ISLAMIC EXPERT SYSTEM

By

Hidayat-Ullah Khan

Department of Computer Science,
Military College of Signals
Rawalpindi

August 2001

ISLAMIC EXPERT SYSTEM

By

Hidayat-Ullah Khan

Submitted to the Faculty of Computer Science,

National University of Science and Technology, Rawalpindi,

In partial fulfillment for the requirement of MS Degree.

August 2001

Dedicated to

My Parents, Wife and Children

Abstract

Expert systems are like human experts. Since currently it is not possible to simulate human experts so we call them as decision support systems, which helps the human experts in decision-making. Decision support systems are typically based on knowledge acquired from domain experts, traditionally using techniques of question answering, form filling, and knowledge editing. Though advances in such techniques have eased the process of knowledge acquisition, this process still remains a bottleneck in developing new decision support systems.

The research argues that systems can automatically acquire new knowledge by solving problems and letting the expert provide advice, criticism, and verification. The system produced by this research work is aimed to be like a base of an Islamic expert system. Currently it is like an intelligent, efficient, referencing system. It contains a huge knowledge base consisting of Quranic Translations, Hadith Translation, an English Dictionary and many more system files consisting of Islamic Knowledge. This system is the implementation of Artificial Intelligence (AI) concepts in the field of Islamic education. The Islamic knowledge is represented in the form of Frames and Semantic Nets. It also has an intelligent query handler built on top of its knowledge base.

Query handler is connected with the learning module of this system. While handling the user queries system also learns from the user. Then it get the verification from a human Islamic scholar for it new knowledge. Islamic scholar can edit the knowledge base of the system. User can also perform simple search through translations of Quran and Hadith.

Thesis report contains all the relevant knowledge of this research work. AI concepts, which were within the scope of this work, are discussed in detail. Then there is complete conceptual information of implementation of the system. This work is a humble effort of implementing the AI concepts.

ACKNOWLEDGEMENTS

First, I owe gratitude to Almighty Allah, the most merciful and compassionate, most Gracious and beneficial whose favor made it possible for me to accomplish this task Assigned to me and for He has taken the responsibility of showing the right path.

First person to be thanked is the teacher of all humanity, Holy Prophet (Peace Be Upon Him), for accomplishing the responsibility of conveying the message of God to us. Who is, forever, a source of guidance and knowledge for the mankind.

I feel honor to express heartiest gratitude to my supervisor, Dr. Mukhtar Hussain for his kind supervision, sincere advice and ever friendly loving attitude during my whole MCS and this research work.

I would like to thanks Mr. Salman Wasi for helping me in this work and giving me useful advice. I am thankful to my all other computer science teachers; all credit goes to them for my current and future work in this field.

I am grateful to all of my official and unofficial teachers from class one to this master class. Every one was exceptional, excellent, marvelous, superb and superior.

Words cannot say the gratitude that I feel for my parents whose affection and prayers have always been the key to my success. I love my brothers and sisters; their thought kept me on the track. Sincere thanks to all my family members for their ever supporting attitude.

I am thankful to all of my old and new friends for their help and moral support, during my life. Their thinking give me strength and power. They are so many that it will take another write-up for just mentioning their names.

My thanks to Military College of Signals and National University of Science and Technology for building an institution of highly qualified instructional staff and provision of advanced research facilities.

CONTENTS

List of Figures

ABBREVIATIONS

AB	**Answer Object**
ADJ	**Adjective**
AI	**Artificial Intelligence**
AO	**Answer Object**
CLOS	**Common Lisp Object System**
GUI	**Graphical User Interface**
IEXPERT	**Islamic Expert system**
I/O	**Input Output**
KB	**Knowledge Base**
Lisp	**List Processing**
MCS	**Master of Computer Science**
ML	**Machine Learner**
NP	**Noun Phrase**
OOP	**Object Oriented Programming**
PBUH	**Peace Be Upon Him**
PC	**Personal Computer**
PRO	**Proverb**
RM	**Reference Marker**
QB	**Question Builder**
QH	**Query Handler**
QO	**Question Object**
VP	**Verb Phrase**

Chapter 1

INTRODUCTION

"Read: In the name of thy Lord Who createth,
Createth man from a clot.
Read: And thy Lord is the Most Bounteous,
Who teacheth by the pen,
Teacheth man that which he knew not."

[96.01-96.05, AL-Quran Translated by Pickthall]

Islam is a religion revealed through Muhammad (PBUH). Followers of Islam are called Muslims. Islam has a huge knowledge base. This knowledge base contains about every field of human life. It is spread from daily life behavior up to the field of Space Science. As the Knowledge itself is a very vast, it is not possible for a human brain to keep the complete knowledge of even a single thing. Islamic knowledge base is the bases of the knowledge of universe including the humans. Islam asks Humans to think and study the life, the rules of nature; Humans are doing so and this is the main reason that they are ruling this planet (Earth).

The verses of QURAN given above tell us about a basic principle, that God has taught the humans by Reading and Writing. Man started using pen to write down the information he gets, these information are utilized by the next generations. Because of this principle of reading and writing the knowledge has become cumulative i.e. each generation is standing on the shoulders of its predecessors. This rule is the main reason of human's progress.

Still we are obeying this rule, knowledge is spreading exponentially. We have shifted from pen and paper to PCs and Diskettes in order to compete with the increasing speed of knowledge. But as the errant use of pen and paper do not fulfill the purpose of knowledge keeping similarly if we do not use our PCs intelligently, we'll not be able to collect the knowledge for later use.

Until we don't shift from PCs and Diskettes to something else, which is more power full in knowledge keeping and retrieving, we would have to use our Computers more and more intelligently. We should build such systems that are intelligent in knowledge keeping, knowledge searching, knowledge engineering and knowledge acquisition.

Field of Artificial Intelligence in Computer Science is concerned with this issue. Our future depends a lot on our success in this field. But it is not simple to make a machine intelligent, It depends that what is the required level of intelligence. In AI the important topics are Machine Learning, Search, Knowledge representation, Pattern Matching etc.

Research is going on in all the departments of AI. Many techniques have been developed in each of these sub-domains. Systems are being developed which use these techniques. But the success ratio in Software Engineering is very low especially in AI it is worse. So it will take much effort to progress in this field but it is vital for our future success.

This chapter will introduce you to this research work, the resulting software and the report.

1.1 MY RESEARCH WORK

The ultimate goal of my research work is to build a system, which behaves like a Human Islamic scholar (Islamic Expert System). Definitely it was impossible for me to achieve this goal alone in one or two years. Keeping this in mind I aimed to provide at least a solid, real and actual, base for such a system. And this itself is an endless goal.

The systems, which behave like human experts of any particular domain, are called expert systems. Expert systems was in fact the one of the hottest area of research in the world. These systems should posses all the characteristics of the human exert, like huge Knowledge Base, problem solving ability in case of tangling input or difficult situation.

1.1.1 Background

There are many questions about all this, first one might think that, why I am doing this? Or in other words why I have selected this particular topic? To answer this question we should think about some more basic and important questions, which we ignore. Simple ones might be what we (humans) are and why we are? Why we are here (in this world)? Where we were previously? What is past and future? What is actual truth? etc. These questions are basic but very difficult to answer. Every one may have its own answers or most of us have never thought about these questions. Now we try to discuss the answers of above basic questions one by one but you may notice that our proceeding discussion is based on Islamic point of view. First what we are? We are creatures created by Someone. We are said to be living due to our some characteristics.

“002.029 It is He Who hath created for you all things that are on earth; Moreover His design comprehended the heavens, for He gave order and perfection to the seven firmaments; and of all things He hath perfect knowledge.”

“002.030 Behold, thy Lord said to the angels: "I will create a vicegerent on earth." They said: "Wilt Thou place therein one who will make mischief therein and shed blood?- whilst we do celebrate Thy praises and glorify Thy holy (name)?" He said: "I know what ye know not." We need some substances for our material existence. But I don’t know about our exact nature. Now why we are, Islam says that we are for the worship of that Someone who created us. Nature of the worship is still very discussible.”

Now the next question is why we are here on this planet?

“002.035 We said: "O Adam! dwell thou and thy wife in the Garden; and eat of the bountiful things therein as (where and when) ye will; but approach not this tree, or ye run into harm and transgression."

“002.036 Then did Satan make them slip from the (garden), and get them out of the state (of felicity) in which they had been. We said: "Get ye down, all (ye people), with enmity between yourselves. On earth will be your dwelling-place and your means of livelihood - for a time."”

“002.038 We said: "Get ye down all from here; and if, as is sure, there comes to you Guidance from me, whosoever follows My guidance, on them shall be no fear, nor shall they grieve.”

[AL-QURAN translated by Yosuf Ali]

So we are here with our some enemies, we have to live here for a certain period of time. Our Creator has told us the certain rules to obey during our stay on this planet. One set of such instructions given by Him, is called religion. The final set of instructions is named as "Islam". It was revealed through the messenger Muhammad (PBUH). And this is the topic of my thesis i.e. I have to organize the

1.1.1.1. Role Of Islamic Scholar In Society

Islamic scholar guides other people of society about the rules and regulations of Islam. People get help from them for their daily life problems. They give their opinion according to textbook of Islam i.e. Al-Quran and may also use the reference books of Hadith, which tell how the prophet himself spent his life.

Quran and Hadith are the only sources of knowledge of Islam. If by chance any matter is not present in both of these then many Muslim Scholars gather and discuss the matter and in the end give final decision. In all their discussions they use Quran and Hadith for their reference.

This research work is about building a system, which has the knowledge equivalent to a human Islamic scholar and will help the Islamic scholar as a "intelligent referencing system".

1.1.2 Scope

Building a complete Islamic expert system is not within the scope of my work. My scope is to provide a base for an expert system. Now this question is how much is this base, I don’t have the clear idea about the boundaries of this base. But roughly in two words it is the ‘knowledge representation’. The meanings of knowledge representation are thoroughly discussed in the next chapter. Simply you can say that it is the way of putting or organizing the knowledge in the system. So the major emphasis of this research work is on formatting the huge Islamic data in a system in such a way that it is said to be the knowledge. Once we have the knowledge it could be utilized in many ways. We might have a query handler on top of it or a tutor or any thing else. One example for utilization of this knowledge is a rule base query handler that would be implemented in this thesis, the software also have a small learner as an example.

What is out of scope — An advance query handler with a complete parser is out of the scope. An advance system learner, exhibiting the best techniques of learning is also not the part of this research work.

Representation of Islamic knowledge along with an example query handler and a small learner is the scope of this research work.

1.1.3 Objectives

My major objective is to provide a ground for future research in the field of artificial intelligence. The real life domain, which is the target of this work, is the religion Islam. But my objective of research is not domain specific, my aim is the actual implementation of AI concepts. Research on symbolic manipulation, knowledge keeping, knowledge acquisition, are of my interest.

There is nothing in the world which do not have a symbol associated with it as its name. Knowing of these names is the knowledge. Lisp is the best tool for symbolic manipulation, that is why I have used it in my implementation of research work and that it why it is widely used in the field of Artificial Intelligence. The other objective as a computer scientist, is that we should look beyond the simple static softwares, which are programmed once and could not meet the future needs. We should build such softwares, which could adopt themselves in the future without the external help.

My underlying objective is to make trend of AI in Pakistan. We should take interest in this area of software development so that we should be able to build real life systems by ourselves. AI application softwares are important for critical domains like defense and medicine. I want that Pakistan has its own AI experts in future for its computer software needs. No one in the world is going to help Pakistan in developing the systems for defense.

The ultimate goal of this work is an “intelligent referencing system” for a human Islamic scholar. With the help of this system the human scholar will be able to access the reference of Quran and Hadith more efficiently and intelligently.

1.1.4 Context of this research

Ever since the beginning of human life, man is trying to provide himself facilities. In effort of doing so he have made himself very busy but the positive aspect is that he have invented such tools which can help him to do his work. This is the code of his survival (as a mass population). Computer is one such tool. Its is an electronic device which can process and store data. It has many components specialized for specific purposes. Now humans are trying to make full use of it. Hardware (physical components) of computer have become advanced but they are useless without useful software (programs.) Artificial intelligence is a subject of computer science which is concerned with engineering of Intelligent software (software which exhibit intelligent behavior).

1.2 OUT LINE OF THESIS REPORT

This thesis report comprises of five chapters. Chapter1 is the introduction of the overall problem, scope, and objective of this work. It relates this work to the start of civilization.

Chapter 2 is the complete and through background. It discusses the concepts of AI which are the subject of this research work, like expert systems, knowledge representation, natural language understanding, and machine learning.

Chapter 3 is the conceptualization of this thesis before the implementation. It contains a brief introduction of Lisp, which is the tool of the practical implementation of this research work.

Chapter 4 is the complete documentation of the implementation of this research work as the software, which have been developed. This software is named as iexpert (Islamic expert system).

Chapter 5 has discussions about the end products (software ‘iexpert’ and this report). It will tell you how much I have succeeded in implementing the ideas of AI in my system? How much I have covered my scope and what objectives have possibly been achieved. It also discusses the future possibilities of the extension of this work.

At the end of this report an appendix is presented. Which contain some important algorithms, some Graphical User Interfaces of iexpert, statistics of the code of iexpert, and some other information of interest.

Chapter 2

BACKGROUND

2.1 INTRODUCTION

The field of Artificial Intelligence in computer science has its own taste. As it is related with the automation of intelligent behavior so we have to study intelligence. But the only intelligent thing which belong to this planet is human being, so most of the time we try to relate AI software with human behavior. Human behavior is assumed to be intelligent. In this subject we study how humans solve their problems, get out of trouble or learn anything. Then we try to program that thing in a computer science programming language. So far we have developed some theories, concepts, and techniques for transforming the human intelligence to software. This chapter addresses such concepts of AI which falls in the scope of this research work. These includes expert systems because we are making a system which would behave like a human Islamic scholar, then knowledge representation as obviously the human scholar have a huge knowledge of Islam so should be our system too. Next AI concept is natural language understanding, as human scholar communicate with others in a natural language therefore our system must also communicate with user through natural language. Last but important issue, which is the essential part of any AI system, is its learning portion. Although human scholar already have enough knowledge but he also learn more and more knowledge with the passage of time. So in last pages of this chapter contain some review of machine learning notes.

2.2 ARTIFICIAL INTELLIGENCE CONCEPTS

As in this research work the concepts of Artificial Intelligence are being mapped on the domain of religion Islam. This section gives you the general view of concepts of AI which have been used in the software of this research work. Remember that in this section the concepts are being discussed generally not in the specific context of Islam.

2.2.1 Expert Systems

Human experts are able to perform at high level because they know a great deal about their areas of expertise. This simple observation is the underlying rationale for the design of knowledge-based problem solvers. An expert system, as these programs are often called, uses domain specific knowledge to provide “expert quality” performance in a problem domain. Generally, expert system designers acquire this knowledge with the help of human domain experts, and the system emulates their methodology and performance. As with skilled humans, expert systems tend to be specialist, focussing on a narrow set of problems. Also, like humans, their knowledge is both theoretical and practical: the human experts that provide the system’s knowledge have generally augmented their own theoretical understanding of the problem domain with tricks, shortcuts, and heuristics for using that knowledge they have gained through problem solving experience. Unlike a human, however, current programs can not learn from their own experience: knowledge must be extracted from humans and encoded in a formal language. This is the major task facing designers of knowledge intensive problem solvers.

[Hashim 2000] says, Expert systems should not be confused with cognitive modeling programs, which attempt to simulate human mental architecture in details. Expert system neither copy the structure of human mind, nor are they mechanism for general intelligence. They are practical programs that use heuristics strategies developed by humans to solve specific classes of problems.

Because of the heuristic, knowledge intensive nature of expert level problem solving expert systems generally:

1. Support inspection of their reasoning processes, both in presenting intermediate steps and in answering questions about the solution process.
2. Allow easy modification, both in adding and in deleting skills form the knowledge base.

3. Reason heuristically, using (often imperfect) knowledge to obtain useful problem solutions.

According to [Steels 1985] the reasoning of an expert system should be open to inspection, providing information about the state of its problem solving, and explanations of the choices and decisions that the program is making. Explanations are important for several reasons: first, if a human expert such as a doctor or an engineer is to accept a recommendation from the computer, he/she must be satisfied the solution is correct. Indeed, human experts will accept advice from another human, let alone a machine, without understanding the justifications for it. This need to have answers explained is more than mistrust on the part of users: explanations help people relate the advice to their existing understanding of the domain and apply it in a more confidant and flexible manner.

Second, when a solution is open to inspection, we can evaluate every decision taken during the solution process, allowing for partial agreement and the addition of new information or rules to improve the performance. This help greatly in debugging the system and refining the knowledge base.

The exploratory nature of AI and expert system programming requires that programs be easily prototyped, tested, and changed. AI programming languages and environments are designed to support this iterative development methodology. In pure production system, for example, the modification of a single rule has no global syntactic side effects. Rules may be added or removed without requiring further changes to the larger programs. Expert system designers have commented that easy modification of the knowledge base is a major factor in producing a successful program [McDermott 1981].

