NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

DEVELOPMENT OF EXPERT SYSTEM FOR
HOTEL SELECTION BY USING
VP-EXPERT

Graduation Project
COM-400

Student: Aladdin Alsoweity (20020931)

Supervisor: Assist. Prof. Dr.Elbrus Irnarıov

Nicosia .. 2006

ACiavOWLEDGMENTS

First of all, I thank Godfor giving me the blessing, the stamina and courage to achieve my objectives.

Foremost I would like to pay my special thanks to my parents, my brothers, and my sisters their helped me on every phase in my life. They boosted me up to reach as well to this point.

I am very much thankful and grateful to my mother and my father their prayers and love for me have encouraged me so make this day come true. They had given me the supporting and consolidation through the time of my study. They it is only because of them that today I am capable of completing my degree.

I would like to thank to my supervisor, Assist. Prof Dr. Elbrus Imanov for intellectual support, the encouragement, which made my project possible and complete, and his patience for correcting any error and I am really thankful to my doctor. He was not just my supervised but also he was my friend.

I want to thank to all friends *whom gave me any encouragement through the studying. I want to honor those all persons who have supported me or helped me in the time of my studying,*

To all of them, my love and respect

ABSTRACT

The aim ofmy project is to design the expert system for solving the problems for hotel system. that need for the reasoning, after putted the information's (knowledgebase) from through the human reasoning, and will give to teach the human how to build expert systems from put the information's (knowledge base), and collect much information's about what the expert human wants to solve the problems.

In my project it include much information (knowledge base) about hotel system and collected these information's after reasoning what it will be my program solve the problem, this program it will display for the user five questions, the user will choose from the multi-choosing, after what the user choose, the program it will give in the fact the names hotels from through what the user choose. This program about hotel system used by VPX system, and shows how to implement them in VPX expert program, arid how to use them to solve problems.

Expert systems can be developed with Expert System Shells. An expert system shell is a software programming environment which enables the construction of expert or knowledge based systems. So the expert systems software can be developed for any problem that involves a selection from among a definable group of choices where the decision is based on logical steps. Any area where a person or group has special expertise needed by others is a possible area for an expert system. Expert systems can help automate anything from complex regulations to aiding customers in selecting from among a group of products, or diagnosing equipment problems.

TABLE OF CONTENTS

INTRODUCTION

My project will give to teach you how to build expert systems from put knowledge base, and collect the information about what you want to solve the problems, shows how to implement them in VPX expert program, and how to use them to solve problems.

In chapter one, it will talk about expert system generally, what is expert system?! Where we can use it?! , why is the important to use the expert system in our life, what is the components for the expert system, many questions of expert system and more details will shows about expert system. And many advantage and disadvantage in final this chapter.

In chapter two, it will explain about knowledge representation, what is knowledge representation?!, and many detail in introduction, what are the representations methods?!, and it will explain each method, although used the production method in my program, and it is the most commonly use it, but it will give other methods just for information's compare between them advantage and disadvantage knowledge representation.

In chapter three, it will talk about (IF...THEN) rules based expert system, rule-based expert systems have played an important role in modem intelligent systems and their applications in strategic goal setting, planning, design, scheduling, fault monitoring, diagnosis and so on. That it is important to solve real-world problems that normally would require human intelligence.

In chapter four, it will give detail about VP-Expert, what is VPX program, what is the components main menu, how you can use it, so it will explain the VPX program step-step to become easy to understand.

In chapter five, it will explain my program, for what it used and how the program it will work when the user choose from the multi-choose and what will display in the fact, in finally from this chapter it will show for the reader the code in my program.

CHAPTER 1.INTRODUCTION TO EXPERT SYSTEM (ES)

1.1 Overview

An expert system is a computer program which uses non-numerical domain-specific knowledge to solve problems with a competence comparable with that of human experts. We can say another definition that expert system is a computer system which emulates the decision-making ability of a human expert.

Anyway, expert system (ES) is a class of Artificial Intelligence (AI) system. They are able to perform various functions, namely: they can consul and advice, analyses and classify, learn and teach, make search, exchange information between system and represent it in the required form, identify and interpret, diagnose and test. They also can control, design, explain, investigate, generate concept, predict and schedules must solve problem, which always require human expert's participation.

ES are created with participation of specialists who are able to explain their sequence of thoughts during concrete problem solving process. An important problem in research on ES is the attainment of functioning level similar to human expects who are to solve separate problem in various situations.

An expert system contains knowledge derived from an expert in some narrow domain. This knowledge is used to help individuals using the expert system to solve some problem. I mention narrow domain since it is quite difficult to encode enough knowledge into a system so that it may solve a variety of problems. We have not reached point yet where this can be done.

The traditional definition of a computer program is usually:

Algorithm + *data structures* =*program*

In an expert system, the definition changes to:

Inference engine+ knowledge= expert system

1.2 Why Use an Expert System

There are many reasons to use an expert system. Here are some of the primary reasons:

1. Helps preserve knowledge=builds up the corporate memory of the firm.
2. Helps if expertise is scarce, expensive, or unavailable.
3. Helps if under time and pressure constraints.
4. Helps in training new employees.
5. Helps improve worker productivity.

1.3 Expert System Use

Expert systems are used in a variety of areas, and are still the most popular developmental approach in the artificial intelligence world. The table below depicts the percentage of expert systems being developed in particular areas:

Area	Percentage
Production/Operations Mgmt	48%
Finance	17%
Infonnation Systems	12%
Marketing/Transactions	10%
Accounting/Auditing	5%
Intemational Business	3%
Human Resources	2%
Others	2%

Table 1.1 Expert System Use

1.4 Components of an Expert Systems

An expert system consists of five main components:

- Knowledge Base.
- Inference Engine.
- User Interface.
- Explanation Subsystem (JUSTIFIR).
- Knowledge Refining System.
- Blackboard.

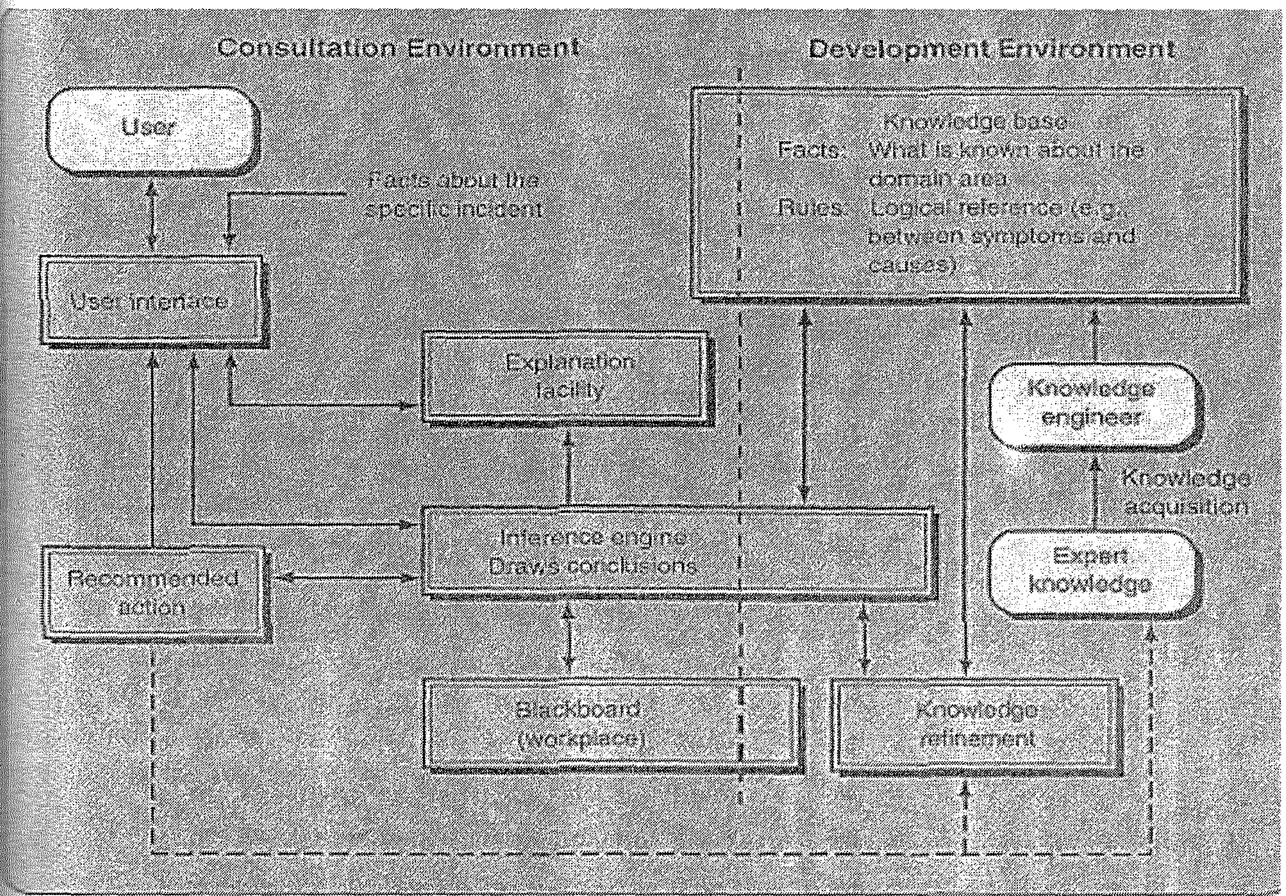


Figure 1.1 Components of an Expert System

1.4.1 Knowledge Base

The knowledge base contains knowledge necessary for understanding, formulating, and solving problems. It includes two basic elements: (1) facts, such as the problem situation and theory of the problem area, and (2) special heuristics or rules that direct the use of knowledge to solve specific problems in a particular domain.

1.4.2 Inference Engine

The "brain" of the ES is the inference engine, also known as the control structure or the rule interpreter (in rule-based ES). This component is essentially a computer program that provides a methodology for reasoning and formulating conclusions. The major functions of the inference engine include the following.

1. Determining which rules will be used.
2. Presenting the user with a question, whenever needed.
3. Adding the answers to the ES memory.
4. Inferring a new fact from a rule.
5. Adding the inference fact to the memory.

1.4.3 User Interface

Expert systems manage communication between the user and the computer. This communication could best be carried out in a natural language, usually presented as questions and answers and sometimes supplemented by graphics. Future interfaces will include voice understanding and synthesis.

1.4.4 Knowledge Refining System

Human experts have a knowledge refining system; that is, they can analyze their own performance, learn from it, and improve it for future consultations. Similarly, such evaluation is necessary in computerized learning so that the program will be able to analyze the reasons for its success or failure. This can lead to improvements that result in a better knowledge base and more effective reasoning. Such a component is not available

III commercial expert systems at the moment, but it is being developed {based on machine learning techniques) in experimental ES.

1.4.5 Explanation Subsystem (JUSTIFIR)

The ability to trace responsibility for conclusions to their source is crucial both in the transfer of expertise and in problem solving.

The explanation subsystem can trace such responsibility and explain the ES's behavior by interactively answering questions such as the following

- ../ Why was a certain question asked by the expert system?
- ../ How was a certain conclusion reached?
- ../ Why was a certain alternative rejected?
- ../ What is the plan to reach the solution?

1.4.6 Blackboard

The blackboard is an area of working memory set aside for the description of a current problem, as specified by the input data; it is also used for recording intermediate results. It is a kind of database.

1.5 Expert System Architecture

Figure 1.2 shows the most important modules that make up a rule-based expert system. The user interacts with the system through a user interface which may use menus, natural language or any other style of interaction). Then an inference engine is used to reason with both the expert knowledge (extracted from our friendly expert) and data specific to the particular problem being solved. The expert knowledge will typically be in the form of a set of IF-THEN rules. The *case* specific data includes both data provided by the user and partial conclusions (along with certainty measures) based on this data. In a simple forward chaining rule-based system the case specific data will be the elements in working memory.

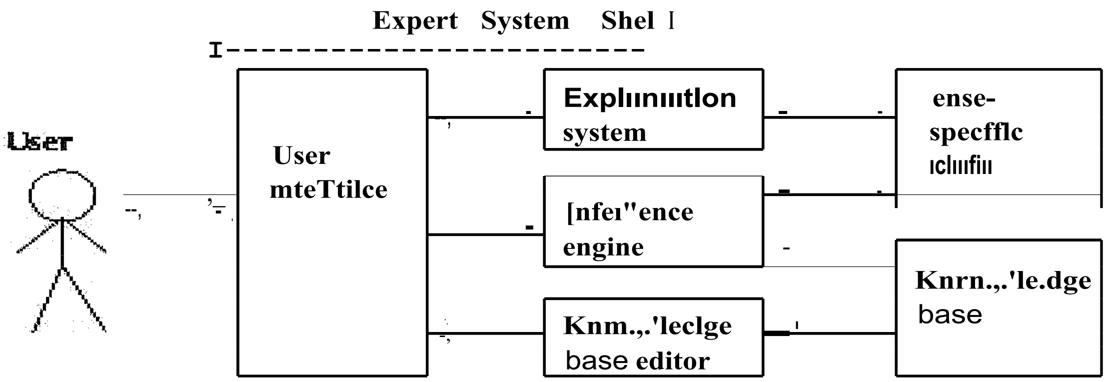


Figure 1.2 Expert System Architecture

Almost all expert systems also have an explanation subsystem, which allows the program to explain its reasoning to the user. Some systems also have a knowledge base editor which helps the expert or knowledge engineer to easily update and check the knowledge base.

One important feature of expert systems is the way they (usually) separate domain specific knowledge from more general purpose reasoning and representation techniques. The general purpose bit (in the dotted box in the figure) is referred to as an expert system shell. As we see in the figure, the shell will provide the inference engine (and knowledge representation scheme), a user interface, an explanation system and sometimes a knowledge base editor.

Given a new kind of problem to solve (say, car design), we can usually find a shell that provides the right sort of support for that problem, so all we need to do is provide the expert knowledge. There are numerous commercial expert system shells, each one appropriate for a slightly different range of problems. (Expert systems work in industry includes both writing expert system shells and writing expert systems using shells.) Using shells to write expert systems generally greatly reduces the cost and time of development (compared with writing the expert system from scratch).

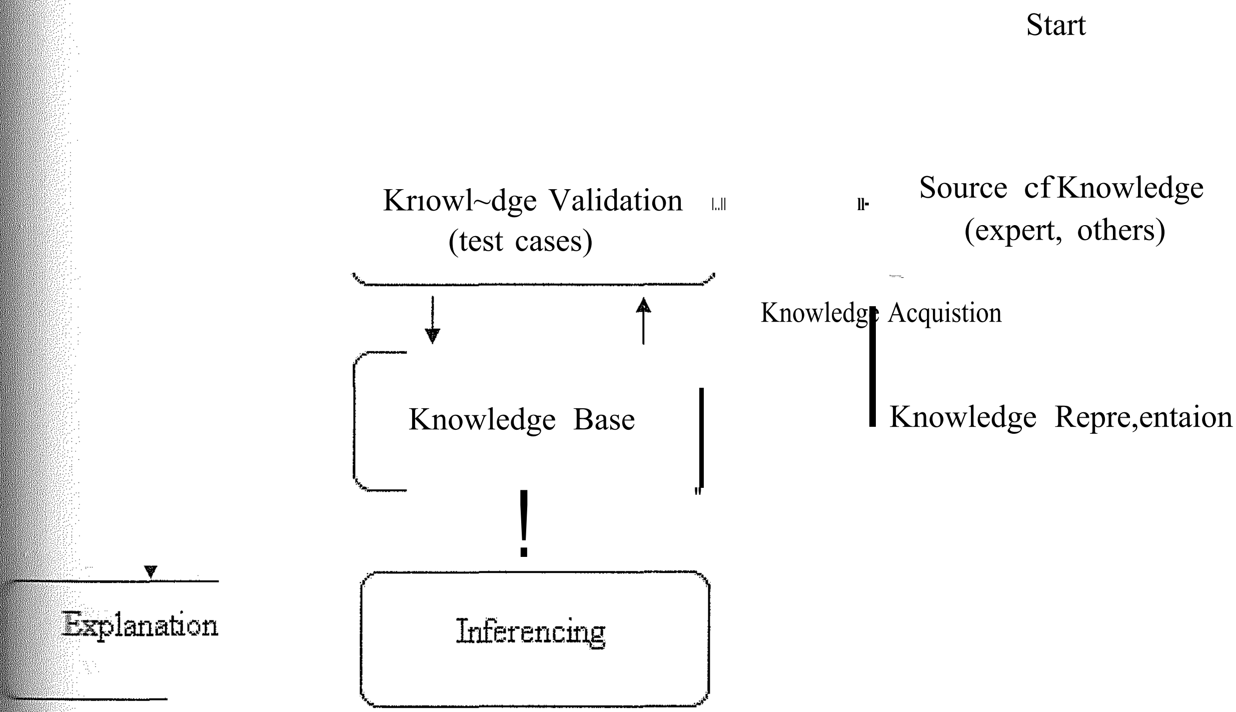
1.6 Building of an Expert System

Constructing an ES may be done in two different ways. Either one builds all (or some) of the components described in the previous section, or one acquires all the components except the content of the knowledge base which must be built. The latter option is known as building an expert system with a shell. That is, all the components of the ES, except the knowledge base, are fairly generic. Therefore, they can be reused for different expert systems.

