YEAR))))


(DEFINE-RULE INITIALIZE-DATE
 (:print-name "INITIALIZE-DATE"
  :certainty 1.0
  :doc-string "GET TODAY'S DATE FROM THE SYSTEM CALENDER"
  :dependency nil
  :direction  :forward
  :explanation-string ""
  :priority 700
  :sponsor TOP-SPONSOR)
 (INSTANCE START-PLAN IS SCREEN-CONTROL
```

```
      WITH STATUS :STARTED)
  THEN
    (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
       WITH DISPLAY "GETTING DATE FROM SYSTEM.")
    (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
       WITH DISPLAY
        (EVALUATE (PROGN (UPDATE-DATE)
           (FORMAT NIL "DATE IS ¬S/¬S/¬S"
            *START-PLAN-MONTH*
            *START-PLAN-DAY*
            *START-PLAN-YEAR*))))
    (INSTANCE THE-PLAN IS PLANNING
       WITH MONTH-NUMBER  (EVALUATE *START-PLAN-MONTH*)
       WITH DAY-NUMBER (EVALUATE *START-PLAN-DAY*)
       WITH YEAR (EVALUATE *START-PLAN-YEAR*)))


(DEFINE-RULE CLEAR-THE-SESSION
  (:print-name "CLEAR-THE SESSION"
   :certainty 1.0
   :doc-string "Clears all the assertions and slot values to make the system ready for re-
run."
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 1000
   :sponsor TOP-SPONSOR)
  (INSTANCE THE-PLAN IS PLANNING
```

```
        WITH CLEAR :YES)
 THEN
    (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
        WITH DISPLAY
"RETRACTING ALL VALUES GIVEN TO THE SYSTEM IN PREVIOUS SESSION.")
    (EVALUATE
        (RETRACT-ALL-SLOTS (QUOTE INPUT-DATA)))
    (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
        WITH DISPLAY
"RECTRACTING ALL VALUES OF PROJECT AND DESIGNER IDENTIFICATION.")
    (EVALUATE
        (RETRACT-ALL-SLOTS (QUOTE THE-PROJECT)))
    (INSTANCE THE-PLAN IS PLANNING
        WITH CLEAR :NO))


(DEFINE-RULE BACK-TO-WELCOME-SCREEN-AND-CLEAR
  (:print-name "BACK-TO-WELCOME-SCREEN-AND-CLEAR"
   :certainty 1.0
   :doc-string "End of the session."
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 1000
   :sponsor TOP-SPONSOR)
  (INSTANCE THE-PLAN IS PLANNING
      WITH END-SESSION :YES)
 THEN
```

```
    (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
        WITH DISPLAY
"END OF SESSION. GO BACK TO THE WELCOME-SCREEN AND CLEAR THE SESSION.")
    (INSTANCE START-PLAN IS SCREEN-CONTROL
        WITH NEW-SCREEN WELCOME-SCREEN)
    (INSTANCE THE-PLAN IS PLANNING
        WITH END-SESSION :NO
        WITH STATUS :NO
        WITH CLEAR :YES))


(DEFINE-RULE CLEAR-THEN-START-THE-SESSION
   (:print-name "CLEAR-THEN-START-THE-SESSION"
    :certainty 1.0
    :doc-string ""
    :dependency nil
    :direction  :forward
    :explanation-string ""
    :priority 1000
    :sponsor TOP-SPONSOR)
   (INSTANCE THE-PLAN IS PLANNING
       WITH STATUS :CLEAR-THEN-START)
 THEN
    (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
        WITH DISPLAY
"MAKING SYSTEM READY TO ACCEPT VALUES AND CLEAR.")
    (INSTANCE THE-PLAN IS PLANNING
        WITH STATUS :START
```

```
        WITH CLEAR :YES))


(DEFINE-RULE PRINT-OUT-OUTPUT
  (;  :print-name "PRINT-OUT-OUTPUT"
:certainty 1.0
   :doc-string ""
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine) -> ?x
  (OR (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH CONFIRMATION :YES)
     (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH DIRECT-RUN :YES))
 THEN
   (evaluate (send *terminal-io* :clear-screen))
   (instance user-output is output-window
        with display
    (evaluate (final-print recom-list)))
    (evaluate (defparameter recom-list ()))
    (evaluate (defparameter count-list-elm 0))
    (evaluate (defparameter counter-r 0))
   (evaluate (read-line)
   (send *terminal-io* :clear-screen))
   (instance user-output is output-window
```

```
        with display
"PRESS <RETURN> TO GO BACK TO WORK-SCREEN, MODIFY THE VALUES AND SE-
LECT <RUN> TO RUN THE SYSTEM AGAIN OR SELECT <END SESSION> TO RESTART
THE PROGRAM.")
   (retract ?x)
   (evaluate (slot-retract-value 'input-data 'direct-run))
   (evaluate (slot-retract-value 'inference-confirm 'answer))
 AND-THEN
   (EVALUATE (READ-LINE)
   (SEND *TERMINAL-IO* :CLEAR-SCREEN))
   (INSTANCE START-PLAN IS SCREEN-CONTROL
       WITH NEW-SCREEN WORK-SCREEN))



(DEFINE-RULE intermediate-assertion
  (:print-name "intermediate assertion "
   :certainty 1.0
   :doc-string "to make the assertion to fire all the inference rules"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 290
   :sponsor TOP-SPONSOR)
  (instance inference-confirm is popup-confirm with answer :yes)
 THEN
   (start-inference-engine))
```