[Wilensky 1986] says, the third feature of expert systems is their use of heuristic problem solving methods. Expert system designers have discovered, in formal “tricks of the trade” and “rules of thumb” are an essential complement to the standard theory presented in textbooks and classes. Sometimes these rules augment theoretical knowledge in understandable ways; often they are simply shortcuts that seem unrelated to the theory but have been empirically shown to work.

Expert systems have been built to solve a range of problems in domains such as medicine, mathematics, engineering, chemistry, geology, computer science, business, law, defense, and education. These programs have addressed a wide range of problem types; the following list, adopted from [Waterman 1986], is a useful summary of general expert system problem categories.

1. Interpretation — forming high level conclusions or descriptions from collections of raw data.
2. Prediction — projecting probable consequences of given situations.

3. Diagnosis — determining the cause of malfunctions in complex situations based on observable symptoms.

4. Design — finding a configuration of system components that meets performance goals while satisfying a set of design constraints.

5. Planning — devising a sequence of actions that will achieve a set of goals given certain starting conditions and runtime constraints.

6. Monitoring — comparing a system's observed behavior to its expected behavior.

7. Debugging and Repair — prescribing and implementing remedies for malfunctions.

8. Instruction — detecting and correcting deficiencies in students' understanding of a subject domain

9. Control — governing the behavior of a complex environment.

2.2.1.1 Basic Architecture

Figure 2.1 shows the most important modulus that make up an expert system. The user interacts with the expert system through a user interface that simplifies communication and hides much of the system complexity (e.g. the internal structure of the reasoning or decision making mechanism). Expert systems employ a variety of interface styles, including question-and-answer, menu-driven, natural language, or graphics interfaces, where the final decision on interface type is a compromise between user needs and the requirements of the knowledge base and interfacing system.

The heart of the expert system is the general knowledge base, which contain the problem solving knowledge of the particular application. In a frame-base expert system this knowledge is in the form of classes and objects, in a rule-base expert system this knowledge is represented in the form of if— then— rules, or the knowledge may be in the form of semantic nets depending upon the nature of the domain knowledge and problem. But the system should contain both general knowledge as well as case specific information.

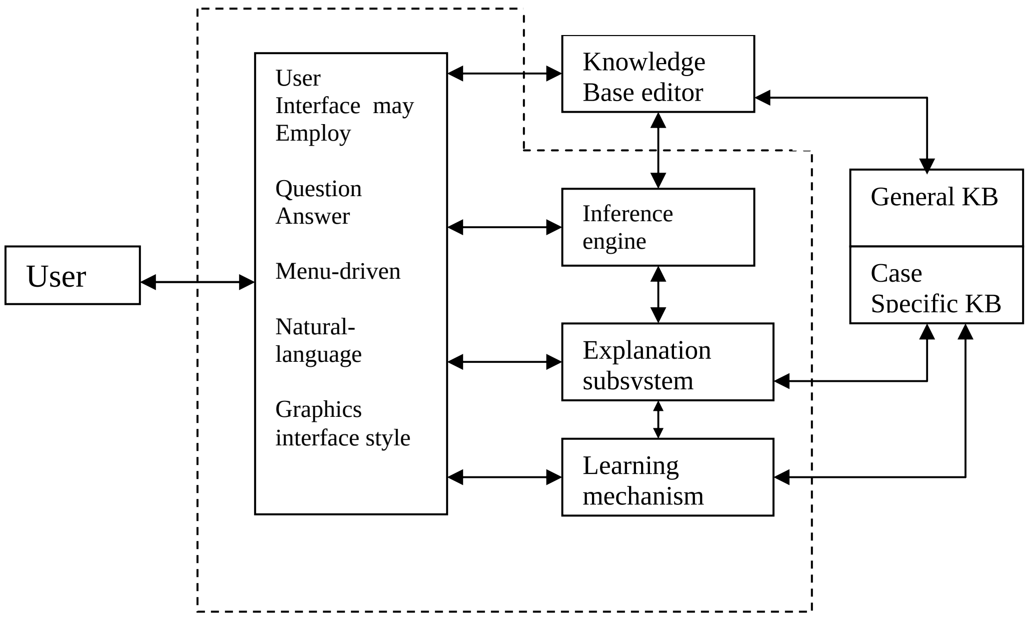


Figure 2.1 Architecture of a typical expert system

The inference engine applies the knowledge to the solution of actual problems. It is essentially an Interpreter for the knowledge base. In the production system, the inference engine perform the recognize--act control cycle. The procedures that implement the control cycle are separate from the production rules themselves. It is important to maintain this separation of the knowledge base and inference engine for several reasons:

1. The separation of the problem solving knowledge and the inference engine makes it possible to represent knowledge in a more natural fashion. If—then—rules, for example are closer to the way in which human being describe their own problem solving techniques than a program that embeds this knowledge in lower level computer code.

2. Because the knowledge base is separated from the program's lower level control structures, expert system builders can focus directly on capturing and organizing problem solving knowledge rather than on the details of its computer implementation.

3. The separation of knowledge and control, along with the modularity provided by rules and other representational structures like frames or scripts used in building knowledge bases, allows changes to be made in one part of the knowledge base without creating side effects in other part of the program.

4. The separation of the knowledge and the control elements of program allows the same control and interface software to be used in variety of systems. The expert system shell has all the components of Figure 2.1 except that the knowledge base and, of course, the case specific data contain no information. Programmers can use the “Empty Shell” and create a new knowledge base appropriate to their application. The broken lines of Figure 2.1 indicate the shell modulus.

5. This modularity allows us to experiment with alternative control regimes for the same system.

The program must keep track of the case-specific data: the facts, conclusions, and other information relevant to the all possible case of the problem.

The explanation subsystem allows the program to explain its reasoning to the user. These explanations include justifications for the system’s conclusions, explanations of why the system needs a particular piece of data and theoretical justifications of the program’s actions.

Many systems also include a knowledge-based editor. Knowledge-base editor helps the programmer locate and correct bugs in the program’s performance, often accessing the information provided by the explanation subsystem. Also may assist in the addition of new knowledge, help maintain correct rule syntax and perform consistency checks on the updated knowledge base.

2.2.2 Knowledge Representation

In building a knowledge base, a programmer must select significant objects and relations in the domain and map these into a formal language. The resulting program must contain sufficient knowledge to solve problems in the domain. It must make correct inferences from this knowledge and it must do so efficiently.

We can think of a knowledge base in terms of mapping between the objects and relations in a problem domain and the computational objects and relations in a program [Bobrow 1975]. The results of inferences on the knowledge base correspond to the results of actions or observations in the world. The computational objects, relations, and inferences available to programmers are determined by the knowledge representation language they select. The proper language can help the programmer acquire, organize, and debug the knowledge base.

There are general principles of knowledge organization that apply across a variety of domains and can be directly supported by a representation language. For example, class hierarchies are found in both scientific and common sense classification systems. How may we provide a general mechanism for representing them? How may we represent definitions, expectations? When should a knowledge based systems make default assumptions about missing information and how should it adjust its reasoning if these assumptions prove wrong? How may we best represent time? Causality? Uncertainty? Progress in knowledge based systems depends on discovering the principles of knowledge organization and supporting them in higher level representational tools.

It is useful to distinguish between a representational scheme and the medium of its implementation [Hayes 1974]. This is similar to the distinction between data structure and programming languages. Programming languages are the medium of implementation, the data structure is the scheme. Generally, knowledge representation languages are more constrained than predicate calculus or programming languages such as Lisp and Prolog. The constraints take the form of explicit structure for representing categories of knowledge. The medium in which they are implemented might be Prolog, Lisp or a conventional language such as C, C++, or Java.

Over the past 25 years, numerous representational schemes have been proposed and implemented each of them having its own strengths and weaknesses. Mylopoulos and [Levesque 1984] have classified these into four categories:

1. **Logical Representation Schemes** — this class of representations uses expressions in formal logic to represent a knowledge base. Inference rules and proof procedures apply this knowledge to problem instances. First order predicate calculus is the most widely used logical representation scheme, but it is only one of a number of logical representation [Turner 1984]. Prolog is an ideal programming language for implementing logical representation schemes.
2. **Procedural Representational Schemes** — procedural schemes represent knowledge as a set of instructions for solving a problem. This contrasts with the declarative representations provided by logic and semantic networks. In a rule-based system, for example, an if—Then— rule may be interpreted as a procedure for solving a goal in a problem domain, by solving the premises in order. A production system may be seen as an example of a procedural representation scheme.
3. **Network Representation Schemes** — Network representations capture knowledge as a graph in which the nodes represent objects or concepts in the problem domain and the arc representations are associations between them. Examples of network representations include semantic networks, conceptual dependencies, and conceptual graphs.

4. **Structured Representation Schemes** — Structured representation languages extend networks by allowing each node to be a complex data structure consisting of named slots with attached values. These values may be simple numeric or symbolic data, pointers to other frames, or even procedures for performing a particular task. Examples of structured representations include Scripts, frames, and objects.

A related question concerns about the granularity of the representation. In predicate calculus and semantic networks we can denote objects in the domain only by using simple symbols. Other representations, such as frames and objects, let us define complex structures by encapsulating multiple features and attributes to indicate a single object in the domain. For example, a person is a single entity that requires many properties for its description. This has the advantage of placing all the information about an object in one place, making it easier to access, update, or delete.

How can representations distinguish between intentional and extensional knowledge? Briefly the extension of a concept is the set of all things denoted by a given concept; for example "ball" has its extension the set of all balls. The intention determines what a concept means in the abstract, such as the definition of a ball in terms of its roundness, its ability to roll, bounce, and be thrown, or its use in games. Though these two different definitions characterize the same set of objects, their role in understanding is very different. This issue is important if we are linking an expert system and a database. The database enumerates objects and relations of a domain (in extension), whereas the expert system includes the intentional knowledge used to reason about these objects.

How can we best represent meta-knowledge, or knowledge about the objects and structures of the representation language itself? This is important if we wish to reason about knowledge, as in describing the sentence “George believes that Tom knows that Mary’s car is red.” Knowledge base editors, such as Teiresias for MYCIN use meta-knowledge to reason about knowledge and proofs.

Another important feature of knowledge is its organization into class hierarchies. The ability to represent the relationship between an object and its class or between a class and its super class has proved so useful that many representation languages include mechanisms that define these relationships. Inheritance is a relation by which an individual assumes the properties of its class and by which properties of a class are passed on to its subclass.

Inheritance is also used to implement default values and exceptions. For example we may state that all chimpanzees live in the jungle, except Bonzo, who lives in circus. Default values are simply inherited from the appropriate super class. We can represent exceptions to these defaults by redefining the property at a lower level in the hierarchy.

Finally, it is often helpful to attach procedures to object descriptions. These procedures, sometimes referred to as methods or demons, are executed when an object is changed or created and provide a vehicle for implementing graphics, I/O, consistency checks, and interaction between objects.

2.2.2.1 Network Representation

Logical representations developed by the efforts of philosophers and mathematicians to characterize the principles of correct reasoning. The major concern of this work is the development of formal representation languages with sound and complete inference rules. An alternative line of research has grown out of the efforts of psychologists and linguists to characterize the nature of human understanding. This work is less concerned with establishing a science of correct reasoning than with describing the way in which humans actually acquire and use knowledge of their world. As a result, this work has proved particularly useful to the AI application areas of natural language understanding and commonsense reasoning.

Associative theory defines the meaning of an object in terms of a network of associations with other objects in a mind or a knowledge base. Although symbols denote objects in a world, this denotation is mediated by our store of knowledge. When we perceive and reason about an object, that perception is first mapped into a concept in our minds. This concept is part of entire knowledge of the world and is connected through appropriate relationships to other concepts. These relationships form an understanding of the properties and behavior of object.

There is psychological evidence that in addition to their ability to associate concepts, humans also organize their knowledge hierarchically, with information kept at the highest appropriate levels of taxonomy. Inheritance systems allow us to store information at the highest level of abstraction, which reduces the size of knowledge bases and helps prevent update inconsistencies. For example if we are building a knowledge base about birds, we can define the traits common to all birds, such as flying or having feathers, the general class bird and allow a particular species of birds to inherit these properties. This reduces the size of knowledge base by requiring their assertion for every individual. Inheritance also helps us to maintain the consistency of knowledge base when adding new classes and individuals.

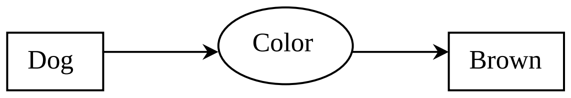
Graphs, by providing a means of explicitly representing relations using arcs and nodes, have proved to be an ideal vehicle for formalizing associative theories of knowledge. A *semantic network* represents knowledge as a graph, with the nodes corresponding to the facts or concepts and the arcs to relations or associations between concepts. Both nodes and links are generally labeled. The inferences are made by following the appropriate links to related concepts. semantic networks also implement inheritance.

The term “semantic network” encompasses a family of graph-based representations. These representations differ chiefly in the names that allowed for nodes and links and the inferences that may be performed on these structures.

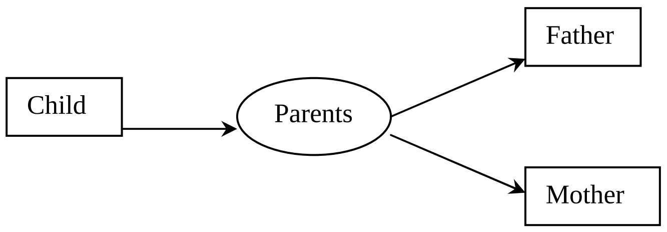
2.2.2.2 Conceptual Graphs: A Network Representation Language

A number of network languages have been developed to model the semantics of natural language and other domains. John Sowa’s conceptual graph is an example of a modern network representation language. We define rules for forming and manipulating conceptual graphs and the conventions for representing classes, individuals, and relationships.

A conceptual graph is a finite, connected, bipartite graph. The nodes of the graph are either concepts or conceptual relations. Conceptual graphs do not use labeled arcs; instead the conceptual relation nodes represent relations between concepts. Because conceptual graphs are bipartite, concepts can only have arcs to conceptual relations, and vice versa. In figure 2.2 dog and brown are the concept nodes and color is a conceptual relation. To distinguish between these types of nodes, we represent concepts as boxes and conceptual relations as ellipses.



a. Color is 2-ary relation.



b. Parents is a 3-ary relation.

Figure 2.2 Conceptual graph examples

In conceptual graphs concept nodes represent either concrete or abstract objects in the world of discourse. Concrete concepts, such as a cat, telephone, or restaurant, are characterized by our ability to form an image of them in our minds. Note that concrete concepts include generic concepts such as cat or restaurant along with concepts of specific cats and restaurants. We can still form an image of a generic cat. Abstract concepts include things such as love, beauty, and loyalty that do not correspond to images in our mind.

Conceptual relation nodes indicate a relation involving one or more concepts. One advantage of formulating conceptual graphs as bipartite graphs rather than using labeled arcs is that it simplifies the representation of relations of any arty. A relation of arty n is represented by a conceptual relation node having n arcs. As shown in Figure 2.2 b.

Each conceptual graph represents a single proposition. A typical knowledge base will contain a number of such graphs. Graphs may be arbitrary complex but must be finite. In conceptual graphs, every concept is a unique individual of a particular type. Each concept box is labeled with *type* label, which indicates the class or type of individual represented by that node.

The theory of conceptual graphs includes a number of operations that allow us to form new graphs from existing graphs. These allow for the creation of a new graph by either specializing or generalizing an existing graph. This is done through four operations called copy, restrict, join, and simplify.

The copy rule allows us to form a new graph, which is exactly copy of the original graph. Restriction rule allows concept nodes in a graph to be replaced by a node representing their specialization. There are two cases: one if a concept is labeled with a generic marker, the generic marker may be replaced by an individual marker.

Secondly, a typical label on a concept may be replaced by one of its subtypes, as long as this is consistent with the referent of concept.

Join rule lets us combine two graphs into single graph. Join is a restriction rule because the resulting graph is less general than either of its components. If a graph contains duplicate relations, then one of them may be deleted, along with all its arcs. This is simplify rule. Duplicate relations often occur as the result of a join operation.

Conceptual graphs include a concept type, *proposition* that takes a set of conceptual graph as its referent and allows us to define relations involving propositions. Prepositional concepts are indicated as a box that contains another conceptual graph. These proposition concepts may be used with appropriate relations to represent knowledge about proposition.

2.2.2.3 Structured Representation: Frames

Using network representations, we view knowledge as organized using explicit links or associations between objects in the knowledge base. Alternatively, we can organize knowledge into more complex units that represent situations or objects in the domain these units are called frames or schemas.

According to [Rich 1982] a frame may be viewed as a static data structure used to represent well-understood stereotyped situations. Frame like structures seem to organize out own knowledge of the world. We adjust eve new situations by calling up information structured by past experiences. We then specially fashion or revise the details of these past experiences to represent the individual differences for the new situation.