The knowledge is the content, or the "inside" of the shell. Using shells, we can build systems much faster and with less programming skills. Building an ES with a shell, however, has limitations: the builder is constrained by the features of the shell so he has less flexibility. In addition, one can buy a ready-made ES off the shelf. But, the number of such ES is limited and they can be used only for generic applications.

1.6.1 Knowledge Engineering

The process of building expert systems by either of the previous approaches ıs described as knowledge engineering. The process includes the following five activities: knowledge acquisition, knowledge validation, knowledge representation, infringing,and explanation,(see Figure 1.3).



Figurel.3 KnowledgeEngineering

1.6.1.1 Knowledge Acquisition

Knowledge is acquired from human experts, books, documents, sensors, or other sources such as computer files. The knowledge may be specific to the problem domain and the problem-solving situation, it may be general knowledge (e.g., knowledge about business), or it may be met knowledge (knowledge about Knowledge). By the latter, we mean information about how experts use their knowledge to solve problems.

1.6.1.2 Knowledge Validation

Knowledge is validated and verified (e.g., by using test cases) until its quality is acceptable.

1.6.1.3 Knowledge Representation

The acquiredknowledgeis organizedin an activitycalled knowledgerepresentation. This activity involvespreparationof a "knowledgemap" and encoding the knowledge into the knowledgebase, we will see knowledgerepresentationin chapter 2.

1.6.1.4 Inference

Inference involves the design of software that will enable the computer to make inferences based on knowledge that a user has regarding a specific problem and then provide advice to the user.

1.6.1.5 Explanation

The design of an explanation is provided by an explanation facility. For example, this includes programming the ability to answer questions like *why* a specific piece of information is needed by the computer or how a certain conclusion was derived by the computer.

1.7 The Development Process

Because an ES is basically computer software, its development follows a software developmentprocess using a prototyping approach. The various tasks that are encountered in building an ES are organized into six phases.

1.7.1 Participant in the Development Process

Several humans usually participate in the development o [an expert system. But at minimum, there is a builder (the knowledge engineer) and there is an expert.

1.7.1.1 The Expert

The expert is a person who has special knowledge, judgment, experience, and methods as well as the ability to apply these talents to give advice and solve problems. It is the expert's job to provide knowledge about how he accomplishes the task that the ES will perform.

The expert knows which facts are important and understands the meaning of the relationships among facts. In diagnosing a problem with an automobile's electrical system, for example, an expert mechanic knows that fan belts can break and cause the battery to discharge. Directing a novice to check the fan belt and interpreting the meaning of a loose or missing belt are examples of expertise. When more than one expert is used, situations can become difficult if the experts disagree.

1.7.1.2 The knowledge Engineer

The knowledge engineer helps the expert(s) structure the problem area by interpreting and integrating human answers to questions, drawing analogies, posing counter examples, and bringing conceptual difficulties to light. She is usually also the system builder.

The shortage of experienced knowledge engineers is a major bottleneck in ES construction. To overcome this bottleneck, ES designers are using productivity tools (e.g., special editors), and research is being conducted on building systems that will bypass the need for knowledge engineers. In some cases, the expert can act as the knowledge engineer with the support of special software tools.

1.7.1.3 The Use

Most computer-based systems have evolved targeted to a single-user mode. In contrast, an ES has several possible types of users. A no expert, client seeking direct advice. In such a case, the ES acts as a consultant or advisor. A student who wants to learn. In such a case, the ES acts as an instructor. An ES builder who wants to improve or increase the knowledge base. In such a case, the ES acts as a partner.

An expert, in such a case, the ES acts as a colleague or an assistant. For example, an ES can provide a "second opinion," so the expert can validate his judgment. An expert can also use the system as an assistant to carry on routine analysis of computations or to search for and classify information.

1.7.2 The Benefit of Expert System

Why have ES become so popular? It is because of the large number of capabilities and benefits ES provide. A few of these are described next.

1.7.2.1 Increased Output and Productivity

Experts systems can work faster than humans. For example, a system called :,{CON has enabled Digital Equipment Corporation (DEC) to increase fourfold the production preparation of minicomputers which are customized for their clients. Thus, DEC is using a mass customization approach in their production.

1.7.2.2 Increased Quality

Expert system can increase quality by providing consistent advice and reducing the error rates. For example, XCON reduced the error rate of configuring computer orders from 35 percent to 2 percent

1.7.2.3 Capture of Scare Expertise

The scarcity of expertise becomes evident in situations where there are not enough experts for a task, where the expert is about to retire or leave the job, or where expertise is required over a broad geographic location.

1.7.2.4 Operation in Hazardous Environment

Many tasks require human's to operate in hazardous enviromnents. ES that interprets information collected by sensors enables workers to avoid hot, humid, or toxic environments, such as a nuclear power plant that has malfunctioned.

1.7.2.5Accessibility to Knowledge and Help Desks

Expert systems make knowledge (and information) accessible to many people in several locations. People can query systems and receive advice. One area of applicability is in support of help desks.

1.7.2.6 Reliability

ES are reliable. They do not become tired or bored, call in sick, or go on strike; and they do not talk back to the boss. ES also consistently pay attention to all details and do not overlook relevant information and potential solutions.

1.7.2.7 Increased Capabilities of Other Computerized System

Integration of ES with other systems makes the other systems more effective; the integrated systems cover more applications, work faster, and produce higher-quality results.

1.7.2.8 Ability to Work with Incomplete or Uncertain Information

hı contrast to conventional computer systems, ESs can work with incomplete information, like human experts. The user can respond with a "don't know" or "not sure" answer to one or more of the system's questions during a consultation, and the ES will' still be able to produce an answer, although it may not be a certain one.

1.7.2.9 Benefits to End Users

Primarily, the benefits of ESs to end users include:

- A speed-up of human professional or semi-professional work typically by a factor of ten and sometimes by a factor of a hundred or more.
- Within companies, major internal cost savings. For small systems, savings are sometimes in the tens or hundreds of thousands of dollars; but for large systems, often in the tens of millions of dollars and as high as hundreds of millions of dollars. These cost savings are a result of quality improvement, a major motivation for employing expert system technology.
- Improved quality of decision making. In some cases, the quality or correctness of decisions evaluated after the fact show a ten-fold improvement.
- Preservation of scarce expertise. ESs is used to preserve scarce know-how in organizations, to capture the expertise of individuals who are retiring, and to preserve corporate know-how so that it can be widely distributed to other factories, offices or plants of the company.

1.8 Application of Expert System

1.8.1 Diagnosis and Troubleshooting of Devices and Systems of All Kinds

This class comprises systems that deduce faults and suggest corrective actions for a malfunctioning device or process. Medical diagnosis was one of the first knowledge areas to which ES technology was applied (for example, see Shortliffe 1976), but diagnosis of engineered systems quickly surpassed medical diagnosis. There are probably more diagnostic applications of ES than any other type.

1.8.2 Planning and Scheduling

Systems that fall into this class analyze a set of one or more potentially complex and interacting goals in order to determine a set of actions to achieve those goals, and/or provide a detailed temporal ordering of those actions, taking into account personnel, materiel, and other constraints. This class has great commercial potential, which has been recognized. Examples involve airline scheduling of flights, personnel, and gates; manufacturingjob-shop scheduling; and manufacturing process planning.

1.8.3 Configuration of Manufactured Objects from Subassemblies

Configuration, whereby a solution to a problem is synthesized from a given set of elements related by a set of constraints, is historically one of the most important of expert system applications. Configuration applications were pioneered by computer companies as a means of facilitating the manufacture of semi-custom minicomputers (McDermott 1981). The technique has found its way into use in many different industries, for example, modular home building, manufacturing, and other problems involving complex engineering design and manufacturing.

1.8.4 Financial Decision Making

The financial services industry has· been a vigorous user of expert system techniques. Advisory programs have been created to assist bankers in determining whether to make loans to businesses and individuals. Insurance companies have used expert systems to assess the risk presented by the customer and to determine a price for the insurance. A typical application in the financial markets is in foreign exchange trading.

1.8.5 Knowledge Publishing

This is a relatively new, but also potentially explosive area. The primary function the expert system is to deliver knowledge that is relevant to the user's problem, in the context of the user's problem. The two most widely distributed expert systems in the world are in this category. The first is an advisor which counsels a user on appropriate grammatical usage in a text. The second is a tax advisor that accompanies a tax preparation program and advises the user on tax strategy, tactics, and individual tax policy.

1.8.6 Process Monitoring and Control

Systems falling in this class analyze real-time data from physical devices with the goal of noticing anomalies, predicting trends, and controlling for both optimality and failure correction. Examples of real-time systems that actively monitor processes can be found in the steel making and oil refining industries.

1.8.7 Design and Manufacturing

These systems assist in the design of physical devices and processes, ranging from high-level conceptual design of abstract entities all the way to factory floor configuration of manufacturing processes.

1.9 Some Notes on Verification & Validation

There are two areas the expert system developer should take note of in the verification & validation process which are peculiar to expert systems: inconsistencies and incompleteness. Inconsistencies can be caused by redundant rules, conflicting rules, subsumed rules, unnecessary premise clauses, and circular rule chains. Incompleteness can be caused by unachievable antecedents or consequences and unreferenced or illegal values.

1.10 Expertise and knowledge

Expertise is the extensive, task-specific knowledge acquired from training, reading, and experience. Knowledge enables experts to make better and faster decisions than no experts in solving complex problems. It takes a long lime (usually several years) to become an expert. Expertise is distributed in organizations in an uneven manner.

The objective of an expert system is to transfer expertise from an expert a computer and then on to other humans (no experts). This process involves fc activities: knowledge acquisition (from experts or other sources), knowledge representation (in the computer), knowledge inference, and knowledge transfer to the use.

Knowledge acquisition involves obtaining knowledge from experts or from documented sources. Through the activity of knowledge representation, acquired knowledge is organized in one of several possible configurations and stored, electronically, in a knowledge base. A unique feature of an expert system is its ability to reason. Given that the necessary expertise is stored in the knowledge base a program has accessibility to databases, the computer is programmed that it can make inferences.

The inference is performed in a component called the inference engine. The inference results in a piece of advice or a recommendation for novices. Thus, the expert's knowledge has been transferred to users. A unique feature of an ES is its ability to explain its advice or recommendation. The explanation and justification is done in a subsystem called the justified or the explanation subsystem.

1.11 Advantages and Disadvantages

1.11.1 Advantages of Expert Systems

- Permanence - Expert systems do not forget, but human experts may.
- Reproducibility - Many copies of an expert system can be made, but training new human experts is time-consuming and expensive.
- If there is a maze of rules (e.g. tax and auditing), then the expert system can "unravel" the maze.
- Efficiency - can increase throughput and decrease personnel costs. Although expert systems are expensive to build and maintain, they are inexpensive to operate. Development and maintenance costs can be spread over many users. The overall cost can be quite reasonable when compared to expensive and scarce

- human experts. Cost savings: Wages - (elimination of a room full of clerks) Other costs - (minimize loan loss).
- Consistency - With expert systems similar transactions handled in the same way. The system will make comparable recommendations for like situations.
- Humans are influenced by regency effects (most recent information having a disproportionate impact on judgment) primacy effects (early information dominates the judgment).
- Documentation - An expert system can provide permanent documentation of the decision process.
- Completeness - An expert system can review all the transactions, a human expert can only review a sample.
- Timelines - Fraud and/or errors can be prevented. Information is available sooner for decision making.
- Breadth - The knowledge of multiple human experts can be combined to give a system more breadth that a single person is likely to achieve.
- Reduce risk of doing business.
- Consistency of decision making.
- Documentation.
- Achieve Expertise.
- Entry barriers - Expert systems can help a firm create entry barriers for potential competitors.
- Differentiation - In some cases, an expert system can differentiate a product or can be related to the focus of the firm.
- Computer programs are best in those situations where there is a structure that is noted as previously existing or can be elicited.

1.11.2 Disadvantages of Rule-Based Expert Systems

- Common sense - In addition to a great deal of technical knowledge, human experts have common sense. It is not yet known how to give expert systems common sense.
- Creativity - Human experts can respond creatively to unusual situations, expert systems cannot.
- Learning - Human experts automatically adapt to changing environments; expert systems must be explicitly updated. Case-based reasoning and neural networks are methods that can incorporate learning.
- Sensory Experience - Human experts have available to them a wide range of sensory experience; expert systems are currently dependent on symbolic input.
- Degradation - Expert systems are not good at recognizing when no answer exists or when the problem is outside their area of expertise.

1.12 Summary

One of the most powerful attributes of expert systems is the ability to explain reasoning. Since the system remembers its logical chain of reasoning, a user may ask for an explanation of a recommendation and the system will display the factors it considered in providing a particular recommendation. This attribute enhances user confidence in the recommendation and acceptance of the expert system.

CHAPTER 2.KNOWLEDGE)lEPRESENTATION (KR)

2.1 Overview

What is a knowledge representation? We argue that the notion can best be understood in terms of five distinct roles it plays, each crucial to the task at hand:

A knowledge representation (KR) is most fundamentally a surrogate, a substitute for the thing itself, used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than taking action in it.

It is a set of ontological commitments, i.e., an answer to the question: In what terms should I think about the world? It is a fragmentary theory of intelligent reasoning, expressed in terms of three components: (i) the representation's fundamental conception of intelligent reasoning; (ii) the set of inferences the representation sanctions; and (iii) the set of inferences it recommends.

We believe that viewing representations in this way has consequences for both research and practice. For research, this view provides one direct answer to a question of fundamental significance in the field. It also suggests adopting a broad perspective on what's important about a representation, and it makes the case that one significant part of the representation endeavor=capturing and representing the richness of the natural world--is receiving insufficient attention. We believe this view can also improve practice by reminding practitioners about the inspirations that are the important sources of power for a variety of representations.

The knowledge in the knowledge base may be organized differently from that in the inference engine through a variety of knowledge representation schemes. The two most common representation methods are described: production rules and frames, but also has other methods there are semantic networks, logic, hybrid and summary, Let's go to discuses about these methods.

2.2 Representation Methods

2.2.1 Production Rules

The concept of production rules was developed by Newell and Simon (1973) for their model of human cognition. The basic idea of these systems is that knowledge is presented as rules in the form of conditions-action pairs: "IF this condition occurs, THEN some action will (or should) occur." Expert systems whose knowledge is represented solely as production rules arc referred to as rule-based systems. Rules may appear **in** different forms. Some examples include the following. The rules are usually expressed in the general form:

IF condition(s)

THEN action(s)

The (if) portion, or left-hand Sid, describes a problem-solving situation, in the form of a set of conditions, sometimes called antecedents. These antecedents must be true in order for this rule to be applicable. The (then) portion, or right-hand side, described the set of actions, sometimes called consequents that follow if the rule is applicable. I will give simple example about production rules:

IF speed of car is too fast

THEN take your foot off the accelerator;

The above example show that if the specific condition "speed of car is too fast" is met, then the action part is invoked. Rules may involve several conditions and/or several actions, which can be combined with the use of AND, called conjunction, and OR, called disjunction.

The following example shows two conditions that make use conjunction. This example it will use of conjunction rule:

IF CGPA for student >=2.0

AND the last semester

THEN the student will graduate;

The above example mains the both of statement" CGPA for student>= 2.0 ", and the statement" the last semester" are true then the action" the studentwill graduate" will pass, both of them it must true, not one of them ,but if one of them true and another statement is false then the action will be notpass.

2.2.1.1HeuristicRules

Each ıule in a ıule base can be used to represent a modular chunk of expert problem -solving knowledge. These are known as problem-solving heuristics. For example a car fault diagnostic domain might be:

IF the result of switching on the headlights= "nothing happens"

THEN the recommended action= "recharge or replace the battery";

2.2.1.2 Domain Rules

Rules can be used to represent relation between objects in the domain. For example:

IF an animal is a tiger,

THEN the animal is a cat

This ıule expresses a relation between species of an animal. That is, a tiger is a member of the cat family. This means that ıules can also express relation between classes of objects.

2.2.1.3 Procedural Knowledge

Rules can also express action sequences, for example:

IF the traffic light is green

THEN the action is stop

This ıule describes an action or procedure to be taken following the occurrence of an event or situation; as such this is called procedural knowledge.

2.2.1.4 The Production Models of Knowledge Representation (KB)

Recently production models (rules) in knowledge representation systems are widely used. Suggested primarily by Post in 1943 they were utilized for the first time in AI systems in 1972. Production models could be implemented both procedurally and declarative. Production models are set of rules of "condition-action" type, where conditions are assertions on some database content and actions and procedures that could modify database content. There are three main components in production models:

1. Unstructured or structured database.
2. Number of production rules or, simply productions. Eacli production consists of two parts:
   i. Conditions (antecedent); here some conditions to be carried out in database for performing corresponding actions are determined.
   ii. Actions (consequent); this part includes description or actions to be executed in database, when corresponding conditions are fulfilled. In simplest production systems (PS) they 'Just determine what the elements must be added (or, sometimes, removed) to database.
3. Inference engine, which consequently determines what the productions must be activated, depending on their conditions part; select one of "the suitable production rules in given situation; performs the actions of the selected procedure.