```
(DEFINE-RULE SECOND-INTERMEDIATE-ASSERTION
  (:print-name "SECOND-INTERMEDIATE-ASSERTION"
   :certainty 1.0
   :doc-string "to assert start-inference-engine to start from work-screen"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 250
   :sponsor TOP-SPONSOR)
  (instance input-data is input-values
      with direct-run :yes)
 THEN
   (start-inference-engine))



(DEFINE-RULE DISPLAY-PROJECT-INFO
  (:print-name "DISPLAY-PROJECT-INFO"
   :certainty 1.0
   :doc-string "to display project information before recommendations"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 100
   :sponsor TOP-SPONSOR)
  (INSTANCE THE-PROJECT IS PROJECT
      WITH PROJECT-NAME ?NAME)
  (INSTANCE THE-PROJECT IS PROJECT
      WITH PROJECT-NUMBER ?NUMBER)
```

```
  (INSTANCE THE-PROJECT IS PROJECT
      WITH DESIGNER-NAME ?DESIGNER)
 THEN
   (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
       WITH CLEAR :YES)
   (INSTANCE USER-OUTPUT IS OUTPUT-WINDOW
       WITH DISPLAY
(" RECOMMENDATIONS FOR PROJECT " ?NAME :RETURN
 " DESIGNER-NAME " ?DESIGNER :RETURN
 " PRESS <PRINT SCREEN> TO PRINT & THEN HIT <RETURN> TO CONTINUE ")))


(DEFINE-RULE STORM-WATER-INFORMATION
  (:print-name "STORM-WATER-INFORMATION"
   :certainty 1.0
   :doc-string "to display screen on storm-water management tips"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH STORM-WATER-INFO YES)
 THEN
   (INSTANCE OUTPUT-STANDARDS IS OUTPUT-WINDOW
       WITH CLEAR-BEFORE-NEW-DISPLAY :YES)
   (INSTANCE OUTPUT-STANDARDS IS OUTPUT-WINDOW
       WITH DISPLAY
```

```
(" STORM-WATER-MANAGEMENT"
 :RETURN
 :RETURN
 " IN CASE OF THE PROJECT WITH THIS MAGNITUDE,  "
 :RETURN
 " DESIGN FOR 10 YEARS EVENT FOR STORM WATER MANAGEMENT."
 :RETURN
 " WHERE EVER POSSIBLE A TEMPORARY CATCHMENT OF 6 TO 9 INCHES "
 :RETURN
 " CAN BE PROVIDED ON THE SITE SO THAT STORM WATER COULD BE "
 :RETURN
 " RELEASED AT REDUCED RATE. PART OF THE PARKING SPACE CAN BE "
 :RETURN
 " USED EFFECTIVELY FOR SUCH TEMPORARY CATCHMENT."
 :RETURN
 "                              PRESS <ENTER> TO CONTINUE   "))
AND-THEN
  (EVALUATE (READ-LINE)
    (SEND *TERMINAL-IO* :CLEAR-SCREEN)))


(DEFINE-RULE PAVING-INFORMATION
  (:print-name "PAVING-INFORMATION"
   :certainty 1.0
   :doc-string "to display screen on paving and slopes information"
   :dependency nil
   :direction  :forward
   :explanation-string ""
```

```
     :priority 0
     :sponsor TOP-SPONSOR)
    (INSTANCE INPUT-DATA IS INPUT-VALUES
       WITH PAVING-INFO YES)
  THEN
    (INSTANCE OUTPUT-STANDARDS IS OUTPUT-WINDOW
       WITH CLEAR-BEFORE-NEW-DISPLAY :YES)
    (INSTANCE OUTPUT-STANDARDS IS OUTPUT-WINDOW
       WITH DISPLAY
     ("  PAVING-INFORMATION"
      :RETURN
      :RETURN
      "  PAVING WITH HARD REFLECTIVE SURFACES SHOULD BE EITHER  "
      :RETURN
    "  REDUCED AS MUCH AS POSSIBLE ON THE WEST SIDE OF THE STRUCTURE "
      :RETURN
    "  OR THE PAVING SHOULD BE SHADED TO REDUCE EXCESS HEAT"
      :RETURN
    "  AND REFLECTIVITY. "
      :RETURN
    "  PERMISSIBLE SLOPES"
      :RETURN
    "  MINIMUM GRADE FOR DRAINAGE "
      :RETURN
    "    PLANTED OR BROAD PAVED AREAS  --- 1% "
      :RETURN
    "    TEMPORARY PONDING IF PERMISSIBLE --- 0.5% "
      :RETURN
```

```
"   BUILDING PERIMETER  — 2% min. "
  :RETURN
"   DRAINAGE SWALES AND DITCHES  — 10% max."
  :RETURN
"                               — 2% min. "
  :RETURN
"                       PRESS <ENTER> TO CONTINUE "))
AND-THEN
  (EVALUATE (READ-LINE)
    (SEND *TERMINAL-IO* :CLEAR-SCREEN)))



(DEFINE-RULE ILLUMINATION-STANDARDS-INFORMATION
  (:print-name "ILLUMINATION-STANDARDS-INFORMATION"
   :certainty 1.0
   :doc-string ""
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH ILLUMINATION-STD-INFO YES)
 THEN
   (INSTANCE OUTPUT-STANDARDS IS OUTPUT-WINDOW
      WITH CLEAR-BEFORE-NEW-DISPLAY :YES)
   (INSTANCE OUTPUT-STANDARDS IS OUTPUT-WINDOW
      WITH DISPLAY
```