[Tufail 1999] says frames, as well as object-oriented systems, provide us with a vehicle for this organization, representing knowledge as structured objects consisting of named slots with attached values. Each individual frame may be seen as a data structure, similar in many respects to the traditional "record," that contains information relevant to stereotyped entities. The slots in the frame contain information such as:

1. Frame identification information.
2. Relationship of this frame with other frames.
3. Description of requirements for frame match. A chair, has its seat between 20 and 40 cm from the floor, its back higher than 60 cm, etc. these requirements may be used to determine when new objects fit the stereotype defined by the frame.
4. Procedural information on use of the structure described. An important feature of frame is the ability to attach procedural code to a slot.

5. Frame default information. These are slot values that are taken to be true when no evidence to the contrary has been found. For instance, chairs have four legs, etc.

6. New instance information. Many frame slots may be left unspecified until given a value for a particular instance or when they are needed for some aspect of problem solving.

The presence, absence, or amount of detail in each of the six slots above depends on the particular problem-solving situation addressed. Frames extend semantic networks in a number of important ways. In the network version, there is simply a collection of nodes and we depend more on our interpretation of the structure. This ability to organize our knowledge into such structures is an important attribute of a knowledge base.

Frames make it easier to organize our knowledge hierarchically. In a network, every concept is represented by nodes and links at the same level of specification. Very often, however, we may like to think of an object as a single entity for some purposes and only consider details of its internal structure for other purposes.

Procedural attachment is a particularly important feature of frames because certain knowledge does not adapt well to declarative representations. For example, we may want to include the ability to generate graphic images in a knowledge base. A graphical language is more appropriate for this than a network language. We use procedural attachment to create demons. A demon is a procedure that is invoked as a side effect of some other action in knowledge base. For example, we may wish the system to perform type checks or to run consistency tests whenever a certain slot value is changed.

Frame systems support class inheritance. The slots and default values of a class frame are inherited across the class/subclass and class/member hierarchy. When an instance of the class frame is created, the system will attempt to fill its slots, either by querying the user, accepting the default value from class frame, or executing some procedure or demon to obtain the instance value.

Frames add to the power of semantic nets by allowing complex objects to be represented as a single frame, rather than as a large network structure. This also provides a natural way to represent stereotypic entities, classes, inheritance, and default values. Although frames, like logical and network representations, are a powerful tool, many of the problems of acquiring and organizing a complicated knowledge base must still be solved by the programmer’s skill and intuition.

2.2.2.4 Scripts

A natural language understanding program must use a large amount of background knowledge to understand even simplest conversation. There is evidence that humans organize this knowledge into structure corresponding to typical situations. There is also a tendency for errors in understanding when the subject of a conversation changes abruptly, presumably because we are confused over which context or schema to use in resolving pronoun references and other ambiguities in the conversation.

A script is a structured representation describing a stereotyped sequence of events in a particular context. The script was originally designed by Schank and his research group as a means of organizing conceptual dependency structures into descriptions of typical situations. Scripts are used in natural language understanding systems to organize a knowledge base in terms of the situations that the system is to understand.

The components of a script are:

1. Entry conditions or descriptors of the world that must be true for the script to be called. In a restaurant script, these include a restaurant that is open and a customer who is hungry.

2. Results or facts that are true once the script has terminated; for example, the customer is full and poorer, the restaurant owner has more money.
3. Pops or the “things” that support the content of the script. These might include tables, waiters, and menus. This allows reasonable default assumptions about the situation; a restaurant is assumed to have tables and chairs unless stated otherwise.
4. Roles are the actions that the individual participants perform. The waiter takes orders, delivers food, and presents the bill. The customer orders, eats and pays.
5. Scenes. Schank breaks the script into a sequence of scenes, each of which presents a temporal aspect of the script.

The elements of the scripts, the basic “pieces” of semantic meaning, are represented using conceptual dependency relationships. Placed together in a frame like structure, they represent a sequence of meanings, or event sequence.

The program reads a small story about restaurants and parses it into an internal conceptual dependency representation. Because the key concepts in this internal description match with the entry conditions of the script, the program binds the people and things mentioned in the story to the roles and props mentioned in the script. The result is an expanded representation of the story contents, using the script to fill in any missing information and default assumptions. The program then answers questions about the story by referring to the script. The script allows the reasonable default assumptions that are essential to natural language understanding.

2.2.3 Natural Language Understanding

We not only gather information with language, but we use it for communication as well. It is probably language, which most clearly identifies us as a human. Target is to make a system, which can 'understand', oral as well as written language. According to [Charniak 1999] understanding means “transformation of the information provided by the outside world and translate it into a precise internal representation”.

2.2.3.1 Levels Of Language Analysis

Linguists have defined different levels of analysis for natural language.

1. *Prosody:* It deals with the rhythm and speech pattern of language. This level of analysis is difficult to analyze and often neglected; however, its importance is evident in the powerful effect of poetry or religious chants, as well as the role played by rhythm in children’s wordplay and the babbling of infants.

2. *Phonology / acoustic-phonetic:* It is responsible for taking the sounds (presumably represented as a plot of how much energy is coming in at various sound frequencies) and translating the input into words.

3. *Morphological analysis:* It is concerned with the components (morphemes) that make up words. These include the rules governing the formation of words, such as the effect of prefixes (un- , anti- , non- etc) and suffixes (-ing , -ly etc) that modify the meaning of root words. Morphological analysis is important in determining the role of a word in a sentence, including its tense, number, and part of speech.

4. *Syntactic analysis:* Linear sequences of words are transformed into structures that show how the words relate to each other. Some word sequences may be rejected if they violate the language's rules (syntax of the language) for how words may be combined.

5. *Semantic analysis:* The tree like structures created by the syntactic analyzer are assigned meanings. In other words, a mapping is made between the syntactic structures and objects in the task domain. Structures for which no such mapping is possible represents meaningless phrases and may be rejected.

6. *Discourse integration:* The meaning of an individual sentence may depend on the sentences that precede it and may influence the meanings of the sentences that

follow it. For example, the word ‘it’ in the sentence “John wanted it,” depends on the prior discourse context, while the world ‘John’ may influence the meaning of later sentences (such as “he always had.”)

7. *Pragmatic analysis:* The structure representing what was said is reinterpreted to determine what was actually meant. For example, the sentence “Do you know what time it is?” should be interpreted as a request to be told the time.

2.2.3.2 An Example

As an example we apply all the analysis stages on the sentence given bellow and see if system based on the above stages can really understand it.

I want to print Bill’s .init file.

Morphological analysis: Morphological analysis will do the following things:

- Pull apart the word “Bill’s” into proper noun “Bill” and the possessive suffix “’s”.
- Recognize the sequence “.init” as a file extension that is functioning as an adjective in the sentence.

In addition, this process will usually assign syntactic categories to all the words in the sentence. This is usually done because interpretation for affixes (prefixes and suffixes) may depend on the syntactic category of the complete word. For example, consider the word “prints.” This word is either a plural noun (with the “-s” marking plural) or a third person singular verb (as in “he prints”), in which case the “-s” indicates both singular and third person.

Syntactic Analysis: Syntactic analysis must exploit the results of morphological analysis to build a structural description of the sentence. The goal of this process, called *parsing*, is to convert the flat list of words that forms the sentence into a structure that defines the units that are represented by that flat list. For our example sentence, the result of parsing is shown in figure 2.3.

What is important here is that a flat sentence has been converted into a hierarchical structure and that the structure has been designed to correspond to meaning units when semantic analysis is performed. One useful thing we have done here, although not all-syntactic systems do, is create a set of entities we call *reference markers.* They are shown in parenthesis in the parse tree. Each one corresponds to some entity that has been mentioned in the sentence. These reference markers are useful later since they provide a place in which to accumulate information about the entities as we get it. Thus although we have not tried to do semantic analysis (i.e.

assign meaning) at this point, we have designed our syntactic analysis process so that it will find constituents to which meaning can be assigned.

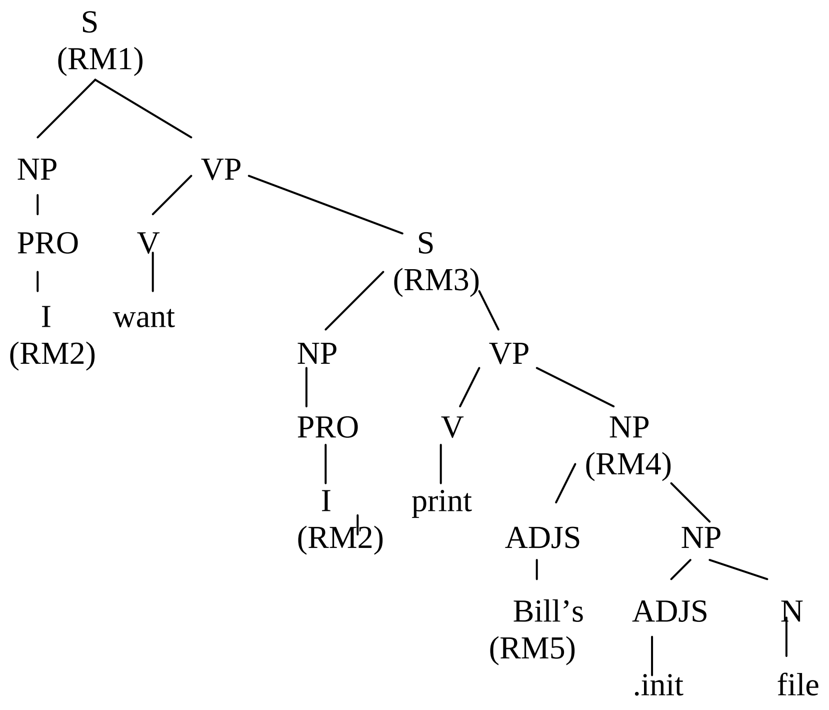


Figure 2.3 Syntactic Analysis of "I want to print Bill's .init file."

Semantic Analysis: Semantic analysis must do two important things:

- It must map individual words into appropriate object in the knowledge base or database.
- It must create the correct structures to correspond to the way the meanings of the individual words combine with each other.

For this example, suppose that we have a frame-based knowledge base that contains the units shown in figure 2.4. Then we can generate a partial meaning, with respect to that knowledge base, as shown in Figure 2.4. Reference marker RM1 corresponds to the top-level event of the sentence. It is a wanting event in which the speaker (denoted by “I”) wants a printing event to occur in which the same speaker prints a file whose extension is “.init” and whose owner is Bill.

Discourse Integration: Till now we have figured out what kinds of things this sentence is about. But we do not yet know which specific individuals are being referred to. Specifically, we do not know to whom the pronoun “I” or the proper noun “Bill” refers. To pin down these references requires an appeal to a model of the current discourse context, from which we can learn that the current user (who typed the word “I”) is *user12* and that the only person named “Bill” about whom we could be talking is *user73*. Once the correct referent for Bill is known, we can also determine exactly which file is being referred to: *F1* is only file with the extension “.init” that is owned by Bill.

User

isa :	*Person*
*loginame :	must be <string>
*password :	must be <string>

File-Struct		
	isa :	*Information-object*
Printing		
	isa :	*Physical-Event*
	**agent :*	must be <animate or program>
	**object :*	must be <information-object>
Wanting		
	isa :	*Mental-event*
	**agent :*	must be <animate>
	**object :*	must be <state or event>
Commanding		
	isa :	*Mental-event*
	**agent :*	must be <animate>
	**performer :*	must be <animate or program>
	**object :*	must be <event>
User12		
	instance :	*User*
	loginame :	John
	password :	hello
User78		
	instance :	*User*
	loginame :	Bill-Smith
	password :	smith
F1		
	instance :	*File-Struct*
	name :	stuff
	extension :	.init

	owner :	User12
	in-directory :	/wsmith/
This-system		
	instance :	*AI-Program*

Figure 2.4 A Knowledge Base Fragment

RM1			{the whole sentence}	
	instance :	***Wanting***		
	agent :	RM2	{ I }	
	object :	RM3		{a printing event }
RM2			{ I }	
RM3				
	instance :	*Printing*		
	agent :	RM2	{ I }	
	object :	RM4		{ Bill's .init file }
RM4				
	instance :	File-Struct		
	extension :	.init		
	owner :	RM5		{ Bill }
RM5				
	instance :	*Person*		
	first-name :	Bill		

Figure 2.5 A Partial Meaning for a Sentence

Pragmatic Analysis: we now have a complete description, in the terms provided by our knowledge base, of what was said. The final step toward effective understanding is to decide what to do as a result. One possible thing to do is to record what was said as a fact and be done with it. For some sentences, whose intended effect is clearly declarative, that is precisely the correct thing to do. But for other sentences, including this one, the intended effect is different. we can discover this intended effect by applying a set of rules that characterize cooperative dialogues. In this example, we use the fact that when the user claims to want something that the system is capable of performing, then the system should go ahead and do it. This produces the final meaning shown in Figure 2.6.

The final step in pragmatic processing is to translate, when necessary,(from the knowledge-base) a command to be executed by the system. In this case, this step is necessary, and we see that the final result of the understanding process is

Lpr/wsmith/stuff.init

Where “lpr” is the operating system’s file print command.

Meaning

Instance :	*Commanding*
Agent :	User12
Performer :	This-system
Object :	P1

P1

Instance :	*Printing*
Agent :	This-system
Object :	F1

Figure 2.6: Representing the Intended Meaning

2.3.3.3 Syntactic Parsing

Syntactic parsing is the step in which a flat input sentence is converted into a hierarchical structure that corresponds to the units of meaning in the sentence. This process is called *parsing.* It plays an important rule in many natural language understanding systems for two reasons:

- Semantic processing must operate on sentence constituents. Syntactic parsing is computationally less expensive than is semantic processing. Thus it can play a significant role in reducing the overall complexity.
- Some times it is not possible to extract the meaning of a sentence without using grammatical facts.

Although there are many ways to produce a parse tree, almost all the systems that are actually used have two main components:

- A declarative representation, called a *grammar,* of the syntactic facts about the language
- A procedure called a *parser,* which compares the grammar against input sentences to produce parsed trees.

2.2.3.3.1 Context Free Grammar

The most common way to represent grammars is a set of production rules. Figure 2.7, shows a simple context-free grammar for English. Read the first rule as, "A sentence is composed of a noun phrase followed by a verb phrase." The vertical bar should be read as "or." The ε denotes the empty string. Symbols that are further expanded by rules are called *nonterminal symbols.* The symbols that correspond directly to strings that must be found in an input sentence are called *terminal symbols.*

A context free-grammar allows rules to have only a single nonterminal on their left-hand side. Consequently, the rule may be applied to any occurrence of that symbol, regardless of its context. Though context-free grammars have proved to be a powerful tool for defining programming languages and other formalisms in computer science, there is reason to believe that they are not powerful enough, by themselves, to represent the rules of natural language syntax.

The parsing process takes the rules of the grammar and compares them against the input sentence. Each rule that matches adds something to the complete structure that is being built for the sentence. The structure to build is a *parse tree*, which simply records the rules and how they are matched.

S $\rightarrow$ **NP VP**
NP → the **NP1**
NP → **PRO**
NP → **PN**
NP → **NP1**
NP1 → **ADJS N**
ADJS → $\varepsilon \mid$ **ADJ ADJS**
VP → **V**
VP → **V NP**
N → file | printer
PN → Bill
PRO → I
ADJ → short | long | fast
V → printed | created | wanted

Figure 2.7 A simple Grammar for a Fragment of English.

Here **S** is the start symbol all the bold faced are *non-terminals* and others are terminals.

2.2.3.3.2 Top-down and Bottom-up Parsing

To parse a sentence, it is necessary to find a way in which that sentence could have been generated from the start symbol. There are two ways that this can be done

- *Top-Down Parsing*--- Begin with the start symbol and apply the grammar rules forward until the symbols at the terminals of the tree correspond to the components of the sentence being parsed.
- *Bottom-UP Parsing* --- Begin with the sentence to be parsed and apply the grammar rules backward until a single tree whose terminals are the words of the sentence and whose top node is the start symbol, has been produced.

The choice between these two approaches is similar to the choice between forward and backward reasoning in other problem-solving tasks.

2.2.3.3.3 Transition Network Parsers

A transition network parser represents grammar as a set of finite-state machines or transition networks. Each network corresponds to a single nonterminal in the grammar. Arcs in the networks are labeled with either terminal or nonterminal symbols. Each path through the network, from the start state to the final state, corresponds to some rule for that nonterminal; the sequence of arc labels on the path is the sequence of symbols on the right-hand side of the rule. See figure 4.28 and 4.29 for reference.

2.2.3.4 Query Handling

Some systems have query handlers for answering the user queries but the don't necessarily have a parser. They work on the basis of pattern matching. They have some built in templates of questions which user can ask. When user enters a query the input statement is pattern match against those templates. Useful information are extracted from that statement that the answer builder portion of the system build the answer from the knowledge base. Knowledge may be in the form of rules or frames. We can't say that such systems literally understand the query. It depends how we define 'understanding'. If understanding is the mapping of input to the internal representation then such systems (systems without parser only based on pattern matching) do understanding although may be right or wrong. But if understanding is something else then they are dumb.

2.2.3.4.1 ELIZA

In 1960's Eliza held a conversation with the user in which it simulated a psychotherapist. It takes a plain English string as input and return the answer also in the form of English sentence. It make heavy use of a technique called pattern matching and seems to be intelligent. But its intelligence is illusion.