The Production models mainly are applied as solvers or inference engines. In system's database the known facts from some problem domain are kept. Productions include the specific for this domain knowledge about what additional facts could be assumed, when the specific data are found in database. Productions actions may consist of active procedures, which perform necessary operations on database content automatically (or like demons check themselves on their activation conditions fulfillment). In this case the knowledge representation form is procedural, though very limited form. During following iterations the facts, added to database can activate other productions etc.

As a rule, in classical production systems the facts are the variable part of system, but rules and interpreter are constant. Being implemented procedurally, the production models have very attractive characteristic of modularity. That is why laıowledge can be added or removed without any side effects.

At present, production models are widely used for Implementing of expert systems, i.e. the systems, armed with human expert laıowledge on some problem area. As a rule, classical production -models don't include the Information about application, i.e. for example, and knowledge about what production to use to achieve the goal. It leads to significant decreasing of their efficiency: although just one production can be activated in each iteration, all productions conditions must be examined. For large number of rules it could require a significant expense of resources. Moreover, the sequence of productions performing on each iteration is depending on the all system variables states. In this case the problem of combinatorial "explosion" occurs.

To solve these problems the approaches, related with database structure and productions conditions improvement methods were suggested. These methods allow increasing the efficiency of functioning. Also the attempts to influence on control course j undertaken.

2.2.1.5 Advantage of Production Rules

The following factors have contributed to the widespread use of rules in expert system they are:

- Simplicity. Rules form a good psychological model for laıowledge representation, because they closely relate to human reasoning. This makes rule-based system easy to build in comparison with other methods for representing laıowledge.
- Modularity. This means that block of rules can be independently written and added to a rule base, and checked for correctness. So rule-based expert system can be broken down into easily manageable components for development. It is this feature that enables laıowledge bases to be constructed incrementally, step by step.
- Explanation. Rule-based systems provide simple transparent explanation facilities.

- Handling uncertainty. A number of techniques have been developed that allow knowledge about uncertainty to be contained within rules.

2.2.1.6 Disadvantages of Production Rule System

Although individual production rules may be transparent, the combined operation and effect may be opaque - especially in systems that contain a large number of rules (i.e. more than 200). This is because some rules in a knowledge base may be domain rules and other may be procedural or control rules. There is no clarity of destination in the uniform if... then syntax, and therefore the overall effect might be difficult to follow. This can make debugging particularly different.

Another shortcoming of rule-based system is that some expert system knowledge cannot adequately be captured in the form of rules. For example, in medical expert system, patient histories are very important. However, rule based system would rely on databases that store such data externally to accomplish something like this. Sometimes it is difficult to achieve successful (Beynon-Davis 1991).Finally, rule-base systems have an inherent inability to learn in that they do not have the ability to modify themselves. These characteristics would be essential for learning system.

2.2.2 Semantic Networks

Semantic networks provide a graphical way ofrepresenting declarative knowledge. The concept was first introduced by (Ross Quillian 1968) from working with language understanding. Knowledge can be represented about objects, concepts and events. And the specific relationships between them. In essence, a semantic network consists of a set nodes and links. The nodes represent objects or concepts. The links indicate the relationships between them. A simple example is Fig.2.1.Inheritance is mechanism for ensuring that an object is not represented by more nodes or links than necessary.

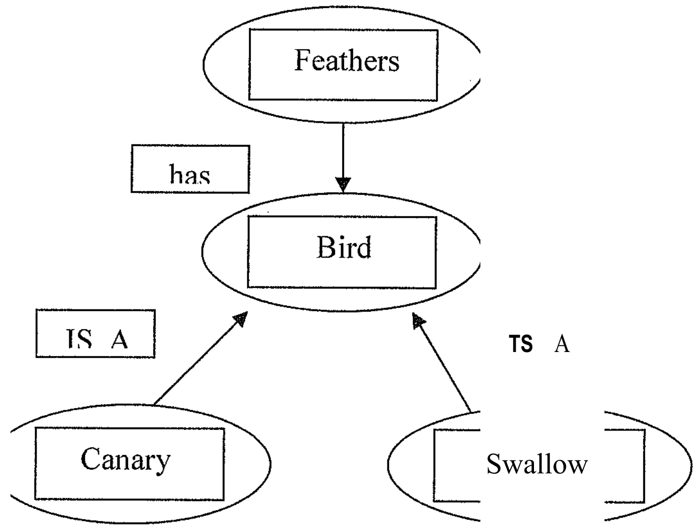


Figure 2.1 A Simple Semantic Network for Bird

This example show that the object Swallow is a bird, and the object Canary is a bird. A hierarchical network of relationships of objects can be developed using a network of this kind. Since the object Bird has Feathers, then the descendant objects canary an Swallow also possess Feathers because they are both members of the class of object called Bird. This is called inheritance. With inheritance, an object can automatically take on all the properties of its super classes. Another example of inheritance is given in the semantic network shown in Fig.2.2

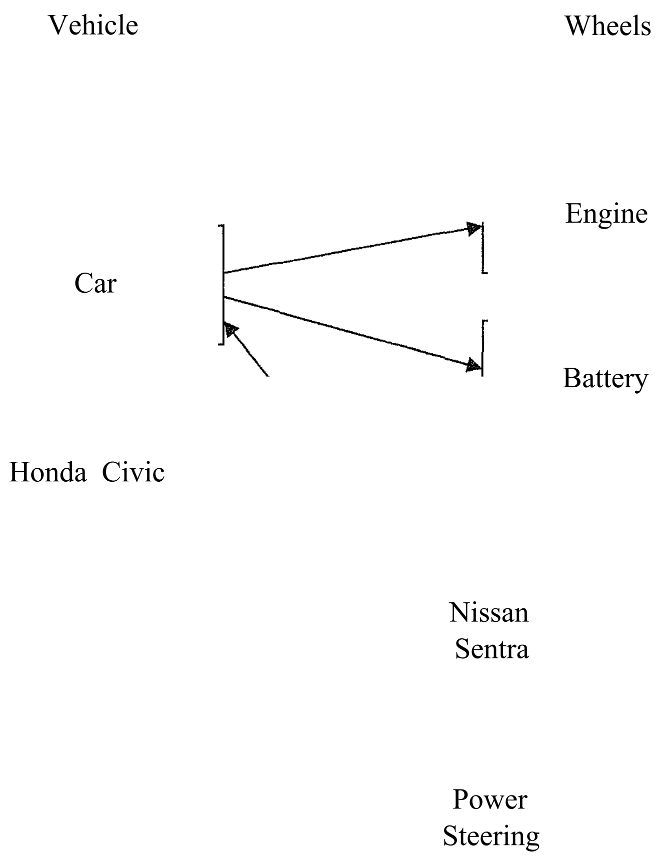


Figure 2.2 An example of A Semantic Network for Car

In this scheme, knowledge is represented in terms of objects and relationships between objects. The objects are denoted as nodes of a graph, with the relationship between two objects being denoted as a link between the corresponding two nodes. The most common form of semantic networks uses the links between nodes to represent IS-A and HAS relationships between objects. In the example shown in Figure 2.2, a car IS-A vehicle; a vehicle HAS wheels. This kind of a relationship establishes an inheritance hierarchy in the network, with objects lowers down in the network inheriting properties from objects higher up.

2.2.2.1 Advantage of Inheritance

There are several advantages of inheritance, as follows:

- It uses less memory because many relationships need not be explicitly stated.
- Inheritance enables the grouping of similar objects.
- Networks are easier to update since one node can inherit relations from ascendant nodes.

2.2.3 Frames

Marvin minsky first conceived the concept of frame in 1975 when he defined a frame as follows: A frame is a structured piece of information about properties, characteristics or features of an object, act or event.

FRAME animal;

NAME	FILLER VALUE	
	IS A	sentient_ being
	NUMBER OF LEGS	default=4
	IS FURRY	default=true

Frames are another perspective scheme of knowledge representation. They have become- popular for AI systems designers very quickly due to their universality and flexibility. A frame serves as a kind of template for holding related cluster of data. This data is usually stored in slots. These slots are simply attributes of that object. For example, the frame for an object called an animal might have the slots shown in Fig. 2.3.

The slot values in a frame can be of two different kinds:

- Named slots in a frame may have filler values that are data items, for example the NUMBER_ OF_LEGS slot. A data slot is used to store items of data, the type of which may be strings, integer, Boolean values, and so on. A slot may also have a default value assigned to it. This is simply a value given to a slot if no value is found for that slot. In the example of animal the default number value would be 4. This is because the expected value for this slot is generally assumed to be 4. In the named slot called IS_FURRY, the default value is assigned to be true. In other word, an animal is assumed to be furry in the event that nothing else is known about the animal. This is a reasonable assumption since most animals have fur and four legs.

- Slot in a frame may also be relations, such as the IS_A slot value in the animal frame above. The IS_A slot value is to link object relationships in a hierarchy. This particular slot is defining a relation between the animal frame and a frame called sentient _being. In this way, a hierarchy of animal objects can be defined using frames.

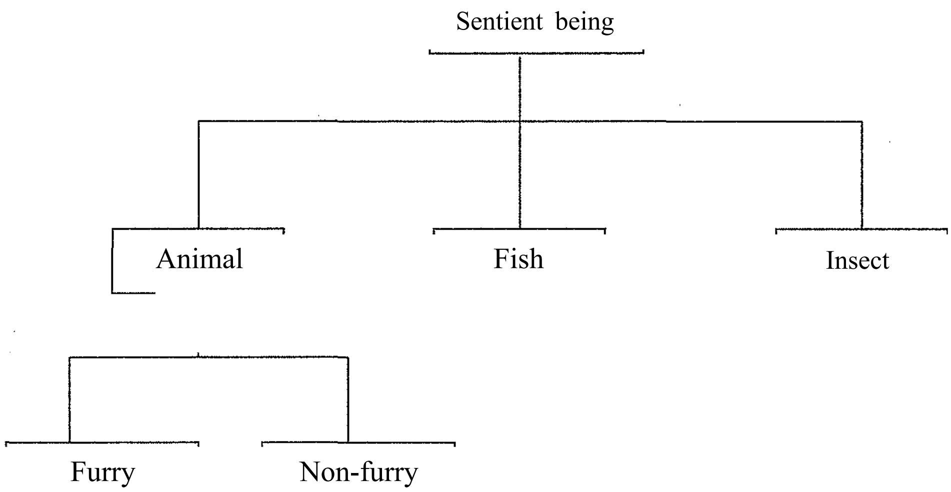


Figure 2.3 Section of Frame Hierarchy

Another example about car frame with another shape:

FRAME car;

(Inheritance slot) IS_A

Value: Car

(Attribute slot) Make

Value: Honda

(Attribute slot) Year

Value: 1989

2.2.4 Logic

Historically, logic has been extensively used in AI programs. The main purpose of logic concerns the soundness or unsoundness of arguments. Typically, an argument consists of statements called propositions, from which other statement(s) called conclusion(s). The main method of knowledge logical representation is using of first order predicate, logic (predicate calculus). In this approach the knowledge about some problem area can be considered as the set of logical formulas. Modification in knowledge representation model happen as a result of adding, replacing of logical formulas.

Logical model representations are easy for understanding and have inference rules necessary for operating on them. However, in logical models of knowledge representation the relations between knowledge elements are expressed by limited set of formal systems used, which don't allow reflecting the problem area specific features perfectly. The tendency to consume large amount of memory is another drawback of logical representation.

2.2.4.1 Propositional Logic

In order to formalize the language of logic, it is necessary to define what is meant by a proposition. A proposition is a sentence that is either true or false. For example, the following are propositions:

"Omar is a happy man" (1)

"All cat are good pets" (2)

Both of the above sentences (1) and (2) are propositions, because each is either true or false. The following phrases not propositions:

"Amy's pet" (3)

"Oh dear me!" (4)

Statements in propositional logic are usually expressed symbolically. For example, the following inference:

"IF Omar is a happy man then Omar is a teacher"

Could be symbolically expressed as:

A- Omar is a happy man

B- Omar is a teacher

This could be expressed in propositional logic as:

IFA THENB

This is written in logic notation as A-B (meaning proposition A implies proposition B). This is an example of a rule of inference called modus pones. Put simply, it says that if proposition A is true, and rule of inference A-+B is true, then B will also be true. Propositions can be combined using logical connectives: for example, as in the statement ("If I listen to radio and the room is warm then I fall asleep").

Rewriting this symbolically:

Let A be the proposition I listen to radio

Let B be the proposition the room is warm

Let c be the proposition I fall sleep

Then this can be written in logic notation as:

AA **B-c** Where this symbol (A) means AND.

Where this symbol (-) means implies

For this purpose the consequents of rules are examine. In order to find such a rule, this would allow achieving the goal. Once such a rule has been found, all its conditions at checked to truth. If conditions are true, the production activated. Otherwise the search of suitable production continued.

2.2.4.2 Predicate Logic

Propositional logic is inadequate for solving some problem because a propositions has to be treated as a single entity that is either true or false. Predicated logic overcomes this by allowing a proposition to be broken down into two components. These are known as arguments and predicates. It also allows the use of variables, in addition to supporting the rules of inference derived from propositional logic (i.e. modus pones etc.).

For example, consider the proposition:

Amy has brown hair

This could be written in predicate logic notation as:

HAS (Amy, brown hair)

In the above example, HAS is called the predicate, and arguments are "Amy" and "brown hair." Now consider the proposition:

BMW is acar

The above can be written in predicate logic as:

IS_A (BMW, car)

2.2.5 **Hybrid Representations**

There is no single representation that will be suitable for representing every problem domain. Each representation schema has advantages and disadvantage. These are summarized in table 2.1 sometimes, a hybrid representation scheme many are better than any single scheme. AI programming tools containing rules, frame and logic are now available that can combine representation schemes in a variety of ways. These tools often work well in practice.

For example, within a frame system, the slots may be made to represent practices and the values and frame names may utilizes the knowledge within the frames. This frame system could also contain production rules that use procedural knowledge and heuristics.

Representation scheme	Advantages	Disadvantages
Production rules	Modular, flexible, and well suited too many domains.	Difficult to represent descriptive knowledge in a natural way. Also difficult to separate domain knowledge and problem and problem-solving knowledge.
Logic	Precision. That is, conclusions guaranteed to be correct if premises correct. Allows programs to be declarative -for example, PROLOG.	Opaque, poor psychological model. Difficult to represent uncertainty.
Semantic Network	Object-based representation; therefore permits inheritance.	Cannot distinguish between the class of an object and a particular object. Unlike frames no

		facility to handle procedural knowledge. Presentation and structure for complex systems could become unmanageable.
Frames	Object-based representation facilities reusability; frames offer facilities for exception handling and defaults, both of which are not easily handled in logic or other representations.	Theoretical difficulties arise from slots in a frame being unrestrained. This means nothing can be certain and so it can be universal definitions to objects.

Table 2.1 Advantages and Disadvantages of Representation Schemes

2.3 The Methods Representing and Structuring Information in ES

2.3.1 Knowledge and Data

Knowledge and data is the information base of intelligent systems. We regard knowledge as the Information, which describe the main regularities of problem area and allow the human to solve certain industrial. Scientific and other problems. Different facts, concepts, interrelations, estimations, rules, heuristics and decision making strategies in this area are examples of knowledge, the kind of information, saved in knowledge bases reflects human- specialist (expert) know-how of certain problem area, and number of all current states of objects, and methods of transitions from one object description to others belong to concept of knowledge in intelligent systems. In other words: "Knowledge = facts +beliefs + rules". There are following types of knowledge: about object and categories of environment; about events, which determine temporal sequences and cause-effect relations about activities, i.e. about capability to execute some actions; met knowledge, i.e. "knowledge about size of our knowledge or about limits of our capabilities" .Knowledge, represented in AI systems incorporate: structure, form characteristics, functions and possible states of object; possible relations between objects, possible events, in which these objects can take part; physical laws; possible effects of actions and states, causes and conditions of occurrence of events and states ; possible intentions, purposes, plans, agreements, etc. Knowledge is characterized by their internal interpretability, structurebtlity, coherence and activeness.

2.3.1.1 The Structurebtlity

The structurebtlity allows representing complex objects in form of a number of more simple ones. For this purpose the relations "part-whole", "class-subclass", "genus-species", etc. are used. In this way the object is identified through the characteristics of components, included in its structure.

2.3.1.2 The Coherence

The coherence of knowledge is determined both by their structure and by regularities, described by this knowledge. The knowledge's in AI systems are active. They include the description of different actions and operations, which must be executed in case of satisfactions of certain conditions. The activity of knowledge is displayed by knowledge generating new knowledge.

2.3.1.3 The Interpretabllity

The Interpretability of knowledge is in the possibility of getting by means of computer the distinct profound answers to any questions, in which the relations, fixed in knowledge bases are explicitly stated. The interpretability of knowledge is provided by representing of each unit of information (slot) in the form of record ((name-label), (value)). The name-label shows the belonging of information unit to the same class or type. For example, the slot ((computer), (IBM PC AT - 386)) allows to establish the fact about a certain model of IBM personal computer.

There are deep and shallow, soft and hard knowledge. Abstractions, images, analogies, in which the understanding of the problem area structure and properties and interrelations of separate concepts are reflected, belong to deep knowledge (in fundamental sciences the laws and theoretical foundations belong to deep knowledge).The shallow knowledge usually concerns only the set of empirical associations and cause-effect relations between the problem area concepts.