```
(" ILLUMINATION STANDARD "
 :RETURN
 " SELF PARKING - 1.0 footcandles LEVEL OF ILLUMINATION "
 :RETURN
 " AVERAGE MOUNTING HEIGHT - 30' (20'-50' range) "
 :RETURN
 " LUMINAIRE SPACING "
 :RETURN
 " RATIO OF LUMINAIRE SPACING TO THE MOUNTING HEIGHT SHOULD BE "
 :RETURN
 " KEPT TO A MINIMUM. (4:1 is preferred) "
 :RETURN
 " SPECIAL CARE SHOULD BE TAKEN TO AVOID OBJECTIONABLE SPILL"
 :RETURN
 " LIGHT AND GLARE ON ADJESCENT PROPERTY. "
 :RETURN
 :RETURN
 "                                PRESS <RETURN> TO CONTINUE "))
AND-THEN
 (EVALUATE (READ-LINE)
   (SEND *TERMINAL-IO* :CLEAR-SCREEN)))


(DEFINE-RULE SPATIAL-STANDARDS-INFORMATION
  (:print-name "SPATIAL-STANDARDS-INFORMATION"
   :certainty 1.0
   :doc-string ""
   :dependency nil
```

```
      :direction  :forward
      :explanation-string ""
      :priority 0
      :sponsor TOP-SPONSOR)
     (INSTANCE INPUT-DATA IS INPUT-VALUES
        WITH SPATIAL-INFO YES)
   THEN
      (INSTANCE OUTPUT-STANDARDS IS OUTPUT-WINDOW
         WITH CLEAR-BEFORE-NEW-DISPLAY :YES)
      (INSTANCE OUTPUT-STANDARDS IS OUTPUT-WINDOW
         WITH DISPLAY
      (" SPATIAL STANDARDS FO COMMERCIAL BUILDING SITE DESIGN "
       :RETURN
       :RETURN
      " DIMENTION OF STANDARD PARKING BAY - 9' X 10' "
       :RETURN
      " EMPLOYEES ONLY - 1/10th OF TOTAL - 8' X 10'  "
       :RETURN
      " SMALL CARS - 1/3rd OF TOTAL - 8.5' X 10' "
       :RETURN
      " HANDICAPPED PARKING - 3% OF TOTAL - 12' X 10'"
       :RETURN
      " ALLOWABLE NUMBER- 5.5 SPACES / 1000 SFT OF GROSS LEASABLE AREA "
       :RETURN
      " LOADING/UNLOADING - "
      "        CLASS A - 14' X 55' X 15'vert (MINIMUM 1) "
       :RETURN
      "        CLASS B - 12' X 30' X 15'vert (ADEQUATE) "
```

```
     :RETURN
     :RETURN
     "                         PRESS <RETURN> TO CONTINUE "))
  AND-THEN
    (EVALUATE (READ-LINE)
       (SEND *TERMINAL-IO* :CLEAR-SCREEN)))
```

***INFERENCING RULES-***

```
(DEFINE-RULE  SLOPE-VERY-FLAT
   (:print-name "SLOPE-VERY-FLAT"
    :certainty 1.0
    :doc-string "if the slope is less than .5%, then reconsider the site for its economic im-
pact for devp. cost"
    :dependency nil
    :direction  :forward
    :explanation-string ""
    :priority 0
    :sponsor TOP-SPONSOR)
   (start-inference-engine)
   (INSTANCE INPUT-DATA IS INPUT-VALUES
       WITH SLOPE-VAL VERY-FLAT)
 THEN
   (EVALUATE
    (RECON-LINE :WRITE-LINE
". RECONSIDER THIS PART OF THE SITE FOR ITS ECONOMIC IMPACT FOR DEVELOP-
MENT COST AS THE SLOPE IS LESS THAN .5% HERE")))
```

```
(DEFINE-RULE  SLOPE-FLAT1
  (:print-name "SLOPE-FLAT1"
   :certainty 1.0
   :doc-string "if the slope is 1%-3% then site could be developed without intention of cut
and fill "
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH SLOPE-VAL FLAT1)
 THEN
   (EVALUATE
     (RECON-LINE :WRITE-LINE
". THIS PART OF THE SITE COULD BE DEVELOPED WITHOUT INTENTION OF CUT AND
FILL AS THE SLOPE IS 1%-3% HERE ")))


(DEFINE-RULE  SLOPE-FLAT2
  (:print-name "SLOPE-FLAT2"
   :certainty 1.0
   :doc-string "if slope is between 3%-8% site may need to be developed with terracing"
   :dependency nil
   :direction  :forward
   :explanation-string ""
```

```
    :priority 0
    :sponsor TOP-SPONSOR)
   (start-inference-engine)
   (INSTANCE INPUT-DATA IS INPUT-VALUES
       WITH SLOPE-VAL FLAT2)
 THEN
    (EVALUATE
       (RECON-LINE :WRITE-LINE
". THIS PART OF THE SITE MAY NEED TO BE DEVELOPED WITH TERRACING AS THE
SLOPE IS BETWEEN 3%-8% HERE")))



(DEFINE-RULE  SLOPE-ROLLING
   (:print-name "SLOPE-ROLLING"
    :certainty 1.0
    :doc-string "if the slope is less than .5%, then reconsider the site for its economic im-
pact for devp. cost"
    :dependency nil
    :direction  :forward
    :explanation-string ""
    :priority 0
    :sponsor TOP-SPONSOR)
   (start-inference-engine)
   (INSTANCE INPUT-DATA IS INPUT-VALUES
       WITH SLOPE-VAL ROLLING)
 THEN
    (EVALUATE
       (RECON-LINE :WRITE-LINE
```

```
". RECONSIDER THIS PART OF THE SITE FOR ITS ECONOMIC IMPACT FOR DEVELOP-
MENT COST AS THE SLOPE IS MORE THAN 8% HERE")))