[Norvig 1992] says Eliza was one of the first programs to feature English output as well as input. Eliza was designed to be instructed by a variety of scripts, or patterns of interaction. Eliza was aimed to behave like a passive psychoanalyst i.e. it let the patient reveal himself. This is easy for Eliza because it involves being passive and reacting, rather than volunteering new information. In the listing that follows, the user's input is in mixed case, and Eliza's response is in capitals:

Men are all alike.

IN WHAT WAY

They're always bugging us about something or another.

CAN YOU BE MORE SPECIFIC

Well my boyfriend made me come here.

YOUR BOYFRIEND MADE YOU COME HERE

He said I am depressed much of the time.

I AM SORRY TO HEAR YOU ARE DEPRESSED

It's true I am unhappy.

DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY

I need some help. That much seems certain.

WHAT WOULD IT MEAN TO YOU IF YOU GOT HELP

Perhaps I could learn to get along with my mother.

TELL ME MORE ABOUT YOUR FAMILY

This seems quite impressive; the program is responding reasonably to a variety of inputs and appears to be exhibiting true "understanding". But this appearance is an illusion, the program has finessed the difficult task of understanding by carefully recognizing, transforming, and echoing pieces of the input. The following parts of natural language understanding are totally absent in Eliza:

1. Eliza don't do anything else when a pattern is matched with the input string except that it return the response. It should have carried out some actions inside the system so the proper understanding and proper answering could be possible. Although it don't have any parser but many steps of natural language understanding are possible.

2. No discourse integration —There is no resolving the references like IT, HE, SHE, THEY , THIS , THAT etc.

3. There is no using of different rule sets while discussion proceeds. Eliza use single rule set it means or one input sentence it has few responses, out of which one is randomly selected at one time.

4. No morphological analysis — Resolution of suffixes like -ly, s, 's, -ing, and prefixes like anti-, un-.

5. Its rules or templates for handling the input sentences remain constant. It means if program do not answer anything, it would never be able to answer it in future. System should be able to add/write new rules on its own at runtime by observing the language/inputs of users.

6. No spell tolerance — If user make a spell mistake, program would never be able to recognize that word since it is doing pattern matching blindly.

7. Other defect in Eliza is that it doesn't have any mechanism for selecting the best-suited template, which are currently matching the input string. It may collect all the rules matching then verify from the user one by one and execute that rule which is verified.

2.2.4 MACHINE LEARNING

The ability to learn must be part of any system that would claim to possess general intelligence. Indeed, in our world of symbols and interpretation, the very notion of an unchanging intellect is a contradiction in terms. Intelligent systems must be able to change through the course of their interactions with the world, as well as through the experience of their own internal states and processes.

Simple definition of machine learning as given by [Sawar 1992] is as follow;

"The learning process demonstrated by intelligent systems is termed as 'machine learning'. It involves the task of making changes to the knowledge base, in order to accommodate new circumstances and, if possible improve the system's performance."

Learning is important for practical applications of artificial intelligence. [Feigenbaum 1983] have called the "knowledge engineering bottleneck" the major obstacle to the widespread use of expert systems. This bottleneck is the cost and difficulty of building expert systems using traditional knowledge acquisition techniques. One solution to this problem would be for programs to begin with a minimal amount of knowledge and learn from example, high-level advice, or their own explorations of the domain. [Simon 1983] provides a precise definition of machine learning;

"Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task drawn from the same population more effectively the next time."

This definition suggests many of the issues involved in developing programs that learn. One important issue is the under constrained nature of empirical learning. Learning involves generalization from experience: performance should improve not only on the "repetition of the same task," but also on similar tasks in the domain. Because interesting domains tend to be large, a learner may only examine a fraction of all possible examples; from this limited experience, the learner must acquire knowledge that will generalize correctly to unseen instances of the domain. This is the problem of induction, and it is central to learning. In most learning problems, no matter what algorithm the learner uses. Learning algorithm must generalize heuristically: they must select those aspects of their experience that are most likely to prove effective in the future. Such selection criteria are known as inductive biases.

[Simon 1983] definition describes learning as allowing the system to perform better the next time. Selecting the possible changes to a system that will allow it to improve is a difficult task. Learning research must address the possibility that changes may actually degrade performance. Preventing and detecting such problems is another problem for a learning algorithm. Learning involves changes in the learner; this is obvious. However, the exact nature of those changes and the best way to represent them are far from obvious.

Learning covers a wide range of phenomena. At one end of the spectrum is skill refinement. People get better at many tasks simply by practicing. At the other end of spectrum lies knowledge acquisition. As many AI programs depends heavily on knowledge. Knowledge is generally acquired through experience.

Knowledge acquisition itself includes many different activities. Simple storing of computed information, or *rote learning* is the most basic activity. Many computer programs, e.g. database systems, can be said to “learn” in this sense, although most people would not call such simple storage learning. However, many AI programs are able to improve their performance substantially through rote-learning techniques.

Another way we learn is through taking advice from others. Advice taking is similar to rote learning, but high-level advice may not be in a form simple enough for a program to use directly in problem solving. The advice may need to be first operationalized.

People also learn through their own problem-solving experience. After solving a complex problem, we remember the structure of the problem and the methods we used to solve it. The next time we see the problem, we can solve it more efficiently. Moreover we can generalize from our experience to solve related problems more easily. In contrast to advice taking, learning from problem-solving experience does not usually involve gathering new knowledge that was previously unavailable to the learning program. That is, the program remembers its experiences and generalizes from them, but does not add to the transitive closure of its knowledge. Transitive closure of a program’s knowledge is that knowledge plus whatever the program can logically deduce from it. In large problem spaces efficiency gains are critical. Practically speaking, learning can mean the difference between solving a problem rapidly and not solving it at all. In addition, the programs that learn through problem-solving experience may be able to come up with qualitatively better solutions in the future.

Another form of learning that does involve stimuli from the outside is *learning from examples.* We often learn to classify things in the world without being given explicit rules. For example, adults can differentiate between cats and dogs, but small children often cannot. As [Bain 1986] specify the major shortcoming of expert systems;

“The inability to save accounts of pervious experiences for future application and modification represents a serious shortcoming of most, if not all, rule-based expert systems”

For example-based learning system have to store all kinds of examples both positive and negative. Then it should have some sort of classification on the basis of similarities, complexities or any thing else. System may have indexing of these examples. Then other modules of learner learn from these examples resulting in increase of efficiency or modification of knowledge base. As described previously this is the inductive method of learning and is the core of learning methodologies.

2.3 SUMMARY

This chapter has given you an overview of some basic concepts of AI. In these last lines of chapter we revise the headings and definitions. First of all the term Expert systems, it is used for such AI systems which have the skill comparable to the human experts of any domain. The basic architecture of an expert system comprises of a knowledge base, learning and explanation modules, an inference engine and a user interface. Each of these parts may have several subparts or several types. Then is the issue of knowledge representation. An expert system has huge knowledge formatted in such a way that it can be distinguished from an ordinary database of other systems. There are various techniques for putting the knowledge in a system like frames, semantic nets, and scripts as discussed in previous pages. User interaction with the system could be through natural language. There are several stages of language understanding like morphological analysis, syntactic analysis, semantic analysis, discourse integration, and pragmatic analysis. Morphological analysis is concerned with the anatomy of words. Syntactic analysis checks the grammar of the sentence. Semantic analysis addresses the issue of meanings of a sentence. In discourse integration references are resolved. Pragmatic analysis analyzes the input sentence for the intended meanings of the sentence producer. And in the last few pages, machine learning has been discussed. There are different ways through which humans learn and a system could learn like rote learning, example-based learning, learning-by-advice, explanation-based learning, exploration-based learning, failure-driven learning etc.

Chapter 3

CONCEPTUALIZATION

3.1 INTRODUCTION

This chapter will explain you the implementation of Islamic expert system. It will discuss each concept of AI given in the previous chapter, one by one. This chapter is divided into six major parts. First part is about the tool I have used for this system, it gives the brief introduction of the tool. Then there is an overall introduction of the system, which has been developed. It is named as iexpert (Islamic expert system). The next four portions of this chapter are actually the modules of iexpert namely, security, knowledge base, query handler, and learner. Each module is discussed in detail. In the end there is summary, which encapsulates the conceptual implementation of this research work.

3.2 IEXPERT

Iexpert is acronym for Islamic Expert system, which has been developed. You can consider it a Intelligent, efficient, referencing system. It will help the human Islamic scholar to see the references from Quran Hadith and to store these references for future decision making. System is currently having two complete Quran Translations one translated by Pickthall and other translated by Yosuf Ali. System also contains one

English translation of Hadith book AL-BUKHARI. There is a dummy English dictionary of about fifty words, for testing that the system can reference the dictionary for unrecognized words by itself automatically.

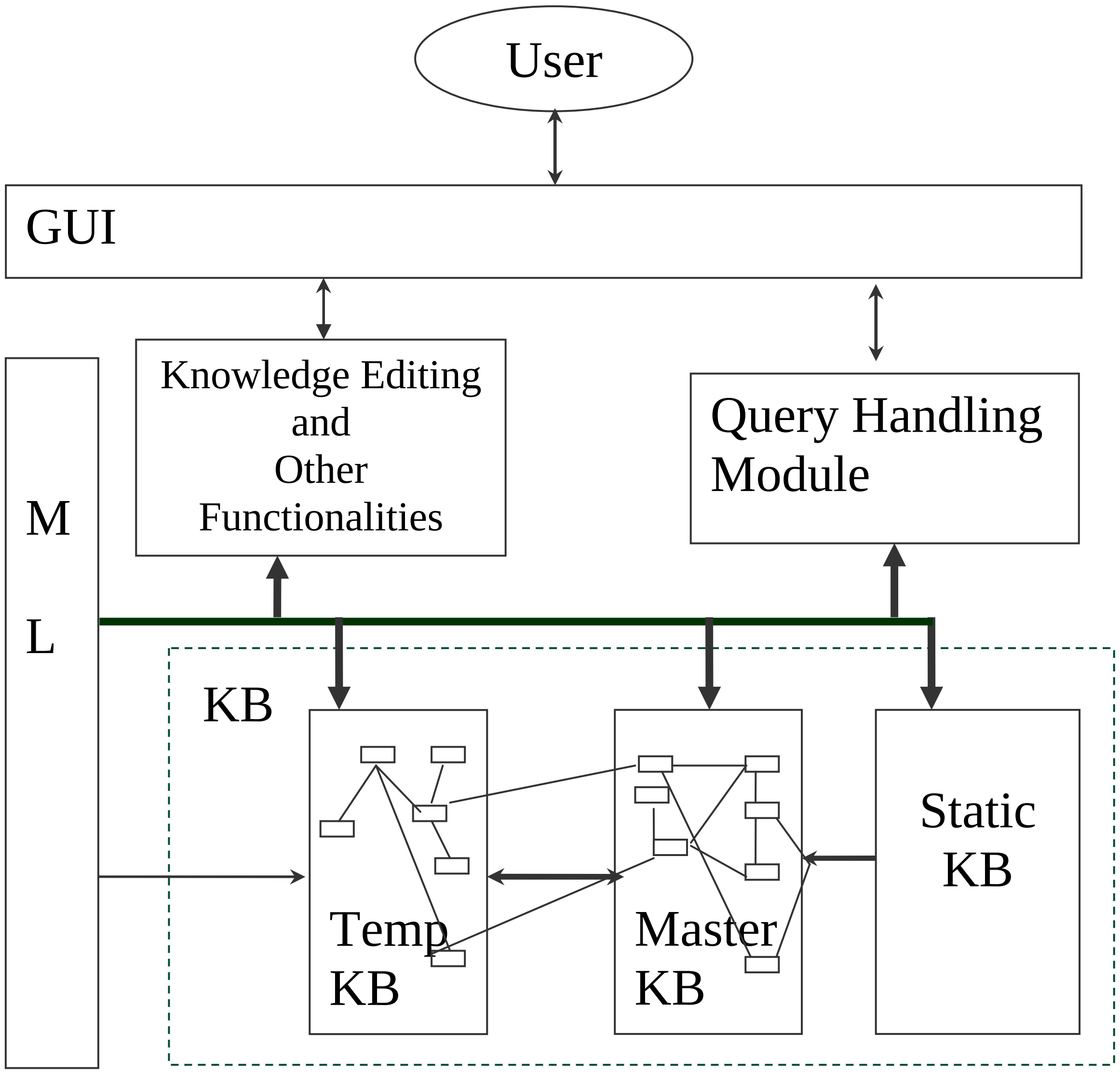


Figure 3.1 Iexpert System Model

There is a user on upper side of Graphical User Interface (GUI), which can be any one; there are different categories of users on the basis of their Islamic knowledge and understanding. GUI is user-friendly; user will easily find how to do what he wants to do.

Then inside the system there are four major modules; first Machine Learner (ML) on left side, second knowledge editing and other functionalities providing module, third a simple Query Handler (QM) which takes an English string as input and answers it. The fourth module of iexpert is represented by dotted line box, enclosing three other boxes. This is the Knowledge Base (KB) of the system. The major emphasis was on this module of the system. Current system is capable of performing simple string search through both the translations of Quran and Hadith and also from dictionary. Query Handler is made by modifying an old program named Eliza. It is subdivided into to parts one Question Builder (QB) and other Answer Builder (AB). The input English string is pattern matched against the pre-existing templates. If it matched with any of the templates the action part of that template is executed, which perform some action inside the system. It may be a learning action or question building work. Then the response is returned to the user. Each such template is called a “rule” and this is why the query handler is some times said rule-based query handler. The iexpert learner can make and add new such templates and rules to the system at run time.

Now we discuss each module in detail.

3.3 SYSTEM SECURITY

Iexpert has some security in terms of its use. First of all each user will have to enter his password and name into the system, if password is correct then the user would be able to use this system. User can change his password any time. System administrators could add and delete user accounts. They can also lock a user account.

Internally using the object oriented methodology each user is represented as an object of 'user' class. This class has different slots and each user has its own values in these slots. No two identical user-names are allowed.

The loginame of the user will be assigned the object of that user so when ever we need the object we will simply evaluate the loginame global variable *user* would be assigned the loginame of current user. Whenever we need all users we will call the function

```
User
     :loginame ;;; name of user
     :authority  ;;;authority level of user
     :password ;;; user entry password
     :locked-p  ;;; user account is  locked or not
     :user-files  ;;; files which user owns
     :no-of-login ;;;count for the number of logins
```

Figure 3.2 User class

Common-person, and Scholar are the two categories of users on the basis of their knowledge and understanding. Each authority level has its own access and limitations. The slot :authority tells about the authority level of the user. This is because an intelligent system should behave differently for different users. An humans do.

Scholar has the maximum authority. He can edit the all user accounts; he can edit the knowledge base of the system or do various verifications in the current knowledge base. Like other users he can use the Query handler and translations of Quran and Hadith stored in the system. Only the scholar can enter the learner and verify what the learner has learnt. Figure below shows the access of the different authority levels. The system orange color windows represent those which are only accessible by the scholars.

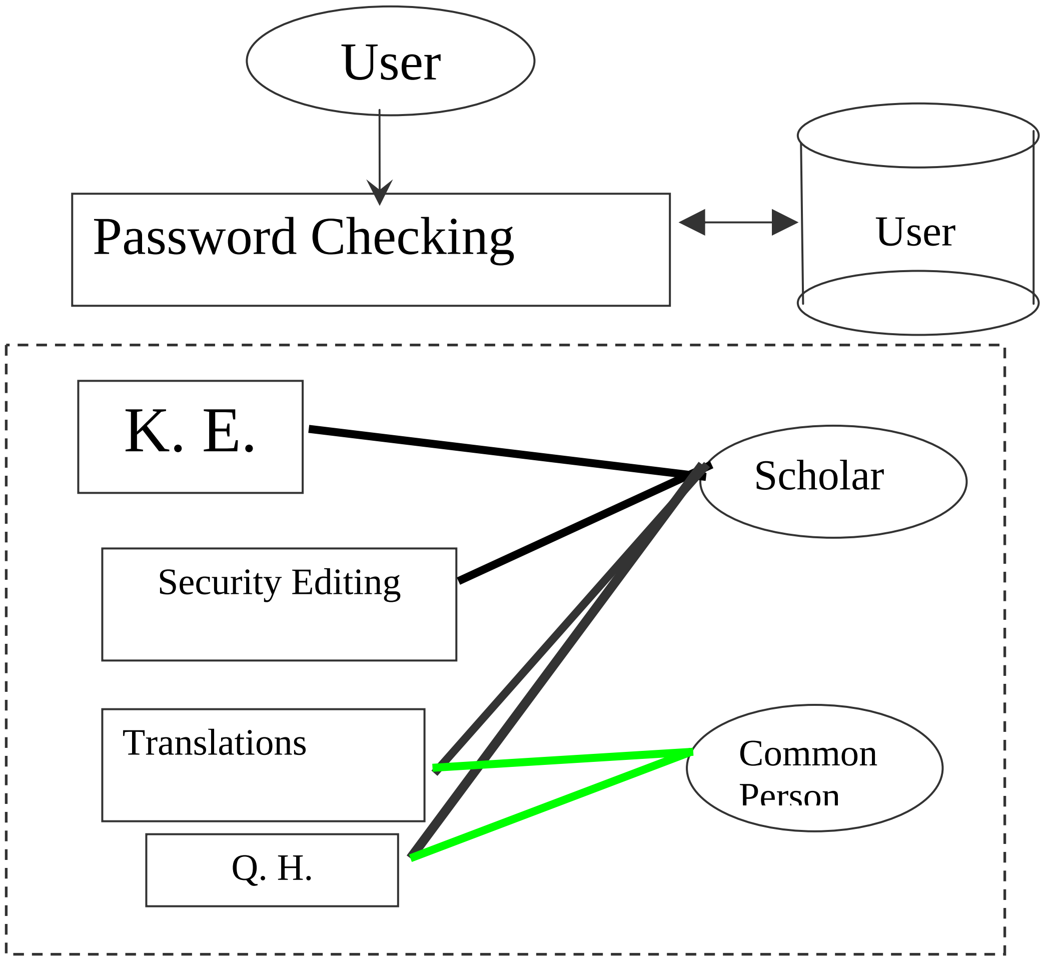


Figure 3.3 System Security Conceptual Model

Scholar can edit the user accounts of lower authority levels i.e. Common person can use query handler and translations of Quran and Hadith but he could not edit any user accounts. Only a scholar can add a new user or lock other user's account. Security management in the current software is not advance and complete.