The hard knowledge allow to get monosynaptic, clear recommendations under the given conditions, but the soft knowledge permit the multiple, fuzzy solutions, different variants of recommendations. We regard the data as the information, represented in form, suitable for its processing by hardware, perhaps with human participation. The representation is the action, making a concept perceivable by some formalism - by language, symbols, graphical Images, records or others conventional signs.

2.3.2 The Data and Knowledge Consists

The formalization of problem area regularities and objects, characteristics and relations between objects, for their reflection in computer memory. Different models are used for data and knowledge representation. They transform clear representation, created by speech, image, natural (Farsi, Azerbaijan, English etc.) or formal (algebra or logic, reasoning etc.) languages into the form, suitable for putting into computer and processing it.

On computer processing of information, the data a prior get the status of absolute truth. However, in intelligent systems one has to operate with information units, which are not the absolute truth. Information saved in database can reflect subjective experience of experts, result of experimental observations, somebody's own opinion etc. Such a situation forces the slot structure to expand. When slot expands, it gets the following form ((slot name), (slot value), (slot status)). One can call the status assigned to slot the status of truth. In this position of slot could present different linguistic estimations of following types: absolute truth, empirical truth, doubtfully, quite probably etc.

2.4 Knowledge Representation Models

One may conditionally separate the knowledge representation models to declarative and procedural ones.

2.4.1 The Declarative Representation

The declarative model of knowledge representation is based on assumption, that the problem area representation task is performed independently from how this knowledge will be used in future. That is why the model consists of two parts: static declarative structures of knowledge and Inference engine. Inference engine operates by these structures and is practically. Independent of their content. Syntactic and semantic aspects of knowledge are separated to a certain degree - that is an advantage of this representation form, because of the possibility of achieving certain universality.

The declarative models don't contain explicit descriptions of procedures to be executed. Usually these models are sets of assertions. The problem area is represented in form of syntactic description of its state (as much as possible). The Inference is based generally on search procedures in the state space.

2.4.7 Procedural Representation

- In procedural representation the knowledge is kept in procedures-programs, which determine how to perform the specific actions (what to do in specific cases). In this case there's no need for description of all possible states of environment or object for Inference realization. It's enough to save some states and procedures, generating the necessary description of situation and actions.
- In procedural representation the semantic is directly kept in description of knowledge base that increases the efficiency of solutions search. In comparison with procedural part their static knowledge base is small. It contains not "invariable axioms" but "assertions", which are currently acceptable, but may be changed or replaced at any time. The general knowledge and inference rules are represented in form of special goal-directed procedures, activating when it is necessary.
- The procedures can activate one another; their execution can be interrupted, and resumed again. It is possible to use the procedures-"demon3", activating when the operations of inputting, changing or replacing of data are performed.

2.4.2.1 Advantage of Procedural

The main advantage of procedural models of knowledge representation is more efficient inference engine because of using additional knowledge on application that however decreases their universality. Another advantage is their expressiveness. Procedural models can simulate practically any knowledge representation model. The expressiveness of procedural systems is displayed in expanded system of inference, realized in them.

2.5 Summary

As previously stated, no single representation structure is likely to be satisfactory for all type of application domain. Production rules are ideal for capturing the expert's rules of thumb expertise. Logic is often successful in building fairly small, logically consistent knowledge bases, and frames are well suited to structured object representations.

CHAPTER 3.RULES BASED EXPERT SYSTEM

3.1 Overview

A rule is a conditional statement that links given conditions to actions or outcomes. Conventional rule-based expert systems use human expert knowledge to solve real-world problems that normally would require human intelligence. Expert knowledge is often represented in the form of rules or as data within the computer.

Depending upon the problem requirement, these rules and data can be recalled to solve problems. Rule-based expert systems have played an important role in modem intelligent systems and their applications in strategic goal setting, planning, design, scheduling, fault monitoring, diagnosis and so on.

A rule-based expert system is an expert system which works as a production system in which rules encode expert knowledge.

Most expert systems are rule-based. Alternatives are:

- Frame-based - knowledge is associated with the objects of interest and reasoning consists of confirming expectations for slot values. Such systems often include rnles too.
- Model-based, where the entire system models the real world, and this deep knowledge is used to e.g. diagnose equipment malfunctions, by comparing model predicted outcomes with actual observed outcomes
- Case-based - previous examples (cases) of the task and its solution are stored. To solve a new problem the closest matching case is retrieved, and its solution or an adaptation of it is proposed as the solution to the new problem.

3.2 Inference Engine in Rules Based Systems

A rule-based system consists of if-then rules, a bunch of facts, and an interpreter controlling the application of the rules, given the facts. These if-then rule statements are used to formulate the conditional statements that comprise the complete knowledge base.

The purpose of the inference engine is to seek information and relationships from the knowledge base and to provide answers, predictions, and suggestions in the way a human expert would. The inference engine must find the right facts, interpretations, and rules and assemble them correctly. Two types of inference methods are commonly used:

3.2.1 Backward Chaining

In a backward chaining system, the hypothesis (or solution / goal) we are trying to reach is processed first, and keep looking for rules that would allow concluding that hypothesis. Backward chaining is the process of starting with conclusions and working backward to the supporting facts. Use the rules to generate new facts until the desired goal is reached. To backward chain the inference engine must:

1. Select rules with conclusions matching the goal.
2. Replace the goal by the rule's premises. These become sub-goals.
3. Work backwards till all sub-goals are known to be true -either they are facts (in WM) or the user provides the information.

3.2.1.1 Suggesting the Backward Chaining Search

- A goal or hypothesis is given in the problem statement or can be easily formulated (theorem-proving, diagnosis hypothesis testing).
- There are a large number of rules that match the facts, producing a large number of conclusions - choosing a goal prunes the search space.
- Problem data are not given (or easily available) but must be acquired as necessary (e.g. medical tests).

3.2.2 Forward Chaining

In a forward chaining system, the initial facts are processed first, and keep using the rules to draw new conclusions given those facts. Forward chaining is starts with the facts and works forward to the conclusions. To forward chain from a goal in WM the inference engine must:

1. Match the condition patterns of rules against facts in working memory (WM).
2. Ifthere is more than one rule that could be used (that could "fire"), select which one to apply (this is called conflictresolution).
3. Apply the mle, maybe causing new facts to be added to working memory.
4. Halt when some useful (or goal) conclusion is added to WM (or until all possible conclusions have been drawn).

3.2.2.1 Suggesting the Forward Chaining Search

- All or most of the data is given in the problem statement (interpretation problems).
- Large number of potential goals but few achievable in a particular problem instance.
- It is difficult to fomıulate a goal or hypothesis.

Data-driven search can appear aimless but produces all solutions to a problem (if desired), mixed reasoning is also possible - facts get added to the WM and sub-goals get created until all the sub-goals are present as facts.

3.2.3 How Backward and Forward Chaining Works

If we have a simple example for single if-then rule, and explain how will the program work?. Assumes the form 'if xis A then y isB' and the if-part ofthe rule 'xis A' is called the *antecedent* or *premise,* while the then-part of the rule $'y$ is B' is called the *consequent* or *conclusion.* As the processing progresses, new sub goals are also set for validation. Forward chaining systems are primarily data-driven, while backward chaining systems are goal-driven. Consider an example with the following set of *if-then* rules:

Rule 1: If A and C then Y

Rule 2: If A and X then Z

Rule 3: If B then X

Rule 4: If Z then D

If the task is to prove that *D* is true, given *A* and *B* are true. According to *forward chaining,* start with Rule 1 and go on downward till a rule that fires is found. Rule 3 is the only one that fires in the first iteration. After the first iteration, it can be concluded that *A, B,* and *X* are true. The second iteration uses this valuable information. After the second iteration, Rule 2 fires adding *Z* is true, which in turn helps Rule 4 to fire, proving that *D* is true. Forward chaining strategy is especially appropriate in situations where data are expensive to collect, but few in quantity.

However, special care is to be taken when these rules are constructed, with the preconditions specifying as precisely as possible when different rules should fire. In the backward chaining method, processing starts with the desired goal, and then attempts to find evidence for proving the goal.

Returning to the same example, the task to prove that *D* is true would be initiated by first finding a rule that proves *D.* Rule 4 does so, which also provides a sub goal to prove that *Z* is true. Now Rule 2 comes into play, and as it is already known that *A* is true, the new sub goal is to show that Xis true. Rule 3 provides the next sub goal of proving that *B* is true. But that *B* is true is one of the given assertions. Therefore, it could be concluded that Xis true, which implies that *Z* is true, which in tum also implies that *D* is true. Backward chaining is useful in situations where the quantity of data is potentially very large and where some specific characteristic of the system under consideration is of interest. If there is not much knowledge what the conclusion might be, or there is some specific hypothesis to test, forward chaining systems may be inefficient.

In principle, we can use the same set of rules for both forward and backward chaining. In the case of backward chaining, since the main concern is with matching the conclusion of a rule against some goal that is to be proved, the 'then' (consequent) part of the rule is usually not expressed as an action to take but merely as a state, which will be true if the antecedent part(s) are true (Donald, 1986). So, if we had this knowledge base again, let's see how the backward is working:

1 If someone is a third year, then they need a job.

2 If someone is a third year, then they live in.

3 If someone needs a job, they will apply to be an accountant.

And we were to add (John is a third year), the system would do nothing at all. But if we were then to ask the question:

Is there anyone who is going to become an accountant?

The system would try to answer. It would begin by searching either for a fact that gives the answer directly, or for a rule by which the answer could be inferred. To find such a rule, it searches the entire knowledge base for rules whose conclusions, if made true, will answer the question.

In this example, there are no facts giving the answer; there's one rule whose conclusion, if true, would supply an answer, and that's rule 3. The system next checks the rule's conditions. Is there anyone who needs a job? As with the original question, we look either for a fact that answers directly, or for a rule. There are no facts, but rule 1 is relevant. So we now check its conditions. Is there a third year? This time, there is a fact that answers this: John is a third year. So we've proved rule 1, and that's proved rule 3, and that's answered the question. But if we look to see how forward chaining works, consider a system with three rules:

1 If someone is a third year, then they need a job.

2 If someone is a third year, then they live in.

3 If someone needs a job, they will become an accountant.

And suppose we put the following fact in working memory:

(John is a third year).

Being a forward chaining, the system is constantly on watch for new data. As soon as this data arrives, the system searches all the rules for any whose conditions weren't true before but are now. It then adds their conclusions to working memory (WM); In this case, rules 1 and 2 have conditions which match this new fact. So the system will immediately create and add the two facts below.

(John needs a job).

(John lives in).

These facts in turn can trigger rules. As each arrives, the system would look for yet more rules that are made true. In this case, the fact John needs a job would trigger rule 3, resulting in the addition of another fact to WM:

John will become an accountant.

The fact John lives in would not trigger anything else though.

3.3 Explanation Facility in Expert Systems

Rule-based systems can be designed to answer questions like:

WHY do you want to know this fact? (i.e. where is the reasoning going?)

HOW did you deduce this fact? (i.e. how did we get here?)

Explanation facilities are useful for debugging a rule base but also for instilling confidence in users of the ES. A tracer module records the rules that have been used. To answer HOW questions these rules are searched for one where the fact in question is a consequent (action/then part). The production system architecture provides the essential basis for explanation facility, and the facility contributes to the success and popularity of rule-based ES.

3.4 Summary

Backward chaining is an efficient way to solve problems that can be modeled as "structured selection" problems. That is, the aim of the system is to pick the best choice from many enumerated possibilities and forward chaining, uses rules similar to those used for backward chaining; however, the inference process is different. The system keeps track of the current state of problem·solution and looks for rules which will move that state closer to a final solution for forward. But the system is to pick the best choice from many enumerated possibilities for backward.

CHAPTER 4.VP-EXPERT SYSTEM

4.1 Overview

VP-expert is the expert system shell we will be using for this course. Like any shell, it contains everything needed for running the expert system (except for the knowledge base of rules for the particular domain). This includes:

- An inference engine for consulting the KB in order to answer queries.
- An editor for creating writing the rules of the knowledge base.
- A user interface capable of handling queries, asking the user questions, and presenting traces and explanations where needed. It also has limited graphical capabilities.

4.2 Running VP-Expert

Their ways to use VP-expeii program one of them install VP-Expert on your own computer, or use your disk in one of the PCs. In either case, VP-Expert is designed to be run from DOS. To run it, go to the disk and/or directory containing the files and type VPX. The window of VP-expert system program that show below in Figure 4.1.

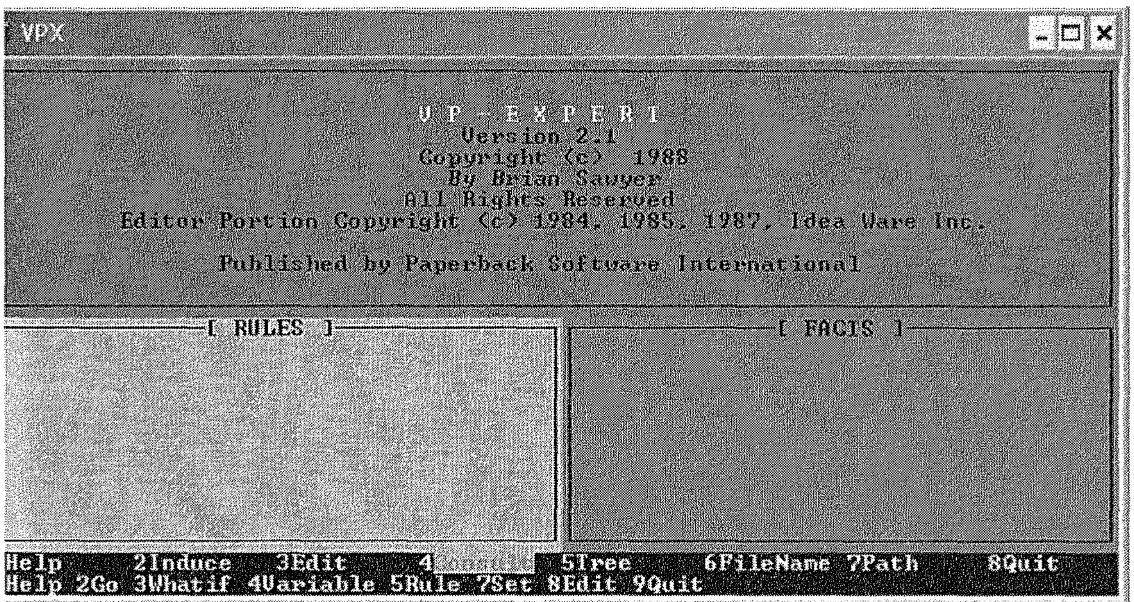


Figure 4.1 VPX Expert System

4.3 The VP-Expert Main Menu

Options from any menu can be selected by using the arrow keys, pressing a function key, a digit, or using the first letter of the option name. The line below the menu also shows the sub-menu (if any) of the option currently highlighted. Important options from the main menu are:

Edit	built-in editor for creating and modifying a knowledge base
Consult	execute the expert system on the current knowledge base
Filename	select another knowledge base (from the current directory) for editing or consulting
Path	change the current drive or directory from which knowledge bases are selected
Quit	Exit VP-Expert

Table 4.1 Components VP-Expert System

The Escape key can always be used to go back a step, in particular, to 'escape' an option that has been selected. A consultation can be aborted by pressing Ctrl-C.
If no file is preselected (using the Filename option) both Edit and Consult will prompt for the file name. The user can than type a file name or select a file from the displayed list. For Edit, the file name may be new, if a new file is to be created (VP-Expeıi will prompt you to enter the new filename).

4.3.1 Running Consultation

The process of using VP-Expert to solve a specific problem, based on the rules in its knowledge base, is called a "consultation." In general, the user does not put queries directly to the system (this is done in the ACTIONS section of the knowledge base), but doe,~ enter answers to questions relevant to the query. To start a consultation, choose Consult from the main menu, and then choose which FileName do u wants it. After that you choose Go to run the program (e.g. choose from filename: ALADDIN.kbs), as show below in Figure 4.2.

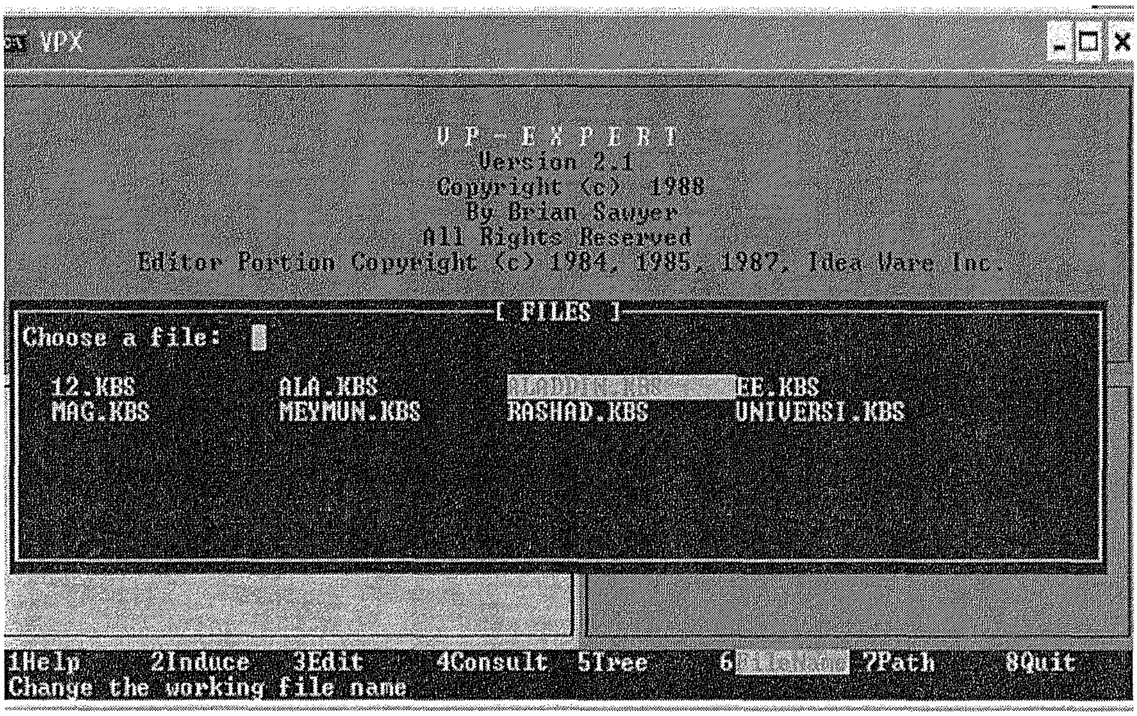


Figure 4.2 VP-Experts with Filename Choosing

The consultation screen contains three windows that show below in Figure 4.3.