(DEFINE-RULE  SLOPE-PAVEMENT-1
  (:print-name "SLOPE-PAVEMENT-1"
   :certainty 1.0
   :doc-string "if slope is more than .5%, then asphalt pavement can be used in this area"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
  (OR (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH SLOPE-VAL FLAT1)
     (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH SLOPE-VAL FLAT2)
     (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH SLOPE-VAL ROLLING))
 THEN
   (EVALUATE
     (RECON-LINE :WRITE-LINE
". IF YOU DECIDE TO DEVELOPE HERE, YOU WILL BE ABLE TO USE ASPHALT AS A
PAVEMENT MATERIAL")))


(DEFINE-RULE  SLOPE-PAVEMENT-2
```

```
 (:print-name "SLOPE-PAVEMENT-2"
  :certainty 1.0
  :doc-string "if the slope is less than .5%, then portland cement is recommended as a
pavement material"
:dependency nil
  :direction  :forward
  :explanation-string ""
  :priority 0
  :sponsor TOP-SPONSOR)
 (start-inference-engine)
 (INSTANCE INPUT-DATA IS INPUT-VALUES
     WITH SLOPE-VAL VERY-FLAT)
THEN
  (EVALUATE
    (RECON-LINE :WRITE-LINE
". IF YOU DECIDE TO DEVELOPE HERE, PORTLAND CEMENT IS RECOMMENDED AS A
PAVEMENT MATERIAL AS THE SLOPE IS LESS THAN .5%")))


(DEFINE-RULE  DRAINAGE-TECH-1
 (:print-name "DRAINAGE-TECH-1"
  :certainty 1.0
  :doc-string "storm water drainage management technique recommendation no. 1"
  :dependency nil
  :direction  :forward
  :explanation-string ""
  :priority 0
  :sponsor TOP-SPONSOR)
```

```
(start-inference-engine)
(INSTANCE INPUT-DATA IS INPUT-VALUES
    WITH DRAINAGE-CAP-VAL WELL)
(INSTANCE INPUT-DATA IS INPUT-VALUES
    WITH WATER-TABLE-VAL LOW)
(OR (INSTANCE INPUT-DATA IS INPUT-VALUES
        WITH SLOPE-VAL VERY-FLAT)
    (INSTANCE INPUT-DATA IS INPUT-VALUES
        WITH SLOPE-VAL FLAT1)
    (INSTANCE INPUT-DATA IS INPUT-VALUES
        WITH SLOPE-VAL FLAT2))
THEN
 (EVALUATE
    (RECON-LINE :WRITE-LINE
". IN THIS AREA 1-10 TYPE OF STORM WATER DRAINAGE TECHNIQUES CAN BE USED.
PLEASE REFER TO THE CHART FOR CORROSPONDING VALUES")))



(DEFINE-RULE  DRAINAGE-TECH-3
  (:print-name "DRAINAGE-TECH-3"
   :certainty 1.0
   :doc-string "storm water drainage management technique recommendation no. 3"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
```

```
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH DRAINAGE-CAP-VAL WELL)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH WATER-TABLE-VAL HIGH)
  (OR (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH SLOPE-VAL VERY-FLAT)
      (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH SLOPE-VAL FLAT1)
      (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH SLOPE-VAL FLAT2))
 THEN
   (EVALUATE
      (RECON-LINE :WRITE-LINE
". IN THIS AREA 2,8-10 TYPE OF STORM WATER DRAINAGE TECHNIQUES CAN BE USED.
PLEASE REFER TO THE CHART FOR CORROSPONDING VALUES")))



(DEFINE-RULE  DRAINAGE-TECH-5
  (:print-name "DRAINAGE-TECH-5"
   :certainty 1.0
   :doc-string "storm water drainage management technique recommendation no. 5"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
```

```
      WITH DRAINAGE-CAP-VAL MODERATE)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH WATER-TABLE-VAL LOW)
  (OR (INSTANCE INPUT-DATA IS INPUT-VALUES
         WITH SLOPE-VAL VERY-FLAT)
     (INSTANCE INPUT-DATA IS INPUT-VALUES
         WITH SLOPE-VAL FLAT1)
     (INSTANCE INPUT-DATA IS INPUT-VALUES
         WITH SLOPE-VAL FLAT2))
 THEN
   (EVALUATE
      (RECON-LINE :WRITE-LINE
". IN THIS AREA 2-4,6-10 TYPE OF STORM WATER DRAINAGE TECHNIQUES CAN BE
USED. PLEASE REFER TO THE CHART FOR CORROSPONDING VALUES")))


(DEFINE-RULE  DRAINAGE-TECH-7
  (:print-name "DRAINAGE-TECH-7"
   :certainty 1.0
   :doc-string "storm water drainage management technique recommendation no. 7"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH DRAINAGE-CAP-VAL MODERATE)
```

```
(INSTANCE INPUT-DATA IS INPUT-VALUES
    WITH WATER-TABLE-VAL HIGH)
(OR (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH SLOPE-VAL VERY-FLAT)
   (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH SLOPE-VAL FLAT1)
   (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH SLOPE-VAL FLAT2))
THEN
 (EVALUATE
    (RECON-LINE :WRITE-LINE
". IN THIS AREA 2,8-10 TYPE OF STORM WATER DRAINAGE TECHNIQUES CAN BE USED.
PLEASE REFER TO THE CHART FOR CORROSPONDING VALUES")))


(DEFINE-RULE  DRAINAGE-TECH-2
  (:print-name "DRAINAGE-TECH-2"
   :certainty 1.0
   :doc-string "storm water drainage management technique recommendation no. 2"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
     WITH DRAINAGE-CAP-VAL WELL)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
```