3.4 KNOWLEDGE BASE

The knowledge base of iexpert is divided into two parts. One is static knowledge base or data base and other is dynamic knowledge base or simply the knowledge base. Static knowledge base is un-editable and so it should be, as it comprises of translations of Quran Hadith and an English dictionary. System itself use it for referencing and also learn form it.

3.4.1 Static Knowledge Base

Static knowledge base is that portion of knowledge base which is un-editable. Scholar can not edit it. He may accept or reject any part of it as a whole. It consist of two English translation of Quran, one by Pickthall and other translated by Yosuf Ali. It also contains the English translation of Al-Bukhari Shareef. Scholar can add new translation of Quran at runtime. And if any translation if doubtful scholar can reject it as a whole he can not edit its any part. Static knowledge base is like a simple database. Instead of relational approach of conventional databases object oriented methodology have been used. There are ten classes as shown in figure 3.4 given below. You can assume each class as a table. One record of such table is actually one object of that class. Current system contain little more than 20,000 of records or objects.

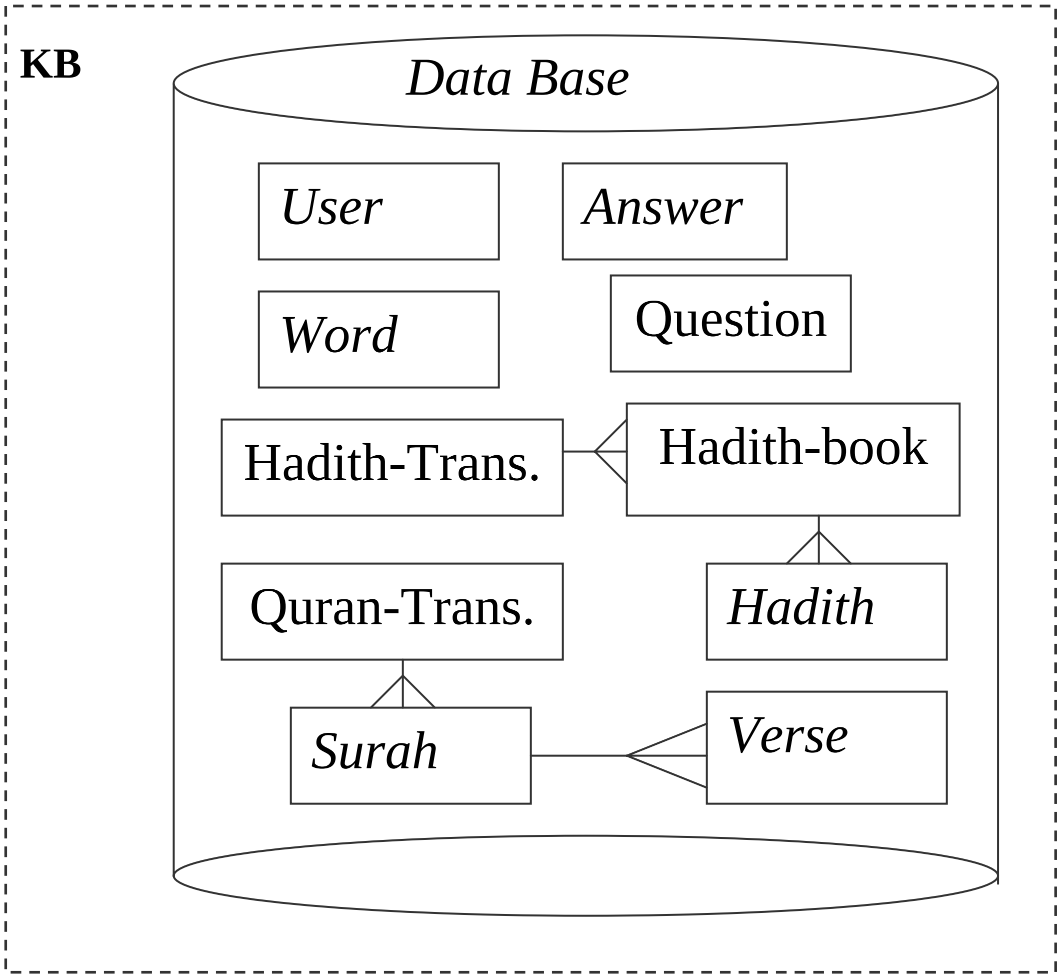


Figure 3.4 Static Knowledge Conceptual Model

User, Question, Answer and Word classes are independent. While other classes of Quran and Hadith translations have one to many relationships as shown in figure above.

User class is shown in figure 3.2 and is used for system security; other classes have been shown below.

```
QUESTION
      : id
      : asker
      : date              ;;; date of questioning>
      : statement      ;;; statement of the question
      : about             ;;; titles of the concepts related to the question
      : asked            ;;;  what is asked
      : required        ;;;    to may be display or may be print etc.
      : answer-id
      : difficulty-level   ;;; showing whether it is understand able to
                                   ;;; *child* *teen* *mature* *educated* or
                                   ;;; *master*
      : search-level
       : sound-symbols
```

Figure 3.5 Question class

When any user ask a query from the Query Handler, system first understand the query statement. It create an object of the Question class shown in figure 3.5. then the answer builder take that object of Question class and build an object of Answer class shown in figure 3.6. id slot of both the question and objects are generated by the system, and id of every object of any class would be unique. The last slot Sound-symbols in question class is for the list of symbols used in the query statement, which appears to be concrete and possibly be other concepts. Search-level slot indicates that how deeply the system have to search the knowledge base for giving the answer of this question.

Difficulty-level slot shows the complexity of understanding of the question. Answer-id slot contain a symbol as its value. That symbol is the id of an answer object, the object is assigned to its id. Similarly date, username and other useful information are stored in a question object.

Answer object somewhat contain similar values. Findings-from-system slot contain an embedded list as its value. And that list is contains every thing what system have searched to answer the user. Answer-to-user slot in current system contains the same value as findings-from-system slot, but in future it may contain that material which system thinks that should be shown to the user and hide the remaining material which it has found from its knowledge base.

```
ANSWER
  : id                   ;;;unique id of the answer
  : date
  : question-id
  : findings-from-system
  : answer-to-user
  : difficulty-level       ;;; showing whether it is understand able
                         to *child*  *teen* *mature* *educated* or
                         *master*
  : search-level
  : unidentified-symbols   Note that this slot will include the
                         sound-symbols    of Question object and also
                         find more symbols from its own findings-from-
                         system learner-cc would insert new-symbols
                         from the  Answer object not from the Question
                         object.
```

Figure 3.6 Answer class

Class word is actually the internal representation of Dictionary. Current system contains a dummy dictionary of about fifty words, in future an original dictionary can be embedded in the system. Others slot in Word class is for adding new slots at runtime. It will store the additional features and their values if any words have besides the ones given originally in figure 3.7.

```
WORD

 : name     ;;; this slot will contain the symbolic name of the word
 : sound    ;;; its value would be a string describing the sound pattern
 : is-a       ;;; noun, verb etc. in a list since a single word can have
                more values.
   : synonyms   ;;; list of related words (a word as string)
   : meanings    ;;; list of strings showing the meanings of the word
   : examples   ;;; list of example strings which are using the word
   :others       ;;; any other slots with their values are put in it
                ;;; in the form of embedded lists.
```

Figure 3.7 Word class

Then comes the classes of Quran and Hadith. Quran-translation is a class for complete translation of Quran which contains surah objects in it. Surah object further contain verse objects. Id slot is for storing the name of AYAT. For example, v002.001-pickthall is id of AYAT 1 of 2nd surah. The object of verse is created, its id is stored in object of SURAH and also in its own object of Verse class. The object of verse containing translation of that verse is itself assigned to its id. In v002.001-yusufali symbol (v is joined to ID of the AYAT to make it symbol) name of the translator is attached along with surah number and verse number.

```
VERSE

  : id                  ;;; unique id i.e AYAT number  v001.010-pickthall
                         ;;; v101.001-picktahll   are example of a verse id
   : contents           ;;; This slot is to make the design sound, aiming for
                        ;;; the original contents of a verse
  : translation
  : surah-number       ;;; number of surah containing it.(for effecient
                          searching)
  : verse-number      ;;; sequence number in surah (for effecincy)
  : directly-related-concepts
                          ;;; list of symbols representing concepts
                         ;;; THIS LIST MAY ALSO BE MODIFIED
                         ;;; BY iexpert LEARNER
```

Figure 3.8 Verse class representing an "AYAT"

The object of verse is assigned to symbol obtained from joining v to id .Only ID, TRANSLATION, SURAH-NUMBER,VERSE-NUMBER,DIRECTLY-RELATED-CONCEPTS is filled by verse ids produced by a method named verse which takes a string as input. The figure

```
SURAH
  : id      ;;; s1-pickthall   is an example of a surah id.
  : titles  ;;; list of symbols representing the titles of that surah
            ;;; only the first symbol will be assigned the object of surah
  : place  ;;; place of revelation Makkah or Madinah (Makki or
                Madni)
   : no-of-verses      ;;; for effeciency purpose
   : verses             ;;; list of all the verses of the surah
                       ;;; method call of VERSE would be written inside
                       ;;; so that objects are created at runtime
```

Figure 3.9 Surah class

Each translation of Quran is object of Quran-translation class shown in figure 3.10. It contains publishing-dates and also the dates during which the translation was written by the translator. Currently these information are not important but the place is their for future possible use.

```
QURAN-TRANSLATION
   : translator          ;;; symbol name of transltor to whom the object of this
                        ;;; class will be assigned SERVE AS ID.
   : surahs             ;;; list of all Symbols (Name of surahs) setqed with
                             SURAH Objects
   : language            ;;; To have translations in different languages
   : publishing-dates     ;;; list of publishing dates
   : translation-dates    ;;; list containing start and end date of translation
```

Figure 3.10 Quran-translation class

Level-of-difficulty slot in Hadith class indicates how much it is difficult to understand this Hadith. It contains a symbolic value which tells who can understand this Hadith.

```
HADITH
   :id                    ;;; unique id i.e HADITH number
                          ;;; h1.001-bukhari
                          ;;; h1.211-bukhari are example of a verse id
   : narrated-by          ;;; this slot may contain string value or
                          ;;; a symbol value which will be the name
   : translation
   : meanings             ;;; This slot is a list of strings representing additional
                          ;;; comments of schalors (old ones)
   : hadith-number        ;;; sequence number in hadith-book (for effecincy)
   : directly-related-concepts
                          ;;; list of symbols representing concepts
                          ;;; THIS LIST MAY ALSO BE MODIFIED
                          ;;; BY iexpert LEARNER
   : level-of-difficulty
```

Figure 3.11 Hadith-class

Hadith-book class contains one set of Hadith. In Al-Bukhari there are nine (9) Hadith-books. Each of which contains hundreds of Hadiths. Figure 3.12 shows the class of Hadith-book.

```
HADITH-BOOK
   : id                ;;; hb1-bukhari   is an example of a HADITH-BOOK id.
   : no-of-hadith      ;;; for effeciency purpose
   : hadiths           ;;; list of all the hadiths of a book (JILD/edition)
                       ;;; method call of HADITH would be written inside
                       ;;; so that objects are created at runtime
```

Figure 3.12 Hadith-book class

Hadith-translation class represent one complete translation of Hadith. It contains the name of translator and the language of translation. The slot books is a list of symbols each of which is assigned a set of Hadiths.

```
HADITH-TRANSLATION
  : translator        ;;; symbol name of transltor to whom the object of this
                      ;;; class will be assigned SERVE AS ID. for example
                      ;;; Bukhari, Muslim etc.
  : books    ;;; list of all Symbols (Name of surahs) setqed with
             ;;; HADITH-BOOK Objects like (hb1-bukhari hb2-bukhari..)

  : language  ;;; To have translations in different languages
```

Figure 3.13 Hadith-translation class

3.4.2 Dynamic Knowledge Base

This is the actual knowledge base. In the beginning of this research work I thought that it is going to be represented in the form of semantic nets as shown in figure 3.14 shown bellow. Semantic nets have been discussed thoroughly in previous chapter.

In semantic nets each node is a symbol, attached with some directed arcs showing its associations with other symbols. These symbols are normally the names of objects or concepts. But in actual implementation I have to mix frames and semantic nets to deal with complexity and to get better organization of knowledge in the system.

In iexpert knowledge base each node is itself an object of a main class CONCEPT. Hence the dynamic knowledge base of iexpert, which it use for query handling and learning, consists of objects of Concept class linked with directed arcs. This is logical representation, actually these links are the slots and their labels are slot values.