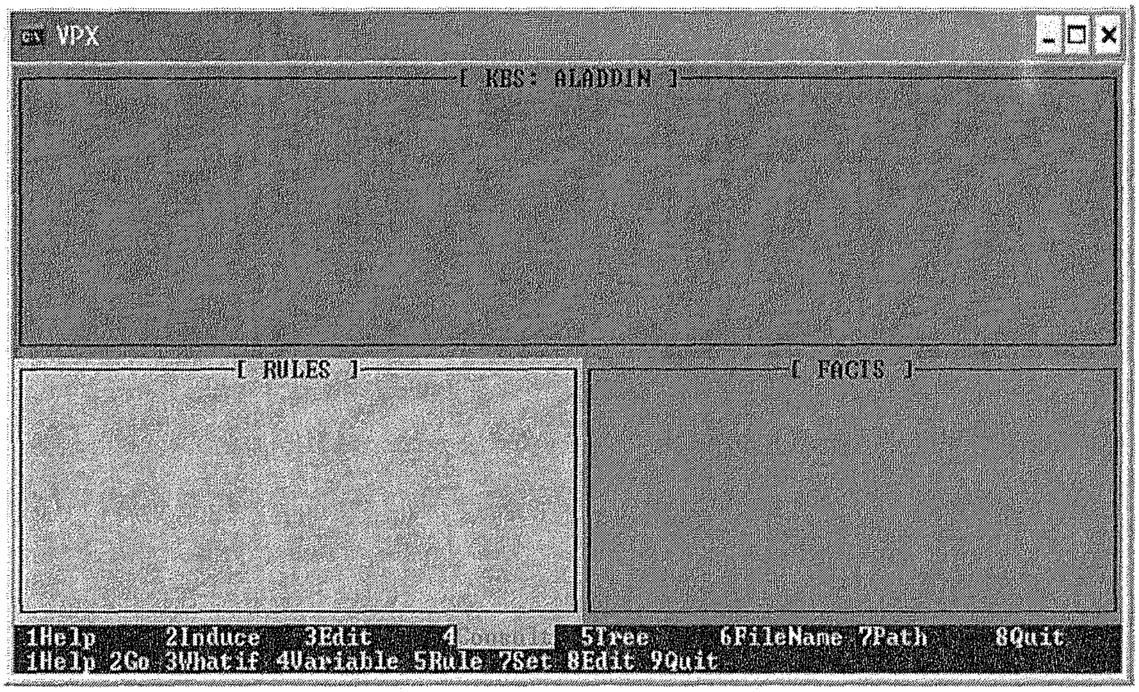


Figure 4.3 Consultation Windows in VPX Program

- The top half of the screen is the interaction window. Input is entered here by the user, and results (or anything else the program prints out) are displayed here.
- The bottom left window shows the rules that are currently being checked. If the consultation speed is slowed, one can follow the sequence of rules being activated.
- The bottom right window shows the values that have been assigned to variables (that is, what facts the system knows so far).

4.3.1.1 Inputting Data to a Consultation

In order to answer a query, VP-Expert uses backward chaining to work from a goal to questions (a question being defined as any proposition which does not appear in the THEN part of any rule). When an answer to a question is needed, the question is presented to the user, possibly along with a menu of possible answers.

- To select an answer from such a menu of values, use the arrow keys to highlight your choice and then press Enter then press End.
- To 'unchoose' a selected value, highlight the chosen value and press Del.
- Finalize your choice(s) by pressing END.
- If the requested input is a number (that is, no menu is presented), type the number and press Enter instead.

4.3.1.2 Extracting Explanations for a Consultation

VP-Expert also allows you to see explanations of how a variable was set, or why a question was asked. This can be done when the program is running by pressing the **/** key (which halts execution temporarily) and then selecting the desired option from the menu. As show as below in Figure 4.4.

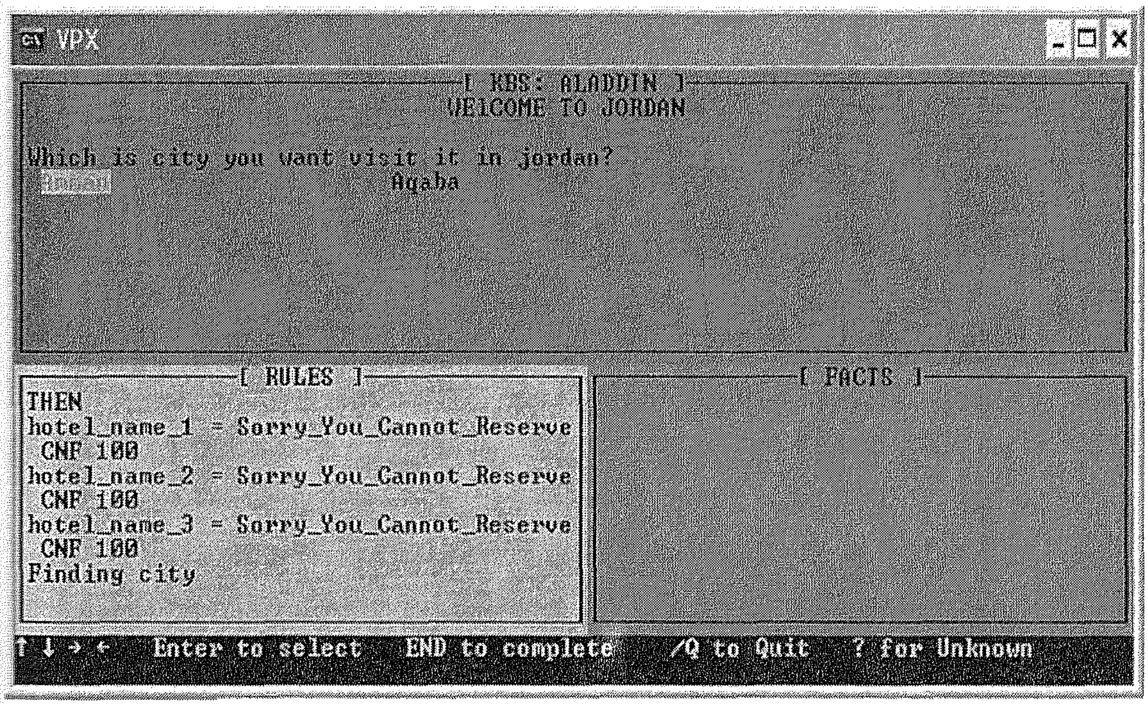


Figure 4.4 Extraction Explanations of a Consultation

4.3.1.3 Performing a Trace of a Consultation

You can also save a copy (called a *trace)* of the inference tree created in the course of a consultation. This can allow you to examine in detail the reasoning process of the system, and is particularly useful for debugging.

To perfomı a trace:

- Choose Set and then trace from the Consult menu.
- Return to the Consult menu, and Run the consultation as you would normally do.
- A record of the rules consulted and facts found during the consultation will now be stored in some file filenaine.trc, where filename is the name of the knowledge base being consulted.

You can view or print it from your own editor, or you can view it directly from VP-Expert by choosing Tree and Text from the main menu.

4.4 Creating a Knowledge Base in VP-Expert

4.4.1 Basic Structure of a Knowledge Base

Creating an expert system with VP Expert is basically a question of entering a knowledge base which consists of three parts:

ACTIONS

RULES

QUERY STAEMENTS

At the center of the process are variables, which take the place of propositions in VP-Expert. The main goal of the inference engine is usually to find a Value for some "goal" variable(s), using backward chaining to each for rules that would assign a value to that variable as part of its consequent.

Note that you can create comments using the! Symbol; the rest of the line after the! Will be ignored by VP-Expert:

! This is a comment

As with any other program, you are strongly encouraged to comment your code, particularly to explain the reasoning and / or source of your rules.

4.4.1.1 The ACTIONS Block

The ACTIONS block consists of statements which control the actions of the shell. These statements are executed in the order in which they appear (the only part of the KB that does). In effect, the ACTIONS block is the 'code' which controls the execution of the inference engine.

The following is an example of a (minimal) ACTIONS block:

```
ACTIONS
DISPLAY "Welcome to Jordan."
FIND hotel_name_1.
DISPLAY "Author: Eng. ALAddin ALsoweity."
```

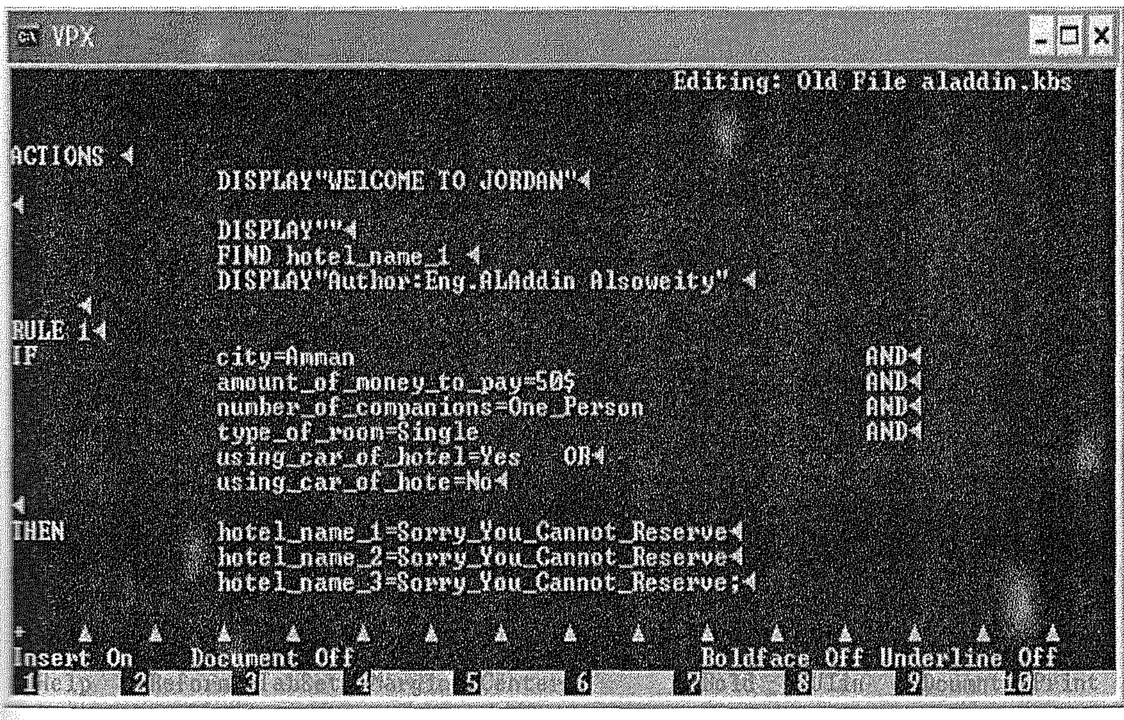


Figure 4.5 The action Block

Note that it begins with the word ACTIONS and terminates with a semicolon (no semicolons between statements). The first DISPLAY statement tells the user what to do. The FIND statement expresses the system's goal. The last statement DISPLAYS the results. Details of each of these statement types are discussed below.

4.4.1.1.1 The FIND Statement

The basic form of this statement is:

FIND *variable*

This statement activates the inference engine, causing it to consult the knowledge base of nıles until a value is found for the variable. This follows the backward chaining described in class:

- It searches for the first nıle which would give the variable a value (as part of the THEN part of the nıle), and then attempts to FIND values for variables in the IF part of the nıle.
- It halts when any value is found for the variable (unless it is a PLURAL variable).
- If no nıles are found which contain the variable in its THEN part, the shell instead As ks the user for its value.
- If there exist nıles which contain the variable in its THEN part, but none of them can be proven, then the variable is considered to have value unknown.

Note that it is possible to have more than one FIND statement in the ACTIONS block -- however, it is rare that we actually do so. It is usually more efficient (usually less questions are asked) to have a single FIND to begin the consulting process, and to use rules to make sure that other values are only found when needed.

4.4.1.1.2 The DISPLAY Statement

Any desired text may be displayed on the screen using a DISPLAY statement and enclosing the text in double quotation marks:

DISPLAY "This is some text to be displayed for the user - perhaps to give instructions on how to use the system".

If you want to make sure that the user has time to read a message before it disappears from the screen, insert a "-" as the last character in the message. This suspends all processing until the user presses some key on the keyboard. The "-" itself is not displayed. Be sure to tell the user how to make the system resume execution:

DISPLAY "Here are some instructions. Press any key to continue-",

The value of any variable may be included in a displayed message simply by enclosing the variable name in curly brackets:

DISPLAY "I prescribe {treatment} for your {diagnosis}."

Note that you can also put DISPLAY statements in the THEN or ESLE part of a rule. This can be particularly useful for providing explanations, or for displaying information only if a certain condition holds.

4.4.1.2 Production Rules in VP-Expert

The production rules of an expert system contain its domain knowledge, expressed as IF-THEN type rules. Unlike the statements in the ACTIONS section, they are not run in the order listed; instead they are consulted as needed during the process of backward chaining. The order of rules is only important when there is more than one rule that can be used to give a variable a value, as the inference engine attempt to prove the rules in the order given (see below for an example).

4.4.1.2.1 Basic Form of Rules

The basic form of a rule is as follows:

RULE *rulename*
IF *antecedent*
THEN *consequent;*

Every rule must have a unique name (up to 40 chars in length) after the word RULE (leaving this out is a very common source of syntax errors). Like any other label you use in programming, it should be descriptive of what the rule means. I will give example (very simple) rule:

RULE Diagnosis_of_measles
IF Diagnosis = measles
THEN Treatment = penicillin;

In this case, Diagnosis and Treatment are variables, and measles and penicillin are potential values for those variables. In other words, this rule states that if Diagnosis has the value measles, we may then assign Treatment the value penicillin. If the inference engine was currently trying to use this rule to FIND a value for the variable Treatment, it would:

- Check whether Diagnosis had already been assigned a value.
- If not, it would use the rules to try to FIND a value for Diagnosis, returning to this rule once it has done so.
- If the value of Diagnosis is measles, then Treatment is assigned the value penicillin. At that point, the inference engine does no more work n finding a value for Treatment.
- If Diagnosis has some other value (including unlaıown), then the rule fails, and the inference engine must search for some other rule to give Treatment a value (if there is none, then Treatment is assigned the value unlaıown).

4.4.1.2.2 Relational Operators in Rule Antecedents

The IF part of a rule consists of one (or more) simple premises of the form:

variable relational_operator value

Where relational operator is one of the following:

	is equal to
<>	is not equal to (used instead of NOT by VP-Expert)
<	is less than
<=	is less than or equal to
>	is greater than
>=	is greater than or equal to

4.4.1.2.3 The ELSE Statement

Rules may also contain an ELSE statement as part of their consequent. For example:

```
RULE Diagnosis_of_flu
IF   Diagnosis = flu
THEN Treatment = bed rest
ELSE Treatment=  get_back_to_work;
```

If the variable Treatment has been assigned no other value by this point in the KB (by any rule, not just the THEN part of this rule), it is assigned the value given by the ELSE.

4.4. 1.2.4 Logical Operators

The IF part of a rule can be composed of up to 20 simple premises connected by the logical operators OR or AND.

For example:

```
RULE measles
IF   temperature= very_high
AND  spots = yes
AND innoculated <> yes
THEN Diagnosis = measles;
```

If VP-Expert is attempting to use this rule to find a value for Diagnosis, it will first check whether temperature is very high (finding a value for temperature if necessary), then whether spots is yes, and so on. For efficiency, VP-Expert gives up on such a rule immediately if it cannot prove one of the premises. For example, if temperature is *not* very_high, it does not try to find spots or innoculated.

OR is handled differently. For example, consider:

```
RULE allergy
IF spots = yes
OR rash=  yes
THEN Diagnosis=  allergy;
```

Even if VP-Expert can prove spots = yes, it will still ask about rash (even though it doesn't need to). To decrease the number of questions asked, you would have to write two separate rules:

```
RULE allergy!
IF spots = yes
THEN Diagnosis = allergy;
RULE allergy2
IF rash= yes
THEN Diagnosis = allergy;
```

Note also that OR has higher precedence than AND. For example, the rule:

```
RULE flu
IF   throat = sore
AND  temperature = high
OR   temperature= very_high
THEN Diagnosis = flu;
```

Is interpreted as (throat= sore) AND (temperature= high OR temperature= very high).