```
        WITH WATER-TABLE-VAL LOW)
    (INSTANCE INPUT-DATA IS INPUT-VALUES
        WITH SLOPE-VAL ROLLING)
  THEN
    (EVALUATE
        (RECON-LINE :WRITE-LINE
". IN THIS AREA 1,4-10 TYPE OF STORM WATER DRAINAGE TECHNIQUES CAN BE USED.
PLEASE REFER TO THE CHART FOR CORROSPONDING VALUES")))


(DEFINE-RULE  DRAINAGE-TECH-4
    (:print-name "DRAINAGE-TECH-4"
     :certainty 1.0
     :doc-string "storm water drainage management technique recommendation no. 4"
     :dependency nil
     :direction  :forward
     :explanation-string ""
     :priority 0
     :sponsor TOP-SPONSOR)
    (start-inference-engine)
    (INSTANCE INPUT-DATA IS INPUT-VALUES
        WITH DRAINAGE-CAP-VAL WELL)
    (INSTANCE INPUT-DATA IS INPUT-VALUES
        WITH WATER-TABLE-VAL HIGH)
    (INSTANCE INPUT-DATA IS INPUT-VALUES
        WITH SLOPE-VAL ROLLING)
  THEN
    (EVALUATE
```

```
    (RECON-LINE :WRITE-LINE
". IN THIS AREA 8-10 TYPE OF STORM WATER DRAINAGE TECHNIQUES CAN BE USED.
PLEASE REFER TO THE CHART FOR CORROSPONDING VALUES")))


(DEFINE-RULE  DRAINAGE-TECH-6
  (:print-name "DRAINAGE-TECH-6"
   :certainty 1.0
   :doc-string "storm water drainage management technique recommendation no. 6"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH DRAINAGE-CAP-VAL MODERATE)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH WATER-TABLE-VAL LOW)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH SLOPE-VAL ROLLING)
 THEN
   (EVALUATE
    (RECON-LINE :WRITE-LINE
". IN THIS AREA 4,6-10 TYPE OF STORM WATER DRAINAGE TECHNIQUES CAN BE USED.
PLEASE REFER TO THE CHART FOR CORROSPONDING VALUES")))
```

```
(DEFINE-RULE  DRAINAGE-TECH-8
  (:print-name "DRAINAGE-TECH-8"
   :certainty 1.0
   :doc-string "storm water drainage management technique recommendation no. 8"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH DRAINAGE-CAP-VAL MODERATE)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH WATER-TABLE-VAL HIGH)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH SLOPE-VAL ROLLING)
 THEN
   (EVALUATE
     (RECON-LINE :WRITE-LINE
". IN THIS AREA 8-10 TYPE OF STORM WATER DRAINAGE TECHNIQUES CAN BE USED.
PLEASE REFER TO THE CHART FOR CORROSPONDING VALUES")))


(DEFINE-RULE  DRAINAGE-TECH-9
  (:print-name "DRAINAGE-TECH-2"
   :certainty 1.0
   :doc-string "storm water drainage management technique recommendation no. 2"
   :dependency nil
```

```
  :direction  :forward
  :explanation-string ""
  :priority 0
  :sponsor TOP-SPONSOR)
 (start-inference-engine)
 (INSTANCE INPUT-DATA IS INPUT-VALUES
     WITH DRAINAGE-CAP-VAL POOR)
 (INSTANCE INPUT-DATA IS INPUT-VALUES
     WITH WATER-TABLE-VAL LOW)
THEN
  (EVALUATE
    (RECON-LINE :WRITE-LINE
". IN THIS AREA 7-10 TYPE OF STORM WATER DRAINAGE TECHNIQUES CAN BE USED.
PLEASE REFER TO THE CHART FOR CORROSPONDING VALUES")))


(DEFINE-RULE  DRAINAGE-TECH-10
 (:print-name "DRAINAGE-TECH-10"
  :certainty 1.0
  :doc-string "storm water drainage management technique recommendation no. 10"
  :dependency nil
  :direction  :forward
  :explanation-string ""
  :priority 0
  :sponsor TOP-SPONSOR)
 (start-inference-engine)
 (INSTANCE INPUT-DATA IS INPUT-VALUES
     WITH DRAINAGE-CAP-VAL POOR)
```

```
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH WATER-TABLE-VAL HIGH)
 THEN
   (EVALUATE
      (RECON-LINE :WRITE-LINE
". IN THIS AREA 8-10 TYPE OF STORM WATER DRAINAGE TECHNIQUES CAN BE USED.
PLEASE REFER TO THE CHART FOR CORROSPONDING VALUES")))



(DEFINE-RULE  SUB-BASE-TREATMENT-1
  (:print-name "SUB-BASE TREATMENT-1"
   :certainty 1.0
   :doc-string "sub-base treatment in relation to sub- soil shrink-swell-capacity"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH SUB-SOIL-VAL HIGH)
 THEN
   (EVALUATE
      (RECON-LINE :WRITE-LINE
". AS THE SUB-SOIL IS HIGH-SHRINK-SWELL CAPACITY HERE, BRING IN FILL MATERIAL
AND ADD 8-10 in. GRAVEL SUB BASE IN THE GROUND AND THEN ADD 1-2 in. OF AS-
PHALT PAVING FOR STABILITY OF PARKING SURFACE TO BE DEVELOPED ABOVE IT")))
```