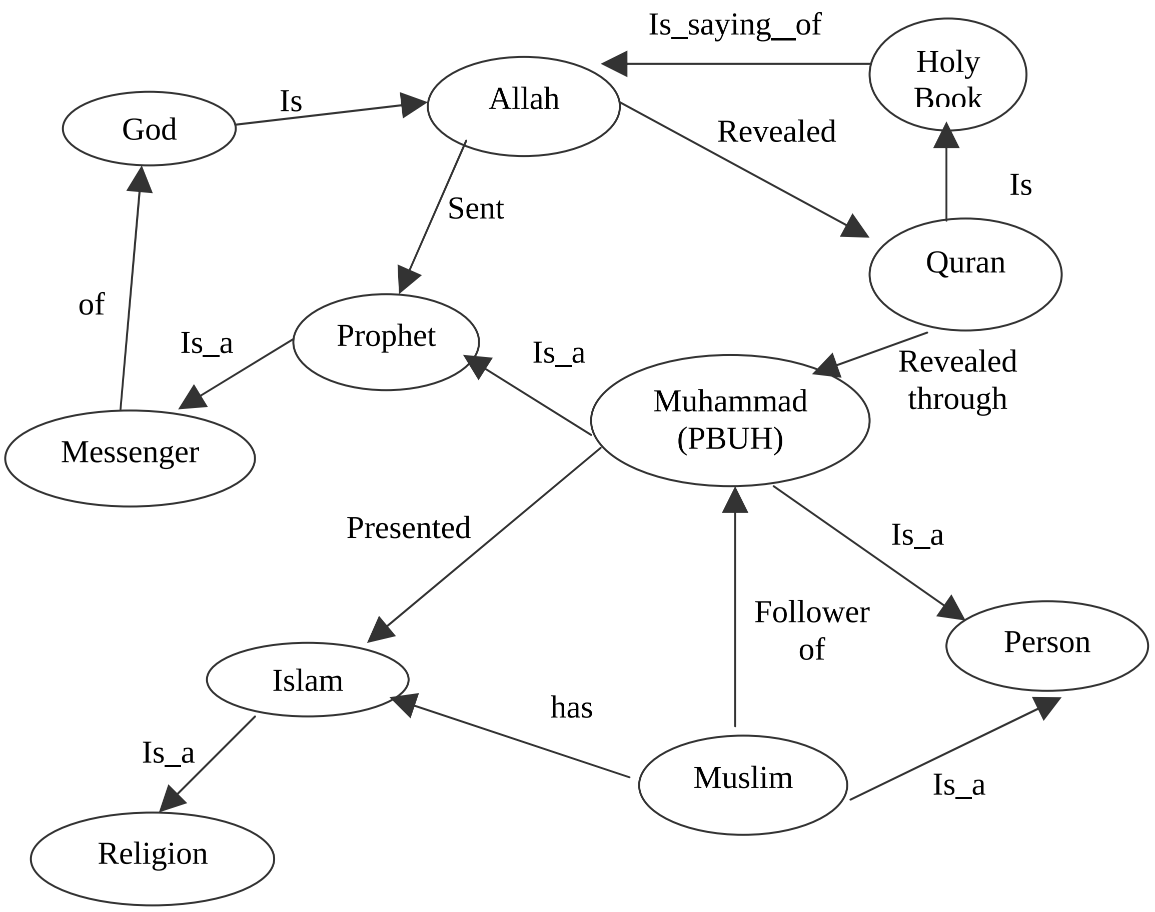


Figure 3.14 Dynamic Knowledge Semantic Nets

Each concept has its own associations with other concepts. This representation is little simpler than the actual frames in which there are different classes with their natural relationships among each other. Here only one class is used but the objects are different in terms of their values.

```
CONCEPT

   : name        ;;; BY USER . This slot will contain the symbolic name of the concept
                 ;;; which is setqed with the same object

   : titles           ;;; List of symbols which represents this concept
                      ;;; BY BOTH USER AND LEARNER.

   : meanings           ;;; List of strings telling the meanings of titles
                        ;;; BY BOTH USER AND LEARNER.

   : directly-related-concepts    ;;;      this slot is for referencing the concept
                    ;;; except first title all will have it as describe above. BY BOTH.

   : related-concepts    ;;; list of concepts related to the current concept. BY BOTH

   : parent-concepts     ;;; list of concepts from which the current concept is inherited.
                       ;;; NOTE that this inheritance is between objects of the same
                       ;;; concept class. and also that multiple inheritance is allowed.

   : child-concepts    ;;; reverse of the above slot. BY BOTH

   : difficulty-level     ;;; BY BOTH USER AND LEARNER. Representing the level
                       ;;; of difficulty in understanding there are five levels  *child*1,
                       ;;; *teen* 2, *mature* 3, *educated* 4, *master* 5. Slot value
                       ;;; would be number

  : concept-generator  ;;;BY SYSTEM ONLY. This slot will tell who have  identified
                       ;;; this concept it may be *iexpert-learner* or any user.

   : concept-editors     ;;; list of loginames of *scholar* s who have edited this concept.
                         ;;; BY SYSTEM ONLY

   : concept-validators  ;;; list of loginames of *scholar* s who have validated it.
                          ;;; BY SYSTEM ONLY

   : creation-date          ;;; date when it was first time identified. BY SYSTEM ONLY

   : modification-dates  ;;; BY SYSTEM ONLY. List of dates on which it was edited
                  ;;; length of this list should be equal to length of list of concept-editors

   : status      ;;; shows the current position or status of the object whether, It is  *new*
                 ;;; *edited*  *dirty*  *validated*. BY SYSTEM ONLY

   : comments   ;;; comments of different scholars. BY USER ONLY
   : related-verses ;;; list of objects of verses directly related to it. BY BOTH
   : related-hadith ;;; list of objects of hadith directly related to this concept. BY BOTH

   : others     ;;; BY BOTH. This slot is for adding new slots and their values at
                ;;; runtime format of its value is ((<slot-name> (<values>))
                ;;; (<slot-name2> (<values>)).... )
```

Figure 3.15 Concept class

In the above figure the slots with comments BY BOTH means that this slot is editable by user and the learner or in other words user and the system itself can change the value of this slot. BY USER means the value of this slot is provided by the user through a user interface. BY LEARNER means that this slot can not be edited by the user, system itself generate its value. Others slot in concept class is of great importance. It contains the links of this concept with other concepts. It is an embedded list. New slots can be added to it an runtime. Hence it is useful for learning purpose.

This is the super/parent class all the classes of iexpert system. All the concepts of the iexpert master base would be the objects of this class. The object of the concept class would be assigned to the symbol being used as its first title. Other titles will also be assigned the objects of concept class but with nil values on all slots except the slot related-concepts, this slot will contain the other name of this object of concept, which is assigned the original copy of the object. So all the titles will be assigned the objects of concept class only for referencing.

Dynamic knowledge base is further divided into to categories i.e. Temp KB and Master KB. Temp knowledge base contains those concepts, which have not been currently verified from the human scholar. Concepts in tempkb may still have links with those present in masterkb. Master KB contains those concepts, which have been verified from human scholar. When a new concept is learnt by the learner it is stored in the tempkb and after verification it will move to the masterkb. When again the learner or scholar edits it it will move back to tempkb. This process continues, more and more knowledge will be added to that concept.

3.5 QUERY HANDLER

Query Handler of iexpert is rule-based. As described in chapter 2 a rule is simply if--- then --- construct. If the condition matched or becomes true then then-part is executed. Query Builder of this system has two major part Question Builder (QB) and Answer Builder (AB). There are two small parts, as shown in figure 3.16, one verifier, which verifies the query statement and other displayer, which displays the answer in suitable format. Question builder part takes a simple English string as an input and returns an object of question class, shown above, as an output. This Question is passed to the answer builder which try to fulfill its requirements and build an answer object. That answer is passed to the displayer code, which displays it to the user in readable format.

Query handling is part of Natural Language Processing. An advance Query handler should have a parser in it which parses the input string according to the syntax and semantics of the language and stores the string in some sort of internal representation. This issue is completely discussed in previous chapter.

This Query Handler is made by modifying an old program named Eliza. In chapter 2 there is complete introduction of it.

3.5.1 Modifications in Eliza

Following modifications are suggested and are possible to make, while some are currently implemented.

1. Action command is added to Eliza rule format which would be its second element and will be executed (besides rule response) when a rule matched with the input. You could give one function call not embedded because flatten is used before evaluation.
2. Resolving the reference by using *reference-table* which contain the current values of IT, HE, SHE, THEY, THIS , THAT etc.
3. Using different rule sets while discussion proceed, by using qh-get-rule-set function.
4. Resolution of suffixes like -ly, s, 's, -ing, and prefixes like anti-, un- , etc (not currently implemented.)
5. System should be able to add/write new rules on its own at runtime by observing the language/inputs of users.(not currently implemented.)
6. Little bit spell tolerance.
7. It may collect all the rules matching the verify from the user one by one and execute that rule command which is verified.

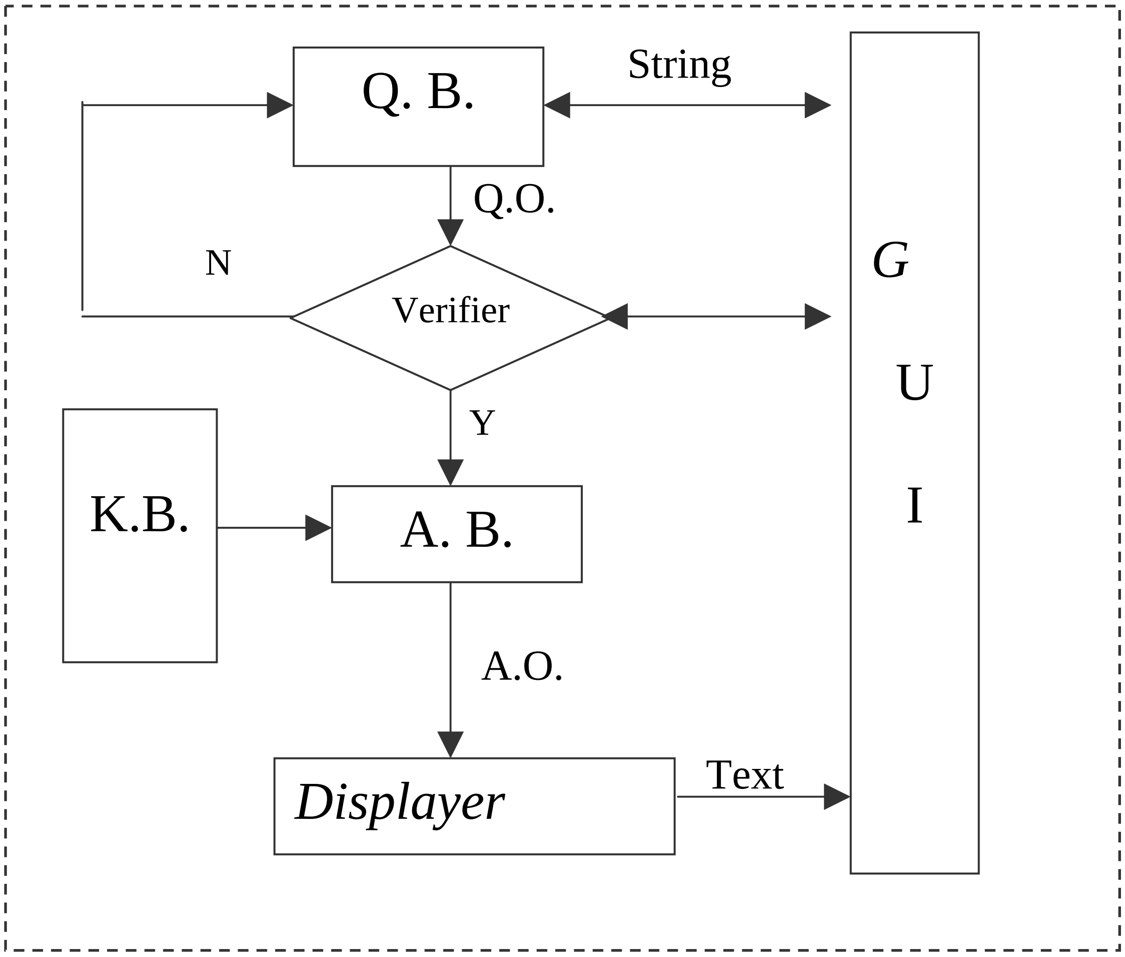


Figure 3.16 Query Handler Conceptual Model

Some part of Learner code is embedded in Query handler. During query handling system also learns form the user. This is implicit learning, user do not know that system is getting something from him.

3.5.2 Question Builder

First part of Query handler is Question builder. It is responsible for understanding the question. As earlier we have defined understanding as mapping of the input to the internal representation. Internal representation of a query is Question class and external representation is a simple English string. This part of code converts the external representation to the internal representation or in other words converts the input English sentence to an object of Question class. It has different rule sets. A rule can be assumed as a template for a query.

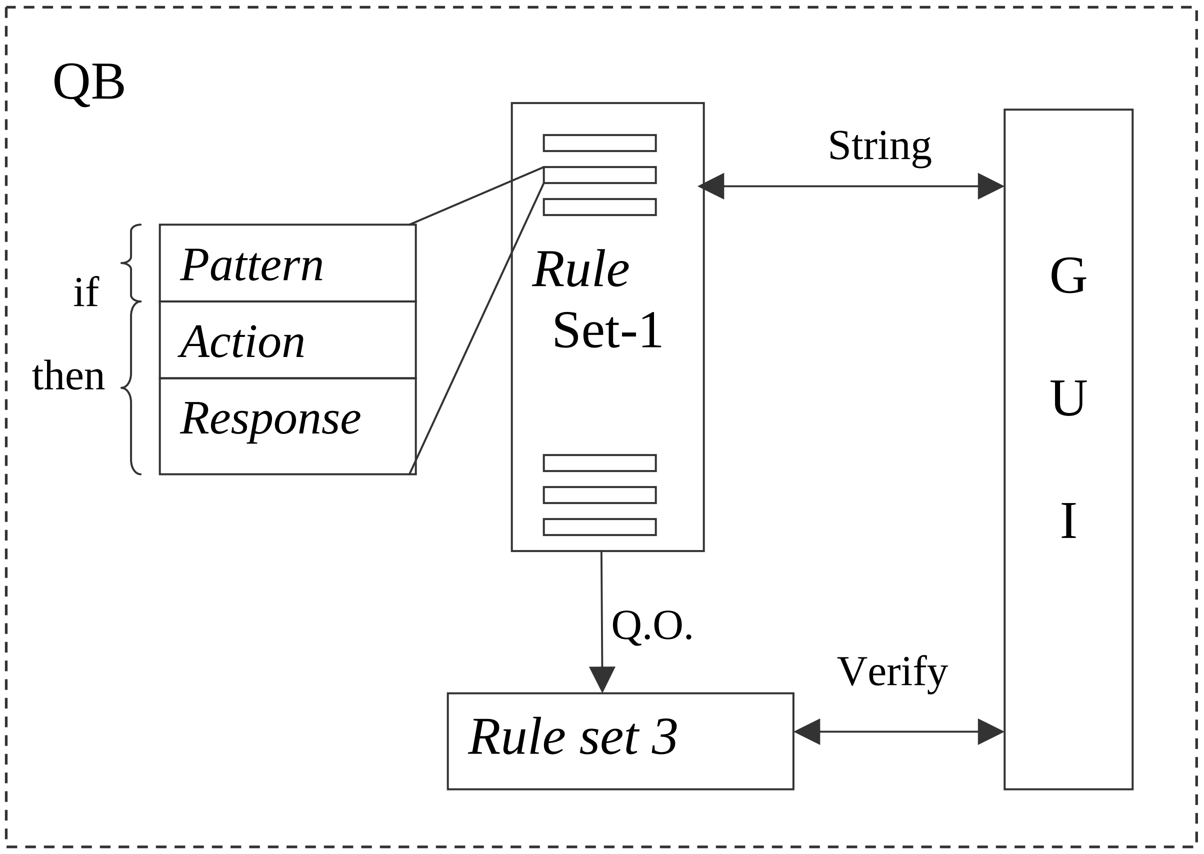


Figure 3.17 Question Builder Conceptual Model

You can see in figure above a rule has three parts. First part is a pattern that pattern is matched with the input string and as a result a binding table is produced. Binding table is a structure in which the variables of the rule pattern are binded with their corresponding values in the input string. If the binding table is produced then the action command of that rule is executed. An action-command is simply one expression of lisp code which may be embedded one and contain many expressions of lisp code inside it. Then the response of the matched rule is returned to the user but the variables in the response are replaced with their corresponding values of the binding table. Most of the rules of Question Builder contain a lisp code in their action command, which creates the Question object. The question builder returns that object produced by the execution of action-command as output. You can see from figure 3.5 what information that question object carries with it. Most important are the about and asked slots, first one tells about which concept that question is related and the second one tells what is asked about that concept.

3.5.3 Answer Builder

That question is verified before answering. This is because it is better to confirm the understanding of question instead of returning the wrong answer. After the verification the answer builder part of query handler is activated. The figure 3.18 shows the complete process of answer building. First of all that concept is searched from the master kb (verified concepts) with which the question is related. If not found (represented by F for Failure in figure 3.18) then edited concepts are searched. If the concept is found then the answer is build on the basis of what the user have asked.

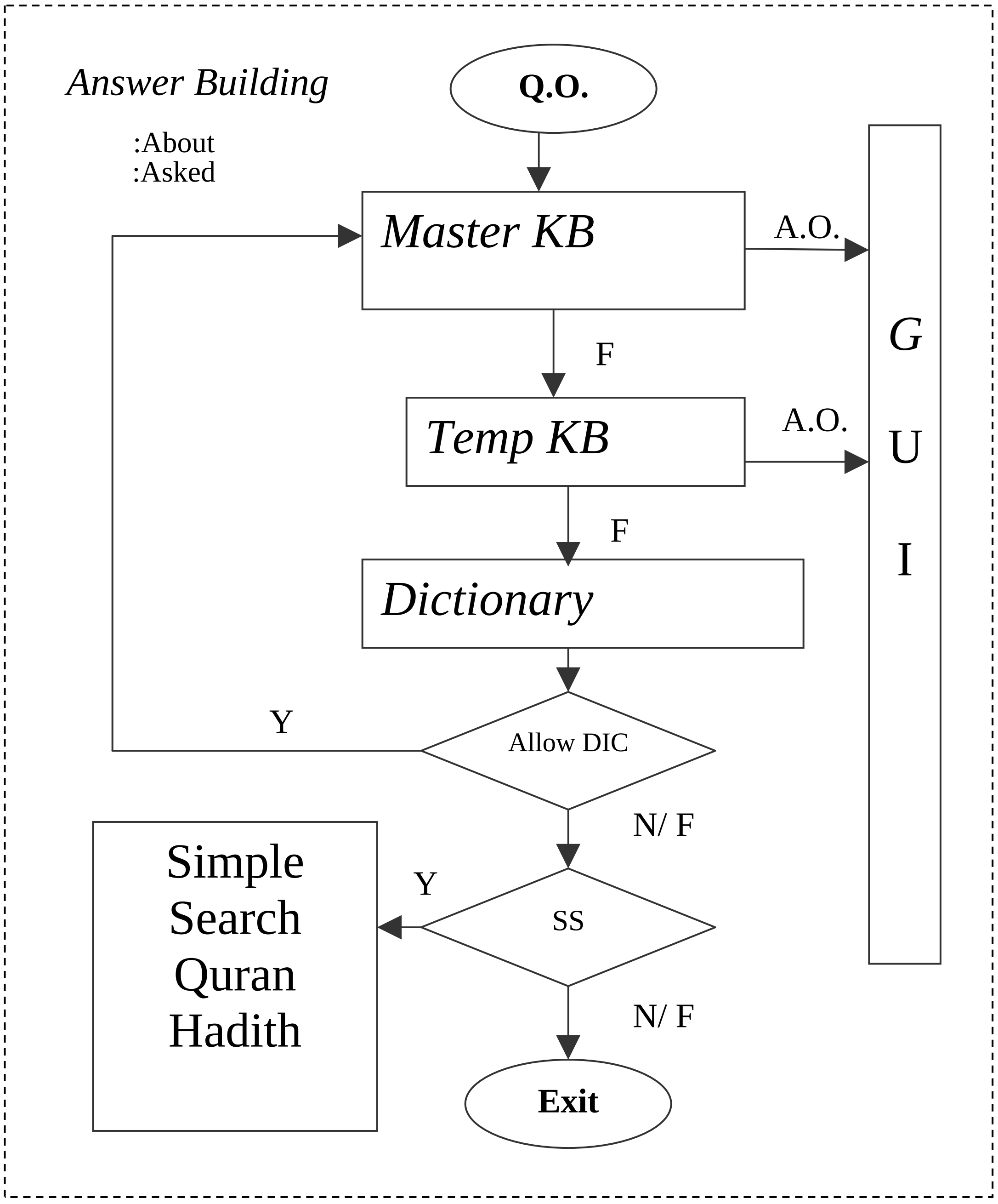


Figure 3.18 Answer Builder Conceptual Model

Then that answer is displayed to the user. If the about-concept is not found in knowledge base then its synonyms are searched from the dictionary and then it is seen that any of its synonym has a concept in knowledge base, if yes then answer is built form that concept. If the concept we are looking for not found in dictionary then on user will the simple search is performed from the translations of Quran and Hadith. And those verses are returned which contains the name of that concept. This is the complete depict of answer building part of query handler. You can see the lisp code in appendices.