4.4.1.2.5 Numeric Assignment and Evaluation

As with other languages, VP-Expert also allows you to store numeric values in variables, and to manipulate them with arithmetic expressions. Since VP-Expert is more concerned with logic, however, the syntax for doing so is rather limited.

4.4.1.2.6 Variables

Numeric computations usually include *variables* on the right side of the operator in assignment or conditional statements (we have only seen *values* up to this point). However, since VP-Expert does not allow you to declare your variables, it needs some way to tell a variable (such as diagnosis) from a value (such as measles) when used on the right side of the experession. This is done by enclosing the variable on the right side of the expression in parentheses. For example:

```
RULE own_pharmacy
IF   pharmacy _used = ours
THEN perscription = (treatment);
```

4.4.1.2. 7 Computation

Arithmetic expressions are also enclosed in parentheses, and use most of the same syntax and operators as most programming languages:

```
RULE profit
IF   income > (expenses)  ! since expenses is a variable, it must be in parens
THEN profit= (income - expenses)
ELSE deficit= (expenses - income);
```

Note that VP-Expert only tries to FIND variables that are on the left side of an expression; for example, the above rule will try to FIND a value for income but not for expenses. If a value has not already been found for expenses, VP-Expert will treat it as "unknown", giving unpredictable results.

One way to fix this is to include a "dummy" condition in the rule to force it to FIND a value for expenses:

```
RULE profit
IF   expenses >= 0  ! hopefully this will always be true!
AND  income > (expenses)  ! will work, since expenses now found
THEN profit= (income - expenses)
ELSE deficit= (expenses - income);
```

4.4.1.2.8 Multiple Answers to Queries

As stated above, the inference engine for VP-Expert will stop once it has found the first value of a variable it is trying to FIND. It is possible, however, to force it to find *all* values for variables instead. This is done with the PLURAL statement (which must be placed before the ACTIONS block):

PLURAL: *list of variables;*

While you don't have to make all of the variables in your knowledge base PLURAL, you must at least include all that would support the kind of multiple results you wish to obtain. In the "medical" knowledge base we have been working with so far, for example, if you wish to make treatment plural, then you must make diagnosis plural as well:

PLURAL: treatment, diagnosis;

You may also have to change the structure of your rules that deal with PLURAL variables. In particular, you cannot count on the "order of evaluation" when structuring your rules; nor can you use ELSE to set the value of a PLURAL variable, as it would always be used.

For example, if the variable cost were PLURAL, we could not use the following rules:

```
RULE cheap
IF price< 20
THEN cost = cheap;

RULE expensive
IF price < 100
THEN cost = medium
ELSE cost = expensive;
```

If price had value 10, then cost would be set to cheap, medium, and expensive! In general, such rules must be set up so that they are *independent* of the order in which they are tried:

```
RULE cheap
IF price< 20
THEN cost = cheap;

RULE medium
IF price < 100
AND price >= 20
THEN cost= medium;

RULE expensive
IF price >= 100
THEN cost = expensive;
```

Another way to deal with an ELSE is to replace it with a check for the value unknown:

```
RULE work
IF diagnosis = unknown
THEN treatment= get_back_to_work;
```

This rule will fire only if diagnosis has been given *no* value by this point. Like the ELSE, it should be the last rule involving treatment for this to work correctly.

4.4.1.2.8.1 PLURAL Inputs

It is also possible to set up your ASK statements to allow the user to choose more than one option from the menu by making that variable PLURAL. This is often faster for the user than having to answer multiple questions. For example:

```
PLURAL: symptoms;

ASK symptoms: "Choose all symptoms that you have.";
CHOICES symptoms: sore_throat, spots, rash, none;
```

The user can then press ENTER by each one which holds, and press END to confirm the list.

4.4.1.2.8.2 Printing Variables With Multiple Values

The DISPLAY statement works with PLURAL variables, printing all of its values in column format. For example, the code:

```
DISPLAY "I prescribe the following:"
DISPLAY "{treatment}"
```

might print:

```
I prescribe the following:
penicillin
allergy pills
bed rest
```

4.4.1.3 Query Statements

Variables which do not appear as the consequent of some rule in the knowledge base (that is, are *leafs* in the deduction tree), are considered potential *questions* for the user. If the inference engine attempts to FIND such a variable, the user will be *prompted* for its value. This is done with ASK and CHOICES statements.

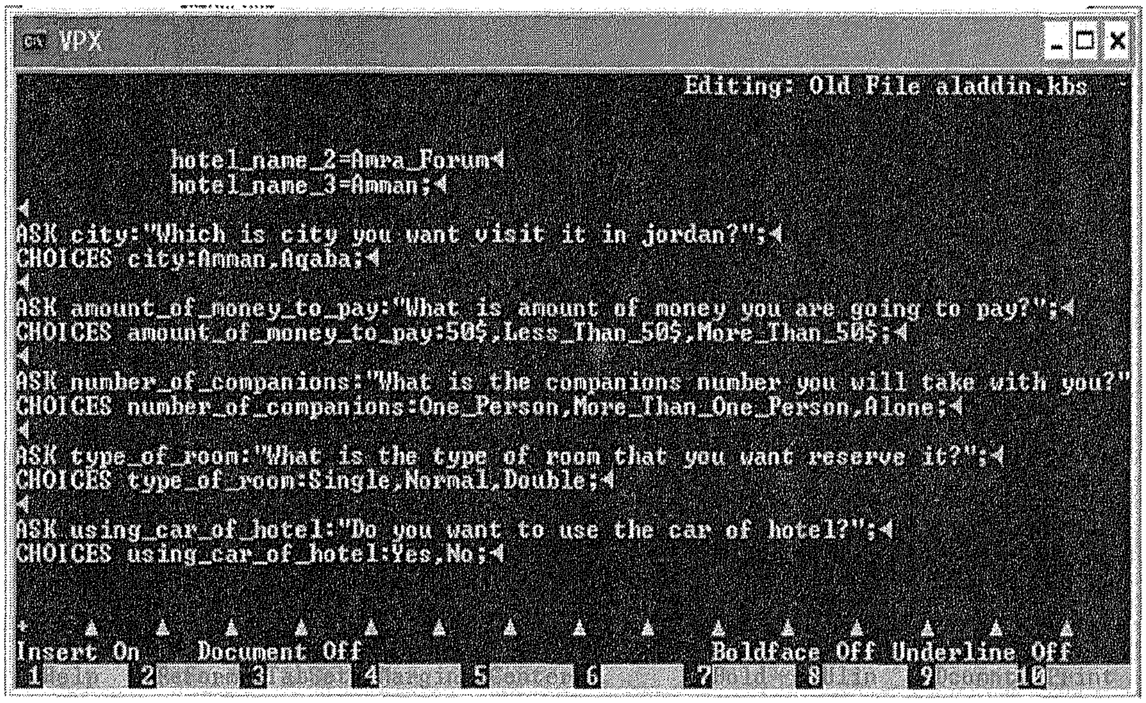


Figure 4.6 Query Statements

4.4.1.3.1 The ASK Statement

The form of the prompt for a variable is defined by an ASK statement. It has the form:

ASK *variable:* "prompt";

For example from my program (aladdin.kbs):

ASK city: "Which is city you want visit it in Jordan?";

As with any other program, these prompts should be informative - that is, they should tell the user how to gather and enter the information.

4.4.1.3.2 The CHOICES Statement

If there are a finite number of possible answers to a question they may be presented in a menu defined by a CHOICES statement. It has the form:

CHOICES *variable: list of values;*

For example from the same file (aladdin.kbs):

CHOICES city: Amman, Aqaba;

This menu is printed out when the question is Asked. Note that if there is no CHOICES statement for a variable, the user will have to type in a value at the cursor after the prompt (this is how numeric variables are usually handled).

4.5 The VP-Expert Editor

VP-Expert includes a simple, but sufficient text editor that can be invoked from the Main Menu and the Consult Menu. It is also invoked automatically when a syntax error is detected in the knowledge base. If invoked from the Main Menu, the user will be prompted for the name of the file to be edited. The user can either type a name or select from the displayed list. A new file can be specified by typing a new name.

Because the editor is rather non-intuitive to use (and because it sometimes has problems loading in rooms 301 and 303), you are encourged to use whatever editor you prefer. Just make sure that the file you create ends in the extension .kbs Once invoked, the VP-Expert editor acts like a normal text editor in insert mode, "pushing over" any text to the right of the cursor. The only exeption is the RETURN keyi, which just moves you down to the next line. To insert a new line, you must type Control-Enter.

4.5.1 The Editor Commands Menu

When in edit mode, the bottom of the screen displays a list of editor commands that can be invoked using the function keys. Note that the choices in the list change when the Alt or Ctrl keys are held down. This show the commands which can be invoked by combining one of these keys with the appropriate function key. The following table lists the most conunonly used editor commands:

ALT-F6	Save the current file and exit the editor
ALT-F8	Exit the editor without saving
ALT-F5	Save the current file without exiting
Control-Enter	Insert a new line
Delete	Delete the character at the cursor position
Backspace	Delete the character to the left of the cursor position
Control-T	Delete from cursor to end of word
Control-Y	Delete this line
Page Up	Up one screen
Page Down	Down one screen
Home	Move to beginning of line
End	Move to end of line

Table 4.2 Editor Commands

CHAPTER 5.HOTEL SYSTEM BY VPX PROGRAM

5.1 Overview

The expert systems it is possible using in the hotel system, so the human found VPX system and another program to solve the problems from through put informations (knowledge base) by experts human and from these informations it can solve the problems and give the results. And keep the experts system instead the human to easy work and reduce the time, and more things I explained it in chapter one.

In this chapter it will detail my program about the hotel system, and what include from the informations in this program. The first thing this program about hotel system in Jordan, using the VPX program, that's include 98 rules, these rules used by production rules (IF... THEN).and each rule statement include five conditions about (city, amount of money to pay, number of companions, type of room, and using car of hotel. And I used many names hotels, for each from them; I putted three hotels after THEN action to show in the facts.

When running my program (ALADDIN.kbs), it will be show for the user screen includes five questions (see chapter four), these questions putted from through the experts human, from what informations putted the program it will work. From what the user chooses, the program will be show for three hotels names in the city, for each city (Amman, Aqaba) its different names hotels, or maybe have the same names hotel in both city, or different system.

5.2 IF THEN Statement Using

IF THEN statements include down the production rule, that's type of representation methods, this type it used in my program, see chapter two for IF ... THEN Statement.

5.2.1 IF Statement

In the IF statements I have five conditions, lets to see them:

- In the first condition two cities:
  - ./ Amman
  - ./ Aqaba

- In the second condition three types for the amount of money to pay:
  - ./ 50$
  - ./ Less than _50$
  - ./ More_ than_50$

- In the third condition three types for number of companions:
  - ./ One_ person
  - ./ Alone
  - ./ More_ than_ one_ person

- In the fourth condition three types for type of room:
  - ./ Single
  - ./ Normal
  - ./ Double

- In the fifth condition two type for using car of hotel:
  - ./ Yes
  - ./ No

5.2.2 The Action (THEN) Statement

The action statement comes after IF statement, that's operate if the condition is true then the action will be operate this rule. So if the all informations in one rule are true so the action it will be true. I used in the action statement three types that are:

- ../ hotel name 1
- ../ hotel name 2
- ../ hotel name 3

For each one from the rule will be display different names of hotels, for two cities (Amman, Aqaba), that it will be different hotels names, and there are in some rules the same hotels names in the both of these cities. And some hotels names they repeat in different rules that because in the hotels we can find there are two hotels or more than the same system. And also I used one-hundred thirty one hotels (131 hotels) the total of hotels.

5.3 The Questions

There are five questions that are putted in program after the rules. These questions display for the user when open the program, choose (ALADDIN.kbs) filename .what is these questions:

- ../ Which is city you want to visit it in Jordan?.
- ../ What is amount of money you are going to pay?.
- ../ What is companions number you will take with you?.
- ../ What is the type ofroom that you want reserve it?.
- ../ Do you want use the car of hotel?.

5.4 How the **Program Work**

When you open the program and from filename, choose (ALADDIN.kbs) file, to want know the hotels names are suitable for you according to what you own from money, so how the program will be work when you (the user) choose from multi choose, let's see what will be happen:

- When display the first question (which is city you want to visit it in Jordan) for you, and chose one of them, assume you chose Amman city and you pressed enter then end, so the program it will find the city is Amman, and it will keep it
- In the second question (what is amount of money you are going to pay), assume you chose 50$ and you pressed enter then end, so the program will be search to find what you chose, after when the program has found the aim, it will be keep it in the memory working (fact)

Until these steps the program has two aims:

The first aim the city is Amman.

The second aim the amount of money to pay equal 50$.

- In the third question (what is companions number you will take with you), assume you chose the One Person and you pressed enter then end, the program it will search the aim, after it found it, it will be keep this aim in memory working (fact).
- In the fourth question (what is the type or room that you want reserve it), assume when you chose Nonnal room and pressed enter then end, the program it will search about the aim. After when catch the aim, the program will be keep the aims in memory working (fact).
- In the fifth question (using car of hotel), assume you chose Yes from multi-choose and pressed enter then end, the program will be start to find the aim where is it. After catch the aim it will keep it in memory working (fact).

After all the question the program will be collect the aims to find which rule have these aim, after it found the rule will be display the three hotels names in fact. But there are some things are important. Assume if you chose Amman city or Aqaba and whether are 50$, Less_than_50$, or More_than_50$, an you chose just for One_Person or More_than_One_Person with type of room Single whether is Yes or No.for these choose it will not give for you hotels names, but will display for you Sorry_You cannot_reserve. These informations putted in knowledge base because it is not possible to reserve for One_Person, and More_than_One_Person, in Single room. All code you can read it in APPENDIX paper.

CONCLUSION

At the end of my project, the development of expert system for hotel system ,that could be solve the problem to give the fact to the user ,more easy, and quick to find the hotels names for the user. This program it will give names of hotel for the user which that is better for him when the user will be chose from the questions. This program executed by VPX-system.

The Expert systems have been used to solve a wide range of problems in domains such as business, medicine, mathematics, engineering, geology, computer science, law, defense and education. The users have been used to solve problems of different types. Types of problem involve diagnosis (e.g., of a system fault, disease or student error); design (ofa computer systems, hotel etc); and interpretation (of, for example, geological data). The appropriate problem solving technique tends to depend more on the problem type than on the domain.

The role of expert system in almost every part of human's life is getting bigger. This is because of the performance the expert system has shown of what it can do to help the human beings. People's life seems to be easier in many ways by using expert system. In this case the expert system may be useful to execute the routine works and let the human expert to do the rest especially the more difficult jobs.

REFERENCES

[1] Implementations & Uses, 1986, Chichester, England: Ellis Horwood Ltd. • Brule, James F. Artificial.

[2] Intelligence: Theory, Logic and Application, 1986, Blue Ridge Summit, PA: TAB Books. •Edwards.

[3] Introduction to Expert Systems, 1988, Santa Cruz, CA: Mitchell Publishing, Inc. •Michaelsen, Robert H.

[4] The basic of Expert (Knowledge Based) systems.

[5] What is a Knowledge Representation? *AI Magazine,* 14(1):17-33, 1993. R. Davis, H. Shrobe, and P. Szolovits.

[6] The Essence of Expert System, 2000, South Bank University, Keith Darlington.