```
(DEFINE-RULE  SUB-BASE-TREATMENT-2
  (:print-name "SUB-BASE TREATMENT-2"
   :certainty 1.0
   :doc-string "sub-base treatment in relation to sub- soil shrink-swell-capacity"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
     WITH SUB-SOIL-VAL MODERATE)
 THEN
   (EVALUATE
     (RECON-LINE :WRITE-LINE
". AS THE SUB-SOIL IS MODERATE-SHRINK-SWELL CAPACITY HERE, ADD 8-10 in.
GRAVEL SUB BASE IN THE GROUND AND THEN ADD 1-2 in. OF ASPHALT PAVING FOR
STABILITY OF PARKING SURFACE TO BE DEVELOPED ABOVE IT")))


(DEFINE-RULE  SUB-BASE-TREATMENT-3
  (:print-name "SUB-BASE TREATMENT-3"
   :certainty 1.0
   :doc-string "sub-base treatment in relation to sub- soil shrink-swell-capacity"
   :dependency nil
   :direction  :forward
   :explanation-string ""
```

```
    :priority 0
    :sponsor TOP-SPONSOR)
   (start-inference-engine)
   (INSTANCE INPUT-DATA IS INPUT-VALUES
       WITH SUB-SOIL-VAL LOW)
  THEN
    (EVALUATE
      (RECON-LINE :WRITE-LINE
".  AS THE SUB-SOIL IS LOW-SHRINK-SWELL CAPACITY HERE, ADD 1.5-1.75 in. OF
CRUSHER RUN FOR SUB-BASE AND THEN ADD 1-2 in. OF ASPHALT PAVING FOR STA-
BILITY OF PARKING AREA TO BE DEVELOPED ABOVE IT")))


(DEFINE-RULE  VEGETATION-HEALTHY
   (:print-name "VEGETATION-HEALTHY"
    :certainty 1.0
    :doc-string "criteria for preservation of vegetation no."
    :dependency nil
    :direction  :forward
    :explanation-string ""
    :priority 0
    :sponsor TOP-SPONSOR)
   (start-inference-engine)
   (INSTANCE INPUT-DATA IS INPUT-VALUES
       WITH VEG-COND-VAL HEALTHY)
   (INSTANCE INPUT-DATA IS INPUT-VALUES
       WITH VEG-AGE-VAL > 10YR)
  THEN
```

```
(EVALUATE
    (RECON-LINE :WRITE-LINE
". AS THE EXISTING VEGETATION ON THE SITE IS YOUNG AND HEALTHY, PRESERVA-
TION CRITERIA IS TRY TO PRESERVE AS MANY TREES AS POSSIBLE WHICH ARE MORE
THAN 4 in. IN DIA.")))


(DEFINE-RULE  VEGETATION-DESEASED
  (:print-name "VEGETATION-DESEASED"
   :certainty 1.0
   :doc-string "criteria for preservation of vegetation no. 2"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
  (AND (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH VEG-COND-VAL DESEASED)
     (INSTANCE INPUT-DATA IS INPUT-VALUES
          WITH VEG-AGE-VAL  <10YR))
THEN
  (EVALUATE
    (RECON-LINE :WRITE-LINE
". AS THE EXISTING VEGETATION ON THE SITE IS NOT EITHER YOUNG OR HEALTHY,
NO TREES NEED TO BE PRESERVED ON THE SITE")))
```

```
(DEFINE-RULE  NATURAL-FEATURE
  (:print-name "NATURAL-FEATURE"
   :certainty 1.0
   :doc-string "recommended action on the areas with existing natural feature"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
     WITH NATURAL-FEATURE-VAL STREAM)
 THEN
   (EVALUATE
     (RECON-LINE :WRITE-LINE
". TRY TO PRESERVE THE WOODLAND ALONG THE STREAM TO KEEP THE BANKS AS
NATURAL AS POSSIBLE. THIS WILL HELP IN STORM-WATER- MANAGEMENT TO AVOID
INCREASED RUNOFF SUBSTANTIALLY")))


(DEFINE-RULE  WETLAND-ACTION
  (:print-name "WETLAND-ACTION"
   :certainty 1.0
   :doc-string "recommendation for the part of the site which is wetland"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
```

```
  :sponsor TOP-SPONSOR)
 (start-inference-engine)
 (INSTANCE INPUT-DATA IS INPUT-VALUES
    WITH WETLAND-VAL YES)
THEN
  (EVALUATE
    (RECON-LINE :WRITE-LINE
". TRY TO PRESERVE AS MUCH AS PORTION OF THE WETLAND AS POSSIBLE. OTHER-
WISE FOR EVERY ACRE OF THE SITE DEVELOPED OUT OF THE WETLAND AREA TWO
ACRES OF THE SITE ARE REPLACED")))


(DEFINE-RULE  ENDANGERED-SPECIES-ACTION-PLANTS
 (:print-name "ENDANGERED-SPECIES-ACTION-PLANTS"
  :certainty 1.0
  :doc-string "criteria for preservation of site used as endangered species habitat"
  :dependency nil
  :direction  :forward
  :explanation-string ""
  :priority 0
  :sponsor TOP-SPONSOR)
 (start-inference-engine)
 (INSTANCE INPUT-DATA IS INPUT-VALUES
    WITH ENDANGERED-PLANT-VAL YES)
THEN
  (EVALUATE
    (RECON-LINE :WRITE-LINE
```

```
". RECONSIDER THE DECISION OF DEVELOPMENT OF THIS PART OF THE SITE FOR ITS
IMPACT ON DEVELOPMENT COST IN TERMS OF ENDANGERED PLANT SPECIES PRO-
TECTION CRITERIA ")))