3.6 IEXPERT LEARNER

Without learning there is no intelligence, although the learner of this system is not very advanced but it contains something which could be said as learning. Before discussing the learner we discuss two important issue, first that when a learner should learn and second the verification from the human Islamic scholar.

3.6.1 When to learn

when the humans learn, at similar situations the system should learn. Humans learn when they see any unfamiliar thing they learn about it. Same thing this system do when it found any unfamiliar symbol it record it to its Working Memory (WM), then ask from the human scholar about that symbol.

The second time when the learn becomes activated is that when the query handler fails to translate the input string into the question object or in other words it fails to understand the question of the user. Learner stores that statement into its working memory, and when Islamic scholar logins it asked him about that un-answered statement. It make a rule from it showing what is asked in this statement about which concept.

3.6.2 Verification from Human Scholar

No system can ever be so advanced that it don’t need any guidance from the human scholar. In learning there is one thing necessary that is the teacher, who makes corrections and verify the correct things. Although the humans verify their concepts from the experimental evidence but the computer systems cannot do any thing they can only ask to do something, as they have limitations on input and output.

Religion is very sensitive matter. No programmer can take the risk that his system give the opinions in religious matters by its own. As the responsibility comes straight to the programmer, so he shifts it back the religious scholars. Hence verifications from the human scholars are necessary especially in this system.

You can see the conceptual model of learner in figure 3.19. Learner of this system learns only three things by itself. Although the knowledge editing can also said to learning, which the scholar do for building of knowledge base with the help of concepts. But here we consider those things which the system itself point out and these are three. First learner collect implicitly those symbols which are new to it during the conversation with any user. Then it presents those symbols to the human scholar when he logins, and ask if any of these are concepts. If scholar approve any symbol as a concept then the learner adds it to the knowledge base of the system as a new concept whose concept-generator is the learner itself.

The second thing is identification of new slots of existing concepts. The learner stores them and then verify from the human Islamic scholar. If verified, learner adds those slots to the concepts in its knowledge base. The third thing is not simple, it is the programming of new rules by the human scholar at run time. Of course the Islamic scholar is not suppose to be a programmer, the job of programming according to the syntax of lisp is done by the learner, the human scholar provides the conceptual help.

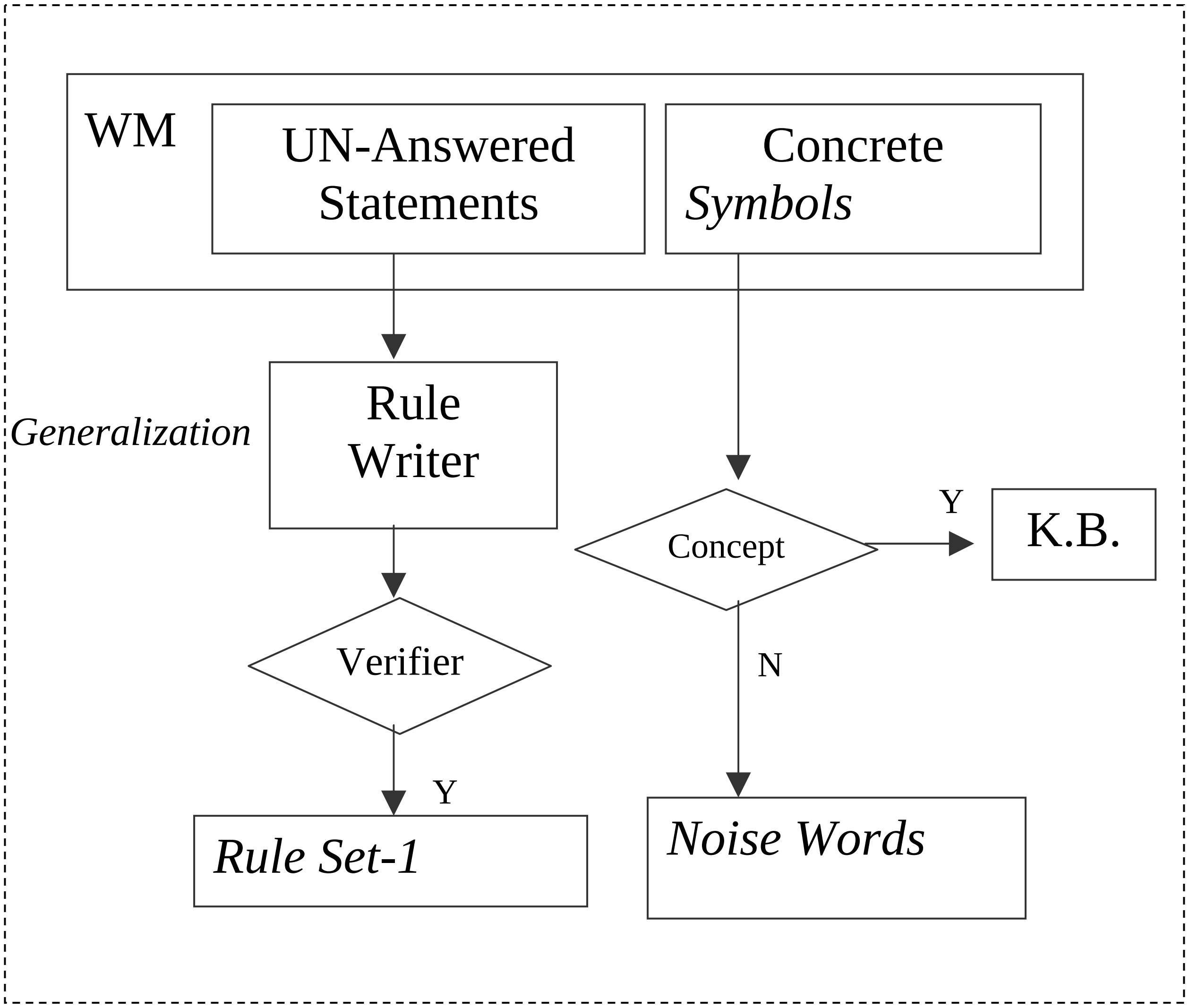


Figure 3.19 Learner Conceptual Model

Generalization is important concept in learning. It is like if we see something about any thing then we try to map it on other things and in this process we do findings. Induction is one type of logical inferencing technique which is similar to generalization.

3.7 SUMMARY

This system is not an actual expert system, rather you can say it an intelligent, efficient, referencing Islamic system. This system can help the Islamic scholar to organize their knowledge in systematic form. They can take help from it to avoid human errors. Humans are very much prone to errors because of their complex and sensitive nature. They may give different response on same thing at different times. Systems like this, help us to be consistent.

System is currently having two complete Quran Translations one translated by Pickthall and other translated by Yosuf Ali. System also contains one English translation of Hadith book AL-BUKHARI. There is a dummy English dictionary of about fifty words, for testing that the system can reference the dictionary for unrecognized words by itself automatically. The knowledge base of iexpert is divided into two parts. One is static knowledge base or database and other is dynamic knowledge base or simply the knowledge base. Static knowledge base is un-editable and so it should be, as it comprises of translations of Quran Hadith and an English dictionary. System itself uses it for referencing and also learns form it. In iexpert knowledge base each node is itself an object of a main class CONCEPT. Hence the dynamic knowledge base of iexpert, which it use for query handling and learning, consists of objects of Concept class linked with directed arcs. This is logical representation, actually these links are the slots and their labels are slot values. Each concept is an object of this class. These objects linked with each other form the dynamic knowledge base of this system. Dynamic knowledge base is further divided into to categories i.e. Temp KB and Master KB. Temp knowledge base contains those concepts, which have not been currently verified from the human scholar. Concepts in tempkb may still have links with those present in masterkb. Master KB contains those concepts, which have been verified from human scholar.

Query Builder of this system has two major part Question Builder (QB) and Answer Builder (AB). There are two small parts one verifier, which verifies the query statement and other displayer, which displays the answer in suitable format. Question builder part takes a simple English string as an input and returns an object of question class as an output. This Question is passed to the answer builder which try to fulfill its requirements and build an answer object. That answer is passed to the displayer code, which displays it to the user in readable format.

Learner of this system learns only three things by itself. First learner collects implicitly those symbols, which are new to it during the conversation with any user. Then it presents those symbols to the human scholar when he logins, and ask if any of these are concepts. If scholar approve any symbol as a concept then the learner adds it to the knowledge base of the system as a new concept whose concept-generator is the learner itself.

Chapter 4

RESULTS AND DISCUSSIONS

4.1 INTRODUCTION

In this chapter we shall discuss what we have deduced from all our research work. We will give the final words about the natural way of knowledge representation as knowledge is stored in human mind. This chapter will elaborate the meanings of understanding. It will talk about machine learning with respect to the human mind. Further it tells about, what is left according to the scope of our work and what are the future possibilities.

All this chapter contains are new my own theories, which I have built at the end of my work. These may be right or completely wrong or partially right and partially wrong.

4.2 NATURAL WAY OF KNOWLEDGE REPRESENTATION

Humans store their knowledge in the form of concepts, as I have done in my work but the concept class of human mind have not have the same design as the concept class of our system have. Each concept have a name for its identification. The whole object of human mind concept is attached to the symbolic pattern and the sound pattern of that concept. For example if I say ‘mango’ or if I write ‘mango’ (as I have done twice), that sound or that symbol will immediately brought the general concept of mango in your mind it may contain some graphics. Then your mind will make further assumptions. You will wonder why I have given this example. Don't worry I am trying to elaborate the relation of a concept to its name. Name could be a symbol or a sound pattern or a graphic pattern. When that name is reminded if you know that thing the concept of that thing immediately comes to your mind, and if you don't know that thing you will be ready to learn about that thing or add new concept in your mind. This step is common to all human beings. Then the further thinking as I have talked above varies from person to person. Remember that first step is the coming of that concept into mind (active part or working memory of mind). Next thinking may be different. And each person will have its own concept of that thing. Notice other important thing that in order to get the concept of 'mango' our mind has not performed any kind of search. The concept was stored with its name. And this is the natural way of knowledge representation and knowledge keeping. If you have not liked that example just change the symbol 'mango' with 'apple' and read the passage again, the picture will be completely different. You can have picture of your own by simply changing the symbol and reading the above passage again, unless you have got the point. What I am saying is that naturally the knowledge is represented in the form of objects of concept class, assigned to their corresponding names. That assignment or pointing by the names is the natural way of storage.

Now take an example of search, if I say “imagine a small red and round thing”. What your mind has reminded, is it a red ball or an apple or something else? Actually your mind has done a heuristic search. It has completely rejected big things and searched for a small thing which have red color. Searching mechanism of our mind is very advanced. When we are “thinking” we are actually “thinking”. If you think more about that small red round thing your mind will search your knowledge base more and more and it may comes up with a new thing instead of ball or apple.

Each concept has complete information of what we know about that thing. If we see or hear the name of something with which no concept is attached our learning mechanism will be activated and we will be ready to build a new concept, even if we can’t get any information of that thing our mind will keep that concept empty. Whenever our mind found information about that thing it will immediately attach it to its name.

If we see a thing who’s name we don’t know, our mind will immediately try to name that thing. And this is learning. This first step is common others depends upon the nature of viewer person.

4.3 UNDERSTANDING

What are the meanings of understanding? When you say that you have understood my point exactly what changes had occurred in your mind or knowledge base? One simple definition of understanding is that it is the transformation of input to the internal representation of knowledge base. If the input is correctly changed into the internal representation then there is correct or right understanding and if the input is not correctly transformed into the internal representation then it is misunderstanding. Like humans an intelligent system should also misunderstand.

In my view understanding is a cognitive process. New thing is added in such a way that it can be used in future processing. Assume that our knowledge base in mind is in the form of semantic nets. When we learn new thing a new concept is built in our mind and this concept will have links with other concepts. Initially the links are very few but with the passage of time when the person have more and more understanding of that new thing, actually he is establishing more and more links of this concept with other concepts and more slots with values are added to this new concept. We call these slots and values, information of the thing. Thus understanding of a thing is establishing more links and adding more slots and values to the concept (internal representation of that thing).

4.4 IS MACHINE LEARNING POSSIBLE

Before discussing the machine learning first we consider that if it is possible or not. Humans learn things, technically we can say that they edit their knowledge base at runtime. With the passage of time they get more and more knowledge due to the dynamic programming in the mind. Remember that the learning mechanism is somehow fixed, they hardly change their learning methodology or learning algorithms. Animals cannot learn because they can not edit their database, they performed the specific tasks which were programmed in them. It means a thing which is programmable at runtime can learn. Every computer program can read and write files, if some effort is made they may write such files which are also executable code. Hence computer programs can have this human's characteristic of learning, which make humans the intelligent creature of this planet. But the difference in physical nature of humans and machines make the learning doubtable.

Other important point to note is that we don't know anything. We are always searching for the reasons. Our whole knowledge is actually a huge data, organized in such way that mind could use it efficiently. We do nothing but rote learning. There is only one form of learning that is rote learning and computers can do this better than humans but the difference comes in internal representation of knowledge.

4.5 WHAT IS LEFT

Although in the start of this work my scope was not having the clear boundaries. The scope of this work was limited to the base of an expert system, but the base itself is not clear. Which things are the part of base and which are above of it? Somehow there are some things that might have been included in this system, discussed below.

4.5.1 Original dictionary

Current system has a dummy dictionary of fifty (50) words. System uses this dictionary for meanings of those words, which are undefined to the system. It should have original English to English dictionary with thousands of words. Current dictionary prototype is for checking the functionality of the system that if it could use the dictionary. It is working fine with the prototype. There are no problems of efficiency in searching the dictionary. Neither this problem will appear in case of big dictionary, because the way of knowledge keeping is natural as discussed in the above section. Dictionary is represented by the class word. Each object of word class contains all the meanings related information of a word or in other words it is equivalent to one record of the dictionary.

4.5.2 More Translations of Quran and Hadith

Current system has two Quran translations and one Hadith translation. Although the Quran translations are enough but their might have been another translation of Hadith. A new translation of Quran can be added to the system at runtime. Although the translation of Quran can be added by the scholar only but there are no extra verifications. System should get verification from more than one Islamic scholars.

4.5.3 Enhanced Knowledge Base

Like a child current system has a small knowledge base. There are initially about twenty concepts in the knowledge base. Since the focus of this work was not the Islamic knowledge rather it was on the implementation of AI concepts. So this knowledge base is only for testing the working of the system. More knowledge should have been added to the system so that some defects may appear, which are hidden due to the small size of knowledge base.

4.5.4 Recitation of Quran And Arabic Text

Current system don't have recitation of Quran facility neither user can see the actual Arabic text of Quran. These things would have been important from commercial point of view. As I was handling English as natural language. Handling the Arabic language would have been more difficult and beyond my skills. Although the future researchers should also work on other languages which have more difficult syntax or grammar.

4.6 CONCLUSION

Research work is incomplete (since the completion means an Islamic Scholar), but may be complete as my scope/ aim was to provide base of an expert system and it has taken one intense year of work. Knowledge Representation is complete, Basic Ideas about Query Handling and Machine Learning has been developed, and can be further enhanced in future research work. It is like a launching pad for the future research.

Current system is like an intelligent efficient referencing Islamic system for the use of human scholar. It is also useful to ordinary persons. They can use query handler of the system. They can read the translations of Quran and Hadith and can also perform simple search through these. The system could not recite the Quran and does not contain the Arabic text of Quran and Hadith.

I think this system is more useful for the future researchers of AI. I don’t know how much this effort have been successful and to what extant this work is new. If there is something that have been discovered by this work?

4.7 FUTURE POSSIBILITIES

As earlier said this work is a base for future research work. Several separate thesis could be done by extending this work. Some of them are given bellow. The future possibilities could be many times more if we handle sounds. They further increase if the Arabic and other languages are added to the scope of the study.

4.7.1 Web Base Front End

System could have a web-based front end so that remote users could use it. System should also learn from them but they may not have the authority of editing the knowledge base. Only human Islamic scholar would be able to edit the knowledge base by adding and deleting the knowledge and making verifications. This would be a better way to increase the system knowledge base, since the more the users of it, more it will learn from them.