[7] http://www.aaai.org/ AITopics/html/ expert.html

[8] http://en.wikipedia.org/wiki/Expert_ System

[9] http://www.macs.hw.ac. uk/e-alison/ai3notes/ chapter2 _5 .html

[10] http://www.pcai.com/web/ai_info/expert_systems.html

[11] http://www.hf.uio.no/iakh/forskning/sarc/iakh/lithic/expsys.html

[12] http://www.geocities.com/Silicon Valley/Lakes/6007/Expert.htm

[13] http://www.wtec.org/loyola/kb/cl_sl.htm

[14] http://www.cs.nott.ac.uk/~sxp/ES3/index.htm

APPENDIX

This all codes for my program:

```
ACTIONS
        DISPLAY"                    WElCOME  TO JORDAN          fl

        DISPLAY""
        FIND hotel  name  1

DISPLAY"****************************************************************
****************"
        DISPLA Y"Author:Eng.ALAddin   Alsoweity"
        DISPLA Y"Title:Hotel   System"
        DISPLA Y"Address:Jordan/Irbid"
        DISPLA Y"Tel:( +962)27100187"
        DISPLAY"E_mail:aladdin_alsoweity@yahoo.com"
        DISPLAY"Year:2006";


RULE 1
IF       city=Amman                                              AND
         amount_ of_money _to _pay=50$                           AND
         number_ of_companions=One _Person                       AND
         type_ of_room=Single                                    AND
         using_car_of_hotel=Yes                 OR
         using_ car_of_hote= No

THEN     hotel_ name_ I=Sorry_ You_Cannot_ Reserve
         hotel_name_ 2=Sorry _You_ Cannot_ Reserve
         hotel_ name_ 3=Sorry _You_ Cannot_ Reserve;


RULE2
IF       city=Amman                                              AND
         amount_of_money_to_pay=50$                              AND
         number_ of_companions=One _Person                       AND
         type_ of_room=Normal                                    AND
         using_ car_of_hotel= Yes

THEN     hotel, name_ 1=Merryland
         hotel_name_ 2=Nairoukh
         hotel_ name_ 3=Qasr _Albalka;
```

RULE3
IF city=Amman AND
amount_ of_ money_ to_pay=50$ AND
number_ of_ companions=One _Person AND
type_ of_room=Normal AND
using_ car_ of_ hotel= No

THEN hotel _ name _ 1=Ramallah
hotel _ name_ 2=Razan
hotel_ name_ 3=Royal;

RULE4
IF city=Amman AND
amount_ of_money_ to_pay=50$ AND
number _of_companions=One _Person AND
type_ of_room=Double AND
using_ car_of_hotel= Yes

THEN hotel _ name_ 1=Cameo
hotel_ name_ 2=Canary
hotel_ name_ 3=Dove;

RULES
IF city=Amman AND
amount_ of_money _to_pay=50$ AND
number_ of_companions=One _Person AND
type_ of_room=Double AND
using_ car_of_hotel= No

THEN hotel _ name_ l=Hala_ Inn
hotel_ name_ 2=Holy _land
hotel_name _3=Rum _Continental;

RULE6
IF city=Amman AND
amount_ of_money_ to_pay=50$ AND
number_ of_companions=More _Than_ One_Person AND
type_ of_room=Single AND
using_car_of_hotel=Yes OR
using_car_of_hotel=No

THEN
hotel_ name_ 1=Sorry_ You_ Cannot_ Reserve
hotel_ name_ 2=Sorry _You_ Cannot_ Reserve
hotel_ name _3=Sorry _You_ Cannot_ Reserve;

RULE7

IF city=Amman AND
amount_ of_ money_ to_pay=50$ AND
number_ of_ companions= More_ Than_ One _Person AND
type_ of_ room=Normal AND
using_ car_ of_ hotel=Yes

THEN hotel _name _ 1=Soladin
hotel_ name_2=Sandy _Palace
hotel_ name_3=Saveen;

RULES

IF city=Amman AND
amount_of_rnoney_to_pay=50$ AND
nuınber_of_companions=More_Than_One_Person AND
type_ of_room= Norrnal AND
using_ car_of_hotel=No

THEN hotel_ name_ 1=Select
hotel_ name_ 2=Sultan
hotel_ name_ 3=Merryland;

RULE9

IF city=Aqaba AND
amount_ of_money _to_pay=50$ AND
nuınber _of_companions=One _person AND
type_ of_room=Single AND
using_car_of_hotel=Yes OR
using_ car_of-hotel=No

THEN hotel_name_1=Sorry_ You_Cannot_reserve
hotel_ name_ 2=Sorry _You_ Cannot_ Reserve
hotel_ name_3=Sorry _You_ Cannot_ Reserve;

RULE 10

IF city=Aqaba AND
aınount_ of_money_ to_pay=50$ AND

number_ of_ companions=One _Person AND
type_ of_room= Normal AND
using_ car_ of_ hotel= Yes

THEN hotel_ name_ 1=Crystal
hotel_ name_ 2=holiday _Inn
hotel_ name _3=Movenpick _Resort_ Taba;

RULE 11
IF city=Aqaba AND
amount_ of_ money _to __pay=50$ AND
number_ of_ companions=One _Person AND
type_ of_room=Normal AND
using_ car_ of_ hotel=No

THEN hotel_ name_ 1=Hilton_ Nelson_ Village
hotel _name _2=A1Zaitouna
hotel_ name_ 3=Domina _AquamorinaII;

RULE12
IF city=Aqaba AND
amount_ of_money_ to__pay=50$ AND
number_ of_companions=One _Person AND
type_of_room=Double AND
using_ car_of_hotel= Yes

THEN hotel_ name_ 1=Tabya_ Boutique
hotel_ name_2=Holiday _Inn_Patio
hotel_ name_ 3=Sheraton _Herods _Palace;

RULE 13
IF city=Aqaba AND
amount_ of_money_ to__pay=50$ AND
number_ of_companions=One _Person AND
type_of_room=Double AND
using_ car_of_hotel= No

THEN hotel_name_ 1=Movenpick _Resort_ Aqaba
hotel_name_2=Holiday_ Inn_Patio
hotel_ name_ 3=Intercontinetual_ Aqaba;

RULE14

IF city=Aqaba AND
amount_ of_ money_ to_pay=50$ AND
number_ of_ companions=More _Than_ One _Person AND
type_ of_ room=Single AND
using_car_of_hotel=Yes OR
using_ car_ of_ hotel= No

THEN hotel_ name_ 1=Sorry_ You_ Cannot_ Reserve
hotel_ name_ 2=Sorry _You_ Cannot_ Reserve
hotel_ name _3=Sorry _You_ Cannot_reserve;

RULE15
IF city=Aqaba AND
amount_ of_ money_ to_pay=50$ AND
number_ of_ companions=More _Than_ One_ Person AND
type_of_room=Normal AND
using_ car_ of_ hotel= Yes

THEN hotel_ name_ 1=Herods _Vitalis _Spa
hotel _ name _ 2=Shweiki
hotel_ name_ 3=Holiday _Inn;

RULE16
IF city=Aqaba AND
amount_ of_money_ to_pay=50$ AND
number_ of_companions=More _Than_ One_Person AND
type_of_room=Normal AND
using_ car_of_hotel= No

THEN hotel_name_1=Golden_Tuilp
hotel_ name_ 2=Alcazar
hotel_ name_ 3=Movenpick _Resort;

RULE17
IF city=Aqaba AND
amount_ of_money_ to_pay=50$ AND
number_ of_companions=More _Than_ One_Person AND
type_of_room=Double AND
using_ car_of_hotel= Yes

THEN hotel_name_ 1=Express_ holiday_ Inn_Eliat
hotel_name_ 2=Sport _Club
hotel_ name_ 3=Hilton;

RULE18
IF city= Amman AND
amount_ of_money _to _pay=50$ AND
number_ of_companions=More _Than_ One_ Person AND
type_ of_room=Double AND
using_ car_ of_ hotel= Yes

THEN hotel _ name _ 1=Mamoura
hotel_ name_2=Meryland
hotel_ name_3=New_ Park;

RULE19
IF city=Aqaba AND
amount_ of_money _to_pay=50$ AND
number_ of_companions=More _Than_ One_Person AND
type_ of_room=Double AND
using_ car_of_hotel= No

THEN hotel_ name_ 1=holiday_ Inn_Patio
hoteljıameZs=MagicSunrise
hotel_ naıne_3=Holiday_ Inn_Express _Eilat;

RULE20
IF city=Aqaba AND
amount_ of_money_ to_pay=50$ AND
number_ of_companions=Alone AND
type_ of_room=Single AND
using_ car_of_hotel= Yes

THEN hotel_ name_ l=Magic_ Palace
hotel_ name_2=Movenpick _Tuilp
hotel_naıne_3=Hilton_Nelson_ Village;

RULE21
IF city=Aqaba AND
amount_ of_money_ to_pay=50$ AND
number_ of_companions= Alone AND
type_ of_room=Single AND
using_car_of_hotel=No

THEN hotel_ name_ l=Palmira
hotel_ name_ 2=radisson _ SAS_ Resort
hotel_name_3=Herads_Fourm;

RULE22

IF city=Aqaba AND
amount_ of_ money_ to_pay=50$ AND
nunıbe _of_ companions=Alone AND
type_of_room=Normal AND
using_ car_ of_ hotel= Yes

THEN hotel _ name _ l=Paradise _ Club
hotel_ name_ I=Tobya_ Boutique
hotel_ name_ I=King_ Salomns_Palace;

RULE23

IF city=Aqaba AND
amount_ of_money_ to_pay=50$ AND
number_ of_companions=Alone AND
type_ of_room=Normal AND
using_car_of_hotel=No

THEN hotel_ name_ I=Movenpick _Tuilp
hotel_name_ 2=Hilton _Eilat_ Queen_ Of_ Sheba
hotel_ name_ 3=Crown_ Palaza;

RULE24

IF city=Aqaba AND
amount_of_money_to_pay=50$ AND
number_ of_companions=Alone AND
type_ of_room=Double AND
using_ car_ of_hotel= Yes

THEN hotel_ name_ I=Golden_ Tulip
hotel_ name_ 2=Paradise _ Club
hotel_ name_ 3=Shweiki;

RULE25

IF city=Aqaba AND
amount_ of_ money _to_pay=50$ AND
number_ of_companions=Alone AND
type_ of_room=Double AND
using_ car_ of_hotel= No

THEN hotel_name_1=New_Caesar
hotel_name_2=Movenpick _Tuilp
hotel_name_3=Lagoona;

RULE26
IF city=Aqaba AND
amount_of_money_to__pay=50$ AND
number_of_companions= Alone AND
type_of_room=Double AND
using_car_of_hotel= Yes

THEN hotel_name_1=Rimonim _Neptune
hotel_name_2=Herods _Vitalis _Spa
hotel_name_3=Dan _Eilat;

RULE27
IF city=Aqaba AND
amount_of_money_to__pay=50$ AND
number_of_companions= Alone AND
type_of_room=Double AND
using_car_of_hotel= No

THEN hotel_name_1=Coral_ Bay
hotel_name_2=Radisson_SAS Resort
hotel_name_3=Royal_Beach;

RULE28
IF city=Aqaba AND
amount_of_money_to_pay= Less_Than_50$ AND
number_of_companions=One _Person AND
type_of_room=Single AND
using_car_of_hotel= Yes OR
using_car_of_hotel=No

THEN hotel_name _1=Sorry_You_Cannot_Reserve
hotel_name_2=Sorry _You_ Cannot_ Reserve
hotel_name _3=Sorry _You_ Cannot_ Reserve;

RULE29
IF city=Amman AND

amount_ of_ money_ to_pay=50$ AND
number_ of_ companions=More _Than_ One _Person AND
type_ of_room=Double AND
using_ car_ of_ hotel= No

THEN hotel_name_1 =Alia_gateway
hotel_ name_ 2= Abjar
hotel_ name_ 3=Carlton;

RULE30
IF city= Amman AND
amount_ of_ money_ to_pay=50$ AND
number_ of_ companions= Alone AND
type_ of_room=Single AND
using_ car_ of_ hotel= Yes

THEN hotel _ name _ 1=Jerusalem _ International
hotel_ name_ 2=Commadore
hotel_ name_ 3=Hisham;

RULE31
IF city=Amman AND
amount_ of_money _to_pay=50$ AND
number_ of_companions=Alone AND
type_ of_room=Single AND
using_car_of_ hotel=No

THEN hotel_ name_ 1=Amman _ Forum
hotel_name_ 2=Crystal
hotel_name_3=Garden;

RULE32
IF city=Amman AND
amount_of_money_to_pay=50$ AND
number_ of_companions= Alone AND
type_ of_room=Normal AND
using_ car_of_hotel= Yes

THEN hotel_ name_ 1=Turnio
hotel_ name_ 2=Middle _ East
hotel_ name _3=Liwan;

RULE33
IF city= Amman AND
amount_ of_ money_ to__pay=50$ AND
number_ of_ companions= Alone AND
type_of_room=Normal AND
using_car_of_hotel=No

THEN hotel_ name_ 1=Region
hotel _ name _ 2=Ambassador
hotel_name_3=Ailiwan;

RULE34
IF city=Amman AND
amount_of_money_to__pay=50$ AND
number_of_companions= Alone AND
type_of_room=Double AND
using_car_of_hotel= Yes

THEN hotel_name_1=Cham
hotel_ name_ 2=Dana_ Plaza
hotel_name_3=Arena _Space;

RULE35
IF city=Amman AND
amount_of_money_to__pay=50$ AND
number_of_companions= Alone AND
type_of_room=Double AND
using_car_of_hotel= No

THEN hotel_name_1=Alia_ Gateway
hotel_ name_ 2=Amman _ International
hotel_name_ 3=Jerusalem _International;

RULE36
IF city=Amman AND
amount_of_money_to_pay= Less_Than_ 50$ AND
number_of_companions=One _Person AND
type_of_room=Single AND
using_car_of_hotel= Yes OR
using_car_of_hotel=No

THEN hotel_name_1=Sorry_You_cannot_Reserve
hotel_name_2=Sorry_You_Cannot_Reserve
hotel_name_3=Sorry_You_Cannot_Reserve;

RULE37
IF city=Amman AND
amount_of_money_to_pay=Less_Than_50$ AND
number_of_companions=One_Person AND
type_of_room=Normal AND
using_car_of_hotel=Yes

THEN hotel_name_1=Bethlehem
hotel_name_2=Eslami
hotel_name_3=Altan;

RULE38
IF city=Amman AND
amount_of_money_to_pay=Less_Than_50$ AND
number_of_companions=One_Person AND
type_of_room=Normal AND
using_car_of_hotel=No

THEN hotel_name_1=Karnok
hotel_name_2=Nebel
hotel_name_3=Happy_Land;

RULE39
IF city=Amman AND
amount_of_n1oney_to_pay=Less_Than_50$ AND
number_of_companions=One_Person AND
type_of_room=Double AND
using_car_of_hotel=Yes

THEN hotel_name_1=Gulf
hotel_name_2=J awhert_Alquds
hotel_name_3=Sun_Rise;

RULE40
IF city=Aqaba AND
amount_of_money_to_pay=Less_Than_50$ AND
number_of_companions=One_Person AND
type_of_room=Normal AND

using_ car_ of_ hotel= Yes

THEN hotel_ name_ 1=Movenpick _Resort
hotel _ name _ 2=Dweikz
hotel_name_ 3=Holiday_ Inn_ Resort;

RULE41
IF city=Aqaba AND
amount_ of_money_ to_pay= Less_Than_ 50$ AND
number_ of_companions=One _Person AND
type_of_room=Normal AND
using_ car_ of_hotel=No

THEN hotel_name_l=Peace_Way
hotel_name_ 2=Golden_ Tulip
hotel_name_ 3=A1Zaitouna;

RULE42
IF city=Aqaba AND
amount_of_money_to_pay=Less_Than_50$ AND
number_ of_companions=One _Person AND
type_of_room=Double AND
using_ car_of_hotel= Yes

THEN hotel_name_ 1=Golden_ Tuilp_Resort
hotel_name_ 2=Movenpick _Tuilp_
hotel_name _3=Commodore;

RULE43
IF city=Aqaba AND
amount_of_money_ to_pay= Less_Than_ 50$ AND
number_ of_companions=One _Person AND
type_of_room=Double AND
using_car_of_hotel=No

THEN hotel_name_ 1=Golden_ Tulip
hotel_name_ 2=Holiday _Inn_Resort
hotel_ name _3=Shweiki;

RULE44
IF city=Aqaba AND
amount_of_money_to_pay=Less_Than_50$ AND

number_ of_ companions=More _Than_ One _Person AND
type_ of_room=Single AND
using_car_of_hotel=Yes OR
using_ car_ of_ hotel=No

THEN hotel_ name_ 1=Sorry_ You_ Cannot_ Reserve
hotel_ name_ 2=Sorry _You_ Cannot_ Reserve
hotel_ name _3=Sorry _You_ Cannot_ Reserve;

RULE45
IF city=Aqaba AND
amount_ of_ money_ to_pay=Less _Than _50$ AND
number_ of_ companions= More_ Than_ One _Person AND
type_of_room=Normal AND
using_ car_ of_ hotel= Yes

THEN hotel_ name_ 1=Crystal
hotel _ name _ 2=ALZaitouna
hotel_ name_ 3=A1Cazar;

RULE46
IF city=Aqaba AND
amount_ of_ money _to_pay=Less _Than_ 50$ AND
number_ of_ companions=More _Than_ One_Person AND
type_ of_ room=Normal AND
using_ car_ of_ hotel=No

THEN hotel_ name_ 1=Holiday _Inn_ Aqaba
hotel_ name_ 2=Golden _Tuilp _Resort
hotel_ name_ 3=Domina _AqumorinaII;

RULE47
IF city=Aqaba AND
amount_of_money_to_pay=Less_Than_50$ AND
number_ of_ companions=More _Than_ One_Person AND
type_ of_ room=Double AND
using_ car_ of_ hotel=Yes

THEN hotel_ name_ I=AquamorinaII_ City
hotel_ name_ 2=AlShula
hotel_ name_ 3=Dweikz;

RULE48
IF city=Aqaba AND
amount_ of_money_ to__pay=Less _Than _50$ AND
number_ of_ companions=More _Than_ One _Person AND
type_ of_room=Double AND
using_ car_ of_ hotel= No

THEN hotel_name_l=Movenpick_Resort
hotel_name _2=Golden _Tuilp
hotel_ name _3=Aquamorinall _City;

RULE49
IF city=Aqaba AND
amount_ of_ money _to _pay= Less_ Than _50$ AND
number_ of_ companions= Alone AND
type_ of_room=Single AND
using_ car_ of_ hotel= Yes

THEN hotel _ name _ 1=Sheraton _ Herods _ Palace
hotel_name_2=Peace_ Way
hotelrıame _3=Commodore;

RULE50
IF city=Amman AND
amount_ of_money_ to_pay= Less_ Than_ 50$ AND
number_ of_companions=More _Than_ One_Person AND
type_of_room=Single AND
using_car_of_hotel=Yes OR
using_car_of_hotel=No