(DEFINE-RULE  ENDANGERED-SPECIES-ACTION-WILD
  (:print-name "ENDANGERED-SPECIES-ACTION-WILD"
   :certainty 1.0
   :doc-string "criteria for preservation of site used as endangered species habitat"
   :dependency nil
   :direction  :forward
   :explanation-string ""
   :priority 0
   :sponsor TOP-SPONSOR)
  (start-inference-engine)
  (INSTANCE INPUT-DATA IS INPUT-VALUES
      WITH ENDANGERED-WILD-VAL YES)
 THEN
   (EVALUATE
      (RECON-LINE :WRITE-LINE
". RECONSIDER THE DECISION OF DEVELOPMENT OF THIS PART OF THE SITE FOR ITS
IMPACT ON DEVELOPMENT COST IN TERMS OF WILD-LIFE PROTECTION CRITERIA")))
```

# APPENDIX B

## CLASS CHARRETTE

### *INSTRUCTIONS*

- No books may be referred for this charrette.

- Requirements

  Prepare a development plan for the given site.

- Detailed requirements are-

  Site analysis in relation to the physical factors of the site.

  (1 drawing)

  Conceptual design. (1 drawing)

- Important data

- 1 car / 375 sft of parking apace
- 5.5 parking spaces / 1000 sft of gross leasable area

- Both drawings carry equal importance.

- Make site analysis drawing as explanatory as possible.

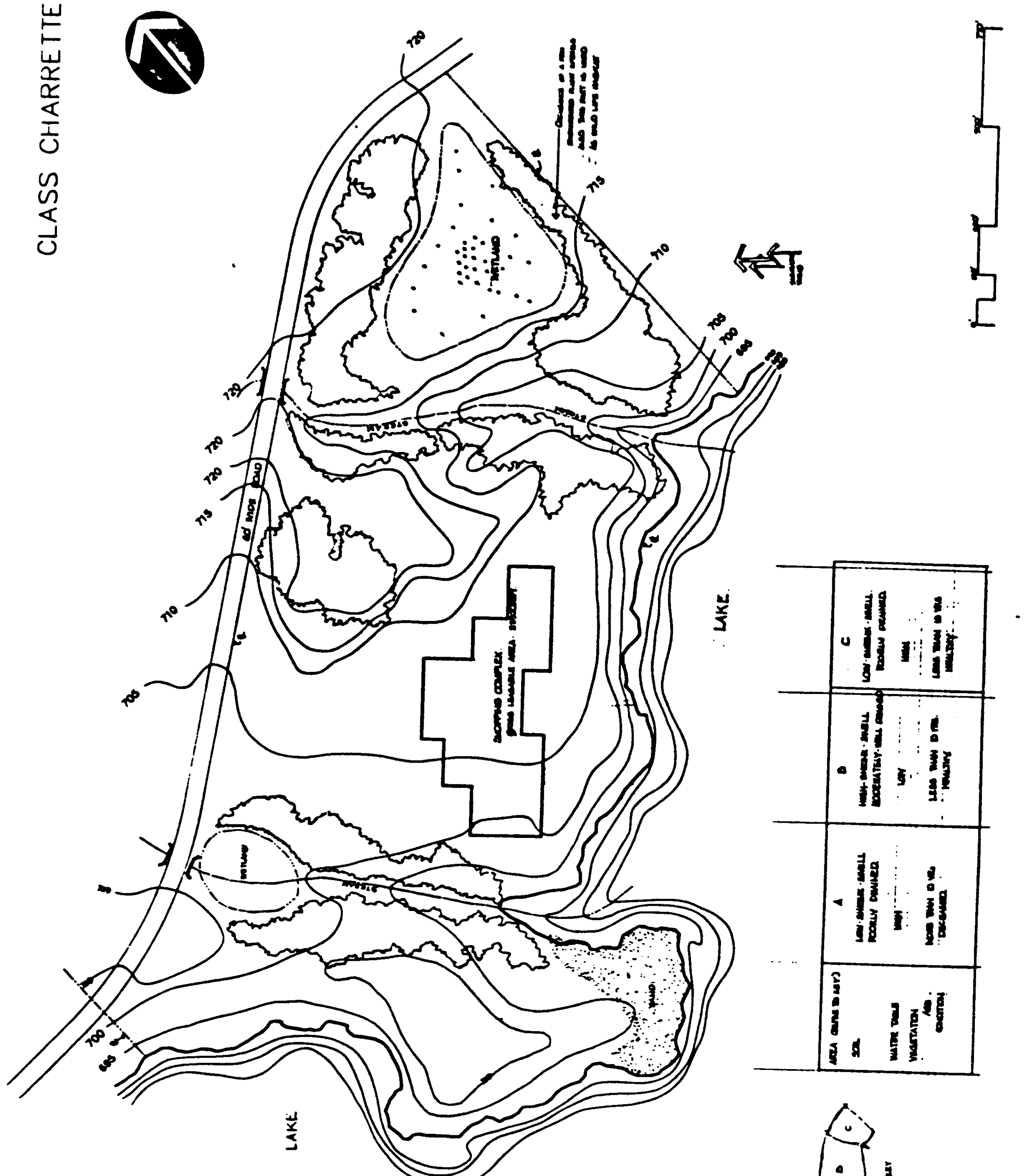
- Connection of site analysis to design decisions is of vital importance.

- Use explanatory notes on drawings where ever you feel necessary.

  - FOR THE GROUPS USING COMPUTER PROGRAM
  - Use the computer program by dividing the site into 500′ X 500′ modules. (around 5 to 7 acres)
  - Use of any graphic medium is allowed.

# INDEX FOR STORM WATER DRAINAGE MANAGEMENT TECHNIQUES

## CLASS CHARRETTE



# APPENDIX C

## EXPERT SYSTEMS DEVELOPED IN ALLIED FIELDS OF ARCHITECTURE

### *Civil Engineering*

STRUCTURAL ENGINEERING

1. ***SACON-*** Developed by Bennett and Englmore (1979) using ES shell EMYCIN. SACON interacts with user for proper application of the MARC finite element structural analysis program. It consist of 170 production rules and 140 consultation parameters such as type of materials, loading and load components. It is intended to help less experienced engineers to use a large general purpose structural analysis software i.e. MARC.

2. ***SSPG-*** Developed by Adeli and Peak (1986) is a small experimental Knowledge based system for design of stiffened steel plate girders. It is written in ELSIP.

3. ***GEPSE-*** Developed by Chehayeb et al. (1985) and written in C. The inference mechanism in GEPSE is forward- chaining. It has been used for the description and verification of a simply supported reinforce concrete beam subjected to a uniformly distributed load.

4. ***BTEXPERT-*** Used for optimum design of bridge trusses. Developed by Adeli and Balasubramanyam (1988)

5. ***RTEXPERT-*** An ES used for design of roof trusses. Developed by Adeli and Al-Rijieh (1987). System is developed in Turbo-Pascal using ES shell INSIGHT 2+.

CONSTRUCTION ENGINEERING AND MANAGEMENT.

1. ***HOWSAFE-*** Developed by Levitt (1986), HOWSAFE uses the micro-computer based ES shell the deciding factor. HOWSAFE is intended to be an electronic means for disseminating the knowledge in construction safety.

2. ***ES for PLANT SELECTION-*** An experimental ES for selecting materials handling equipment for construction of concrete frame building is described by Wijesundera and Harris (1985). This system has been developed using ES shell Savoir.

3. ***DURCON-*** Developed by Clifton et al. DURCON ia a prototype ES for selecting the constituents of concrete exposed to aggressive environments. DURCON is a rule-based system developed in PASCAL on an IBM PC using the forward chaining mechanism.

ENVIRONMENTAL ENGINEERING