4.7.2 Advance Query Handler

Query handler of the existing software is not very much advanced. Since it is not easy to handle or understand a natural language. Now it could take only one statement query and return its answer. Although the answer may be of many lines. We can have an impressive query handler with a parser built in it and which could handle a multi-sentence query as the humans do. It may handle the case studies as people go to the Islamic scholars with some particular cases, scholar tells them the best decision with reference of Quran and Hadith, system should also be able to do so.

4.7.3 Advance Machine Learner

Learning mechanism of this system is not advanced. We can build a separate learner and then plug-in this new learner to this system. As described in review of literature, learning is of many types. So we can further divide this project into small pieces. Each small piece could be a practical implementation of one type of learning. Then these parts may be joined to build a joint learner. Then like query handler, learner might have been plugged-in the main system.

4.7.4 Graphical Representation of Concepts

There might be another project related to the graphics. We may have a graphical front end for the knowledge base of this system. It may be in the form of semantic nets with each node representing a concept. It would be convenient to see the current knowledge of the system and how its concepts are linked with each other. Humans understand graphics more quickly as compared to the text. So it will help the user to comprehend the semantic network of knowledge of the system.

Bibliography

[Bain 86]	Bain, W. M. *"JUDGE: A Case-Based Reasoning System."* Kluwer Academic Publishers. 1986
[Bobrow 75]	Bobrow, D. G. *"Dimensions of Representation."* Morgan Publishers. 1975.
[Charniak 99]	Charniak, E. *"Introduction to Artificial Intelligence."* Addison Wesley Longman Inc. 1999.
[Feigenbaum 83]	Feigenbaum, E. A. *"The Fifth Generation Artificial Intelligence."* Addison Wesley Longman Inc. 1983.
[Hashim 00]	Hashim, M. *"Proof of Ammunition Acception and Rejection Criteria."* MS-CS thesis, National University of Science and Technology Rawalpindi. 2000.
[Hayes 74]	Hayes, P. J. *"Some Problems and Non-problems in representation Theory."* University of Sussex. 1974.

[Levesque 84] Levesque, H. "*Foundations of Functional Approach to Knowledge Representation.*" Cambridge University Press. 1984.

[McDermott 81] McDermott, J. "*The Formative Years*". AI Magazine. 1981.

[Norvig 92] Peter Norvig, "*Paradigms of Artificial Intelligence Programming: case studies in common lisp,*" Morgan Kaufmann Publishers, Inc. 1992.

[Rich 82] Rich, E. "*Artificial Intelligence.*" Prentice Hall Inc. 1982.

[Sawar 92] M.J.SAWAR, "*A Learning Apprentice Medical Decision Support System,*" Ph.D. thesis, The University of Leeds, 1992.

[Simon 83] Simon, H. A. "*Why Should Machines Learn?*" In Michalski et al.1983.

[Steels 85] L. Steels, J. A. Cambel (editors), "*Progress in Artificial Intelligence,* " Ellis Horwood Limited, 1985.

[Tufail 99] M. M. Tufail, “*Rationalization of Knowledge Representation in Post Operative Expert Medical System,*” MS Thesis, National University Of Sciences and Technology, Rawalpindi, 1999.

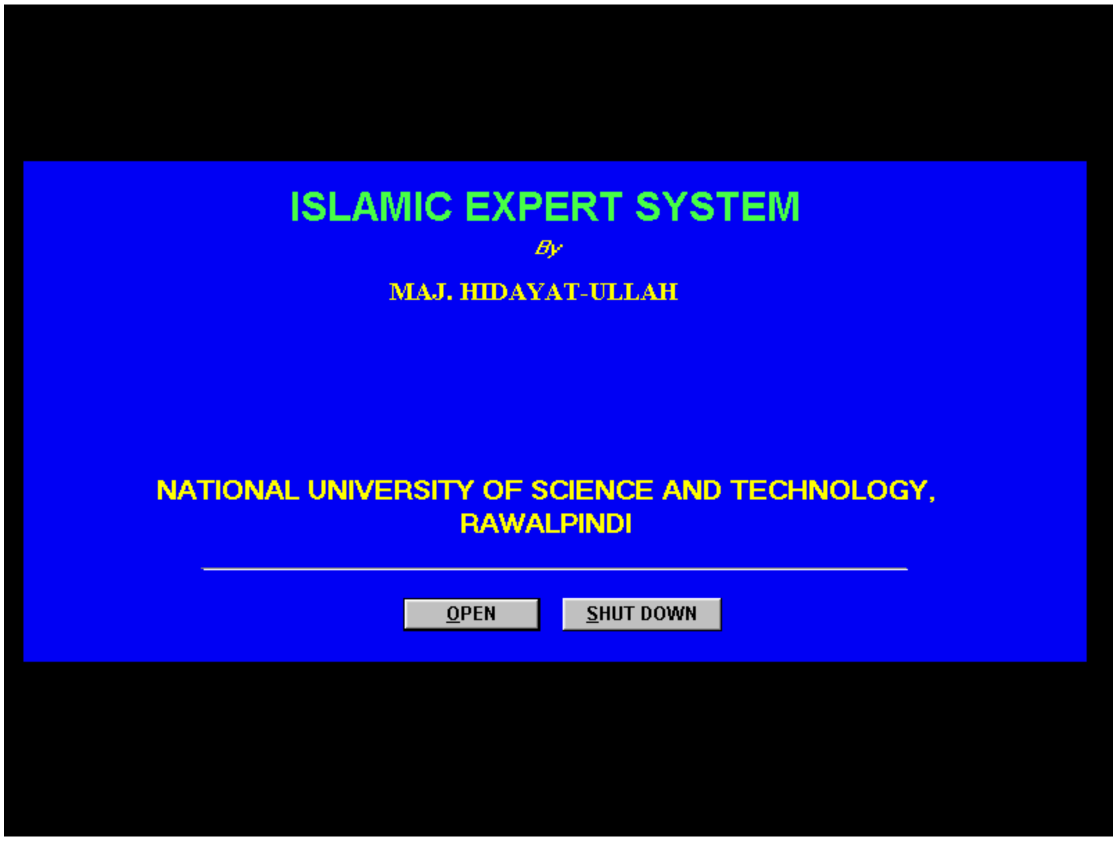
[Waterman 86] Waterman, D.A. “*A Guide to Expert Systems.*” Addison-Wesley. 1986.

[Wilensky 86] Wilensky, R. 1986. Common LISPcraft, W.W. Norton & Company. p. 1-130.

Appendix-1

IEXPERT SYSTEM

GRAPHICAL USER INTERFACE


ISLAMIC EXPERT SYSTEM

By

MAJ. HIDAYAT-ULLAH

NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, RAWALPINDI

Figure A-1 Starting Window

When iexpert start this is the first window which appears. User can press open to login the system or (s)he can shut down the iexpert by clicking the shut down button

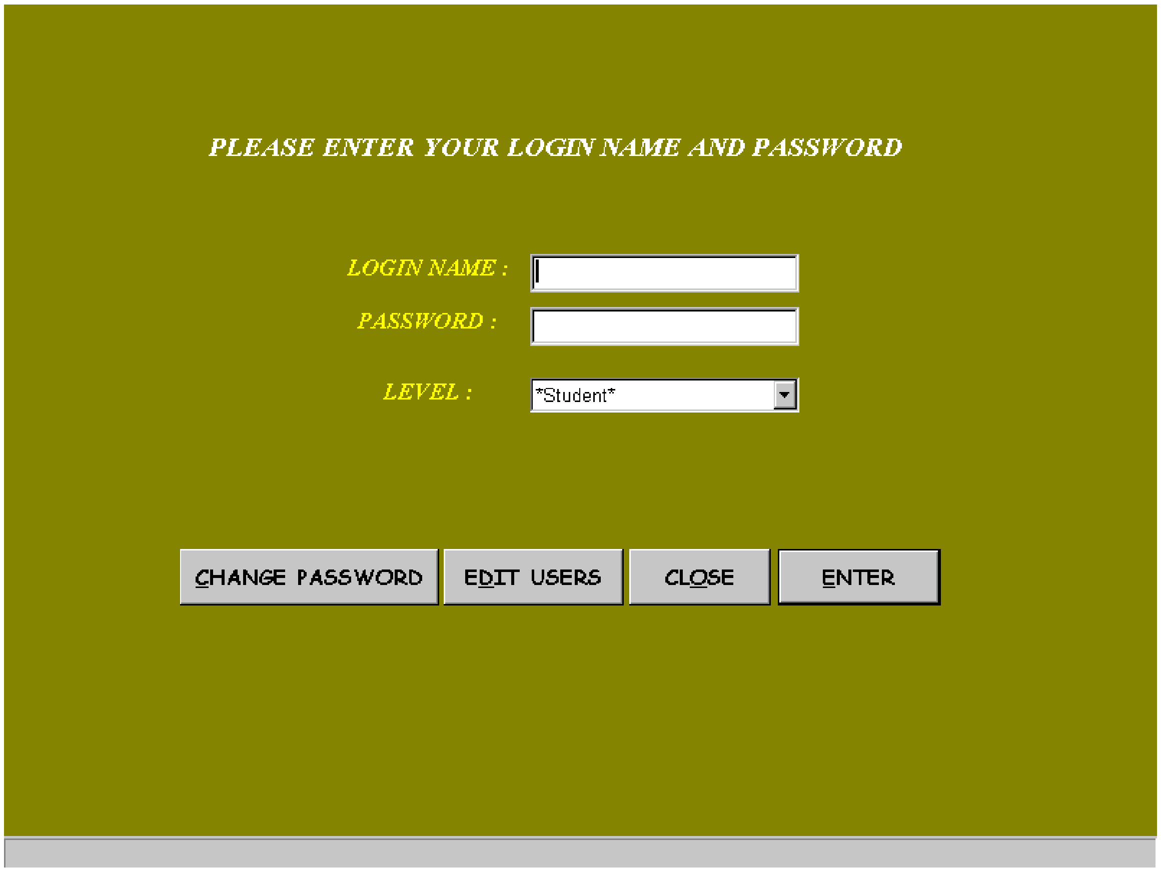


Figure A-2 Login Window

Each user will have to enter its loginame and password along with authority level before entering the system. If the password is worng, user will not be allowed to enter the system. There are two authority levels, common person, and scholar. Each level has its own access and limitations.

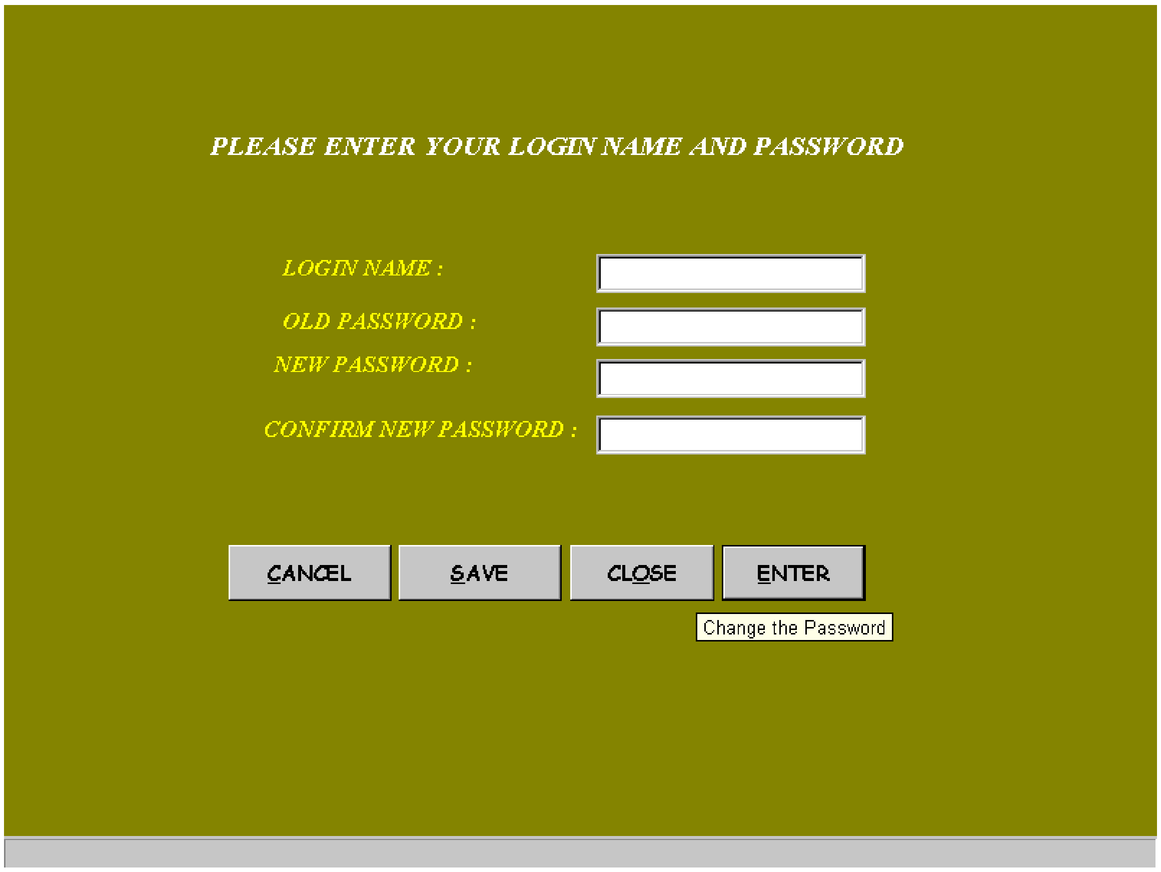
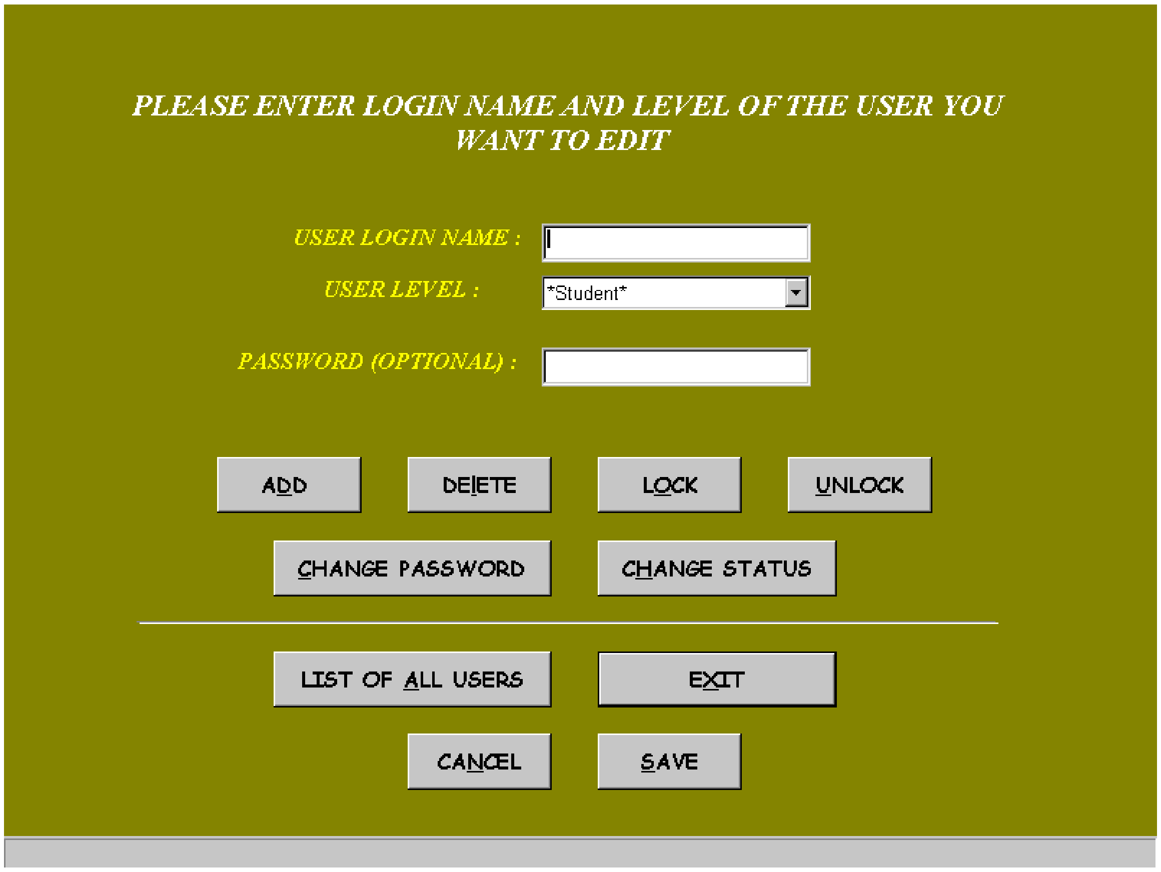


Figure A-3 Change Password Window

Each user can change his password. By entering old password and confiming the new password.


PLEASE ENTER LOGIN NAME AND LEVEL OF THE USER YOU WANT TO EDIT