THEN hotel_name_ 1=Sorry_ You_ Cannot_ Reserve
hotel_name_ 2=Sorry_You_ Cannot_ Reserve
hotel_name_ 3=Sorry_You_ Cannot_ Reserve;

RULE 51
IF city=Amman AND
amount_of_money_ to__pay=Less_Than_50$ AND
number_ of_companions=More _Than_ One_Person AND
type_of_room= Normal AND
using_car_of_hotel=Yes

THEN hotel_name_l=Merryland
hotel _ name _ 2=0cean
hotel_name _3=Geneva;

RULE52
IF city=Aınnıan AND
amount_of_money_to_pay=Less_Than_50$ AND
number_ of_companions=More _Than_ One_Person AND
type_of_room= Normal AND
using_ car_of_hotel=No

THEN hotel_name _1=Shepord
hotel_ name_ 2=Liwan
hotelrıame _3=Cameo;

RULE53
IF city=Aınnıan AND
amount_ of_money_ to_pay= Less_ Than _50$ AND
number_ of_companions=More _Than_ One_Person AND
type_of_room= Double AND
using_ car_of_hotel= Yes

THEN hotel_name_ l=Dove
hotel_ name_ 2=Canary
hotel_ name_ 3=Caravan;

RULE54
IF city=Aınınan AND
amount_of_money_to_pay=Less_Than_50$ AND
number_ of_companions=More _than_ One_person AND
type_ of_room=Double AND
using_ car_of_hotel=No

THEN hotel_name _1=Qasr _Albalka
hotel_name_2=Holyland
hotel_ nan1e_3=Rum_ continental;

RULE55
IF city=Aınnıan AND
aınount_ of_ money_ to_pay= Less_ Than_50$ AND
number_ of_ companions= Alone AND

type_of_room=Single AND
using_car_of_hotel= Yes

THEN hotel_name_l=Firas_ Wing
hotel_name_2=Hala_Inn
hotel_name_3=Sandy _palace;

RULE56
IF city=Amman AND
amount_of_money_to_pay=Less_Than_50$ AND
number_of_companions= Alone AND
type_of_room=Single AND
using_car_of_hotel=No

THEN hotel_name_1=Crystal
hotel_name_2=Alqasr
hotel_name_3=Hisham;

RULE57
IF city=Amman AND
amount_of_money_to_pay=Less_Than_50$ AND
number_of_companions= Alone AND
type_of_room= Normal AND
using_car_of_hotel=Yes

THEN hotel_name_1=Select
hotel_name_2=Sultan
hotel_name_3=Caravan;

RULE 58
IF city=Aqaba AND
amount_of_money_to_pay= Less_Than_50$ AND
number_of_companions= Alone AND
type_of_room=Single AND
using_car_of_hotel= No

THEN hotel_name_l=Dweikz
hotel_name_2=Sheraton_herods_Palace
hotel_nmae_3=movenpick _Resort_ Aqaba;

RULE59

IF city=Aqaba AND
amount_ of_money _to _pay=Less _Than _50$ AND
number_ of_ companions= Alone AND
type_ of_room= Normal AND
using_ car_ of_hotel= Yes

THEN hotel_name _1=Holiday _Inn
hotel _name _2=A1Cazar
hotel_name _3= ALZaitouna;

RULE60
IF city=Aqaba AND
amount_ of_money _to_pay=Less _Than _50$ AND
number_ of_companions= Alone AND
type_of_room= Normal AND
using_car_of_hotel=No

THEN hotel_name_ 1=Shalom _Plaza
hotel_name _2=Holiday _Inn _Resort _Aqaba
hotel_ name_ 3=Commodore;

RULE 61
IF city=Aqaba AND
amount_ of_money _to_pay=Less _Than _50$ AND
number_ of_companions= Alone AND
type_ of_room=Double AND
using_ car_ of_hotel=Yes

THEN hotel_name_ 1=Crystal
hotel_ name_ 2=Hilton _Taba
hotel_name _3=Golden _Tulip_ Eliat;

RULE62
IF city=Aqaba AND
amount_ of_money _to_pay=Less _Than _50$ AND
number_ of_companions= Alone AND
type_ of_room=Double AND
using_ car_of_hotel=No

THEN hotel_name_ 1=Golden_ Tulip _Resort
hotel_name_2=Movenpick_Resort
hotel_name _3=Hilton _Nelson_ Village;

RULE63
IF city=Aqaba AND
amount_ of_ money _to _pay=More _Than _50$ AND
number_ of_ companions=One _Person AND
type_ of_room=Single AND
using_car_of_hotel=Yes OR
using_ car_ of_hotel=No

THEN hotel_name _1=Sorry_ You_ Cannot_ Reserve
hotel_name _2=Sorry _You_ Cannot_ Reserve
hotel_ name _3=Sorry _You_ Cannot_Reserve;

RULE64
IF city=Aqaba AND
amount_ of_money _to _pay= More_ Than_ 50$ AND
number_ of_ companions=One _Person AND
type_ of_room= Normal AND
using_ car_ of_hotel= Yes

THEN hotel _ name _ 1=Crown _Plaza
hotel_name_2=Tobya_Boutique
hotel_nmae_3=Sport_Club;

RULE65
IF city=Amman AND
amount_of_money_to_pay=Less _Than_50$ AND
number_of_companions= Alone AND
type_of_room=Normal AND
using_car_of_hotel=No

THEN hotel_name_1=Alia_Gatway
hotel_name_2=Crystal
hotel_name_3=Amra _F orum;

RULE66
IF city=Amman AND
amount_of_money_to_pay=Less _Than_ 50$ AND
number_ of_companions= Alone AND
type_of_room= Double AND
using_car_of_hotel=Yes

THEN hotel_ name_ 1=Cham

hotel_name_2=Granda
hotel_name_3=Saveen;

RULE67

IF city=Amman AND
amount_of_money_to_pay=Less _Than_50$ AND
number_of_companions= Alone AND
type_of_room=Double AND
using_car_of_hotel=No

THEN hotel_name_1=Gardenia
hotel_name_2=Amman_ International
hotel_name_3=Holiday_Inn;

RULE68

IF city=Amman AND
amount_of_money_to_pay= More_Than_50$ AND
number_of_companions=One_Person AND
type_of_room=Single AND
using_car_of_hotel=Yes OR
using_car_of_hotel=No

THEN hotel_name_1=Sorry_ You_ Cannot_ Reserve
hotel_name_2=Sorry_You_ Cannot_Reserve
hotel_name_3=Sorry _you_ Cannot_Reserve;

RULE69

IF city=Amman AND
amount_of_money_to_pay=More_Than_50$ AND
number_of_companions=One _Person AND
type_of_room=Nonnal AND
Using_car_of_hotel= Yes

THEN hotel_name_1=Sheraton
hotel_name_2=Amra_ Forum
hotel_name_3=Amman _Crown;

RULE 70

IF city=Amman AND
amount_of_money_to_pay=More_Than_50$ AND

number_ of_companions=One _Person AND
type_ of_room=Normal AND
using_car_of_hotel=No

THEN hotel_name_1=Ramada
hotel_ name_2=Turino
hotel_name _3=Tyche;

RULE 71
IF city=Amman AND
an1ount_of_money_to_pay=More_Than_50$ AND
number_ of_companions=One _Person AND
type_of_room=Double AND
using_ car_of_hotel= Yes

THEN hotel_name_1=Hillside
hotel_name_2=Firas
hotel_ name_ 3=Grand_Place;

RULE 72
IF city=Aımnan AND
amount_ of_money_to_pay=More _Than_ 50$ AND
number_ of_companions=One _Person AND
type_ of_room=Double AND
using_car_of_hotel=No

THEN hotel_name_1=Jerusalem International
hotel_name_2=Dove
hotel_name _3=garden;

RULE 73
IF city=Amman AND
amount_ of_money_to_pay= More_Than_50$ AND
number_ of_companions=More _Than_ One_Person AND
type_of_room=Single AND
using_car_of_hotel=Yes OR
using_car_of_hotel=No

THEN hotel_name_ I=Sorry_you_ Cannot_Reserve
hotel_name_2=Sorry _You_ Cannot_ Reserve
hotel_name_3=Sorry _You_ cannot_ Reserve;

RULE 74

IF city=Amman AND
amount_of_money_to_pay= More_Than_ 50$ AND
number_of_companions=More _Than_ One_Person AND
type_of_room=Norrnal AND
using_car_of_hotel= Yes

THEN hotel_narne_l=Alqasr
hotel_name_2=Darotel
hotel_name_3=San_Rock;

RULE75
IF city=Amman AND
amount_of_money_to_pay=More _Than_ 50$ AND
number_of_companions=More _Than_ One_Person AND
type_of_room=Nonnal AND
using_car_of_hotel=No

THEN hotel_name_l=Hisham
hotel_name_2=Middle_ East
hotel-name _3=Holiday _Inn;

RULE 76
IF city=Aqaba AND
amount_of_money_to-pay=More_Than_50$ AND
number_of_companions=One _Person AND
type_of_room=Norrnal AND
using_car_of_hotel=No

THEN hotel_name_1=Palmira
hotel_name_2=Rimonim_Neptune
hotel_name_3=Sheraton_Mariah_Eliat;

RULE 77
IF city=Aqaba AND
amount_of_money_to_pay=More_Than_50$ AND
number_of_companions=One _Person AND
type_of_room=Double AND
using_car_of_hotel=Yes

THEN hotel_name_1=King_ Solomons_Palace
hotel_name_2=New_ Caesar
hotel_name_3=Sofitel;

RULE 78
IF city=Aqaba AND
arnount_of_money_to_pay=More_Than_50$ AND
number_ of_companions=One _Person AND
type_of_room=Double AND
using_ car_of_hotel=No

THEN hotel_name_l=Dan_Panorawa_Eilat
hotel_ name_ 2=Herads_Forum
hotel_name_3=Le_Meridien _Eilat;

RULE 79
IF city=Aqaba AND
amount_ of_money_ to_pay=More _Than_ One_Person AND
number_ of_companions=More _Than_ One_Person AND
type_ of_room=Single AND
using_car_of_hotel=Yes OR
using_ car_ of_hotel=No

THEN hotel_ name_ 1=Soory_You_ Cannot_Reserve
hotel, narne_2=Sorry _You_ Cannot_Reserve
hotel_ name_ 3=Sorry_You_ Cannot_Reserve;

RULE80
IF city=Aqaba AND
amount_ of_money_ to_pay=More _Than_ 50$ AND
number_ of_companions=More _Than_ One_ Person AND
type_of_room=Normal AND
using_car_of_hotel=Yes

THEN hotel_ name_1=Crown_ Plaza
hotel name_2=Dan_Eilat
hotel: name_3=Eilat_Princess;

RULE 81
IF city=Aqaba AND
amount_of_money_to_pay=More_Than_50$ AND
number_ of_companions=More _Than_ One_Person AND
type_ of_room=Normal AND
using_ car_of_hotel=No

THEN hotel_name_1=Galei
hotel_name_2=Hiton_Taba_Resort
hotel_name_3=Holiday_Inn_Taba;

RULE82 AND
IF city=Aqaba AND
amount_of_money_to__pay=More_Than_50$ AND
number_of_companions=More_Than_One_Person AND
type_of_room=Double
using_car_of_hotel=Yes

THEN hotel_name_1=Lagoona
hotel_name_2=Mercure_Mirage_Eilat
hotel_name_3=Sonesta_Beach;

RULE83 AND
IF city=Amman AND
amount_of_money_to__pay=More_Than_50$ AND
number_of_companions=More_Than_One_Person AND
type_of_room=Double
using_car_of_hotel=Yes

THEN hotel_name_1=Arena_Space
hotel_name_2=Golden
hotel-name_3=Arabila;

RULE84 AND
IF city=Aqaba AND
amount_of_money_to_pay=More_Than_50$ AND
number_of_companions=Alone AND
type_of_room=Single
using_car_of_hotel=Yes

THEN hotel_name_1=Commodore
hotel_name_2=Aquamorinall_City
hotel_name_3=A1Zaitouna;

RULE85 AND
IF city=Aqaba AND
amount_of_money_to_pay=More_Than_50$

number_ of_companions=Alone AND
type_of_room=Single AND
using_car_of_hotel=N o

THEN hotel_name_ I=Aqaba _Gulf
hotel_name_ 2=AquamorinaII_ City
hotel_name_ 3=Dweikz;

RULE86
IF city=Aqaba AND
amount_of_money_to_pay=More _Than_50$ AND
number_ of_companions=Alone AND
type_of_room=Normal AND
using_car_of_hotel=Yes

THEN hotel_name_ l=Intercontineutal_ Aqaba
hotel_name_2=Peace_ Way
hotel_name_3=Herods_ Vitalis_Spa;

RULE87
IF city=Aqaba AND
amount_of_money_to_pay=More _Than_50$ AND
number_ of_companions= Alone AND
type_of_room=Normal AND
using_car_of_hotel= No

THEN hotel_ name_ l=Sheraton _ Herods_ Palace
hotel_ name_ 2=Commodore
hotel_name_ 3=Crystal;

RULE88
IF city=Aqaba AND
amount_of_money_to_pay=More _Than_ 50$ AND
number_ of_companions=Alone AND
type_of_room= Double AND
using_car_of_hotel= Yes

THEN hotel_name_ l=Domina_ Aquaınorina
hotel_ name_ 2=A1Zaitouna
hotel_nanıe_3=Golden_Tuilp;

RULE89

IF city=Aqaba AND
amount_of_money_to_pay=More_Than_50$ AND
number_ of_companions= Alone AND
type_ of_room=Double AND
using_ car_of_hotel=No

THEN hotel_nmae_1=Dweikz
hotel_name_2=MovenPick_Resort
hotel_ name_ 3=Sheraton;

RULE90

IF city=Amman. AND
amount_of_money_to_pay=More_Than_50$ AND
number _of_companions=More _Than_ One_Person AND
type_ of_room=Double AND
using_ car_of_hotel=No

THEN hotel_name_1=Cham
hotel_ name_2=Lipton
hotel_name _3=Gulf;

RULE91

IF city=Arnman AND
amount_ of_money_ to_pay= More_ Than_50$ AND
number_ of_companions= Alone AND
type_of_room=Single AND
using_ car_ of_hotel=Yes

THEN hotel_name_l=Nebel
hotel_ name_ 2=0lympia
hotel_ name _3=Rozana;

RULE92

IF city=Amman AND
amount_ of_money_ to_pay= More_ Than _50$ AND
number_ of_companions= Alone AND
type_ of_room=Single AND
using_ car_of_hotel=No

THEN hotel_name_l=Bobelon
hotel name 2=Alreınal

hotel_ name_ 3=Cleopotra;

RULE93
IF city= Amman AND
amount_ of_money _to_pay=More _Than_ 50$ AND
number_ of_ companions=Alone AND
type_of_room=Normal AND
using_car_of_hotel=Yes

THEN hotel name l=Asia
hotel _ name _ 2=Nebel
hotel_ name_ 3=lords;

RULE94
IF city=Amman AND
amount_ of_money_to_pay=More _Than_ 50$ AND
number_ of_companions= Alone AND
type_ of_room=Normal AND
using_ car_ of_hotel= No

THEN hotel_ name_1=Daraghmeh _Furnished _Apartment
hotel_ name_ 2=0lympia
hotel_ name _3=Ywca;

RULE95
IF city=Amman AND
amount_ of_money_ to_pay=More _Than_ 50$ AND
number_ of_companions=Alone AND
type_ of_room=Double AND
using_ car_ of_hotel= Yes

THEN hotel_name_l=Capri
hotel_ nmae_ 2=Kindi
hotel_name_3=Jaddah_Palace;

RULE96
IF city=Amman AND
amount_ of_money_to_pay= More_ Than_50$ AND
number_ of_companions= Alone AND
type_ of_room=Double AND
using_ car_ of_hotel= No

```
THEN hotel_name_1=Dove
        hotel_name_2=Razan
        hotel_name_3=Royal;


RULE 97
IF     city=Amman                                            AND
       amount_of_money_to_pay=Less _Than_50$                 AND
       number_of_companions=One _Person                      AND
       type_of_room=Double                                   AND
       using_car_of_hotel=Yes

THEN   hotel_name_1=Amman_International
        hotel_name_2=Holiday _Inn_Amman
        hotel_name_3=Region;


RULE98
IF     city=Amman                                            AND
       amount_of_money_to_pay=Less_Than_50$                  AND
       number_of_companions=One _Person                      AND
       type_of_room=Double                                   AND
       using_car_of_hotel=No

THEN   hotel_name_1=Amman_International
        hotel_name_2=Amra_Forum
        hotel_name _3=Amman;


ASK city:"Which is city you want visit it injordan?";
CHOICES city:Amman,Aqaba;

ASK amount_of_money_to_pay:"What is amount of money you are going to pay?";
CHOICES amount_of_money_to_pay:50$,Less_Than_50$,More_Than_50$;

ASK number_of_companions:"What is the companions number you will take with you?";
CHOICES number_of_companions:One_Person,More_Than_One_Person,Alone;

ASK type_of_room:"What is the type of room that you want reserve it?";
CHOICES type_of_room:Single,Normal,Double;

ASK using_car_of_hotel:"Do you want to use the car of hotel?";
CHOICES using_car_of_hotel: Yes,No;
```