1. An ES for fault diagnosis of hazardous waste incineration facilities, developed by Huang et al. (1986). This system is designed to diagnose malfunctioning in hazardous waste incineration facilities, using micro-computer based ES shell M-1.

2. An ES for selection of Waste disposal landfill sites. Developed by Rouhani and Kangari (1987) presented a prototype ES for the selection of waste disposal landfill using INSIGHT 2+ ES shell. The knowledge base is based on the rules from the US environmental protection agency manual for ranking of uncontrolled hazardous waste sites.

3. An ES for assessing groundwater contamination potential by organic chemicals. Developed by Ludvigsen et al. (1986), this system assesses the groundwater contamination potentials due to organic chemicals.

### *Geotechnical Engineering*

1. ***SITECHAR***- An ES for geotechnical site categorization. Rehak (1985) developed a framework for developing an ES for interpretation of geotechnical site characterization data leading to a probable geometry of deposits and engineering properties of sub-grade soils, called SITECHAR. Knowledge types included in the knowledge base of initial SITECHAR are knowledge of geometry and trends, matching soils by description, geomorphology, geology, searching for marker beds and proximity.

2. ***CONE***- Developed by Mullarkey et al. (1986) CONE is developed for interpretation of geotechnical characterization data from cone penetrometer logs. CONE is intended to check validity of raw data, to classify soil types, to profile the soil strata and to infer the geotechnical design parameters such as the friction angle of sands and undrained shear strength of clays.

3. An ES for design of retaining walls- Developed by GERO and Coyne (1986), the knowledge base of the system contains knowledge of various prototypical retaining wall cross-sections and how to select among them. The ES has been implemented in QUINTUS PROLOG and the graphics in C.

## *References*


1. Bennett, J. S. and Englemore, R. S., ***SACON: a knowledge-based consultant for structural analysis.*** Preceedings 6th International joint conference on Artificial Intelligence, Tokyo, 1979, pp. 47-49.

2. Adeli, H. and Paek, Y., ***Computer aided design of structures using LISP*** Journal of Computers and Structures, 22, No. 6, 1986, pp. 939-956.

3. Chehayeb, F.S. et al., ***An environment for building engineering knowledge based systems*** Application of Knowledge-Based systems to Engineering Analysis and Design (ed. C. L. Dym), AD-10, ASME, New york, 1985, pp. 9-28.

4. Adeli H. and Balasubramanyam, K. V. ***A knowledge based system for design of bridge trusses*** Journal of computing in Civil Engineering, ASCE, vol-2, No. 1, 1988.

5. Adeli H. and Al-Rijleh, M. M. ***An expert system for design of roof trusses*** Microcomputers in Civil Engineering, vol-2, No. 3, 1987.

6. Ludvigsen, P. J. et al. ***A demonstration expert system to aid in assesing groundwater contamination potential by organic chemicals*** Preceedings 4th conference on Computing in Civil Engineering, ASCE, New York, 1986.

7. Rehak, D. R. et al., ***SITECHAR: an expert system component of a geotechnical site characterization workbench*** Applications of Knowledge-Based Systems to Engineering Analysis and Design (ed. C. L. Dym), AD-10, ASME, New-york, 1985, pp. 117-133.

8. Mullarkey, P. W. et al., ***CONE: An Expert System for Interpretation of Geotechnical Characterization Data from Cone Penetrometers*** Report R-85-147, Department of Civil Engineering, Carnegie-Mellon University, 1985.

9. Gero J. S. and Coyne, R. D., ***Developments in expert systems for design synthesis*** Expert Systems in Civil Engineering, ASCE, New York, 1986, pp. 193-203.

The vita has been removed from the scanned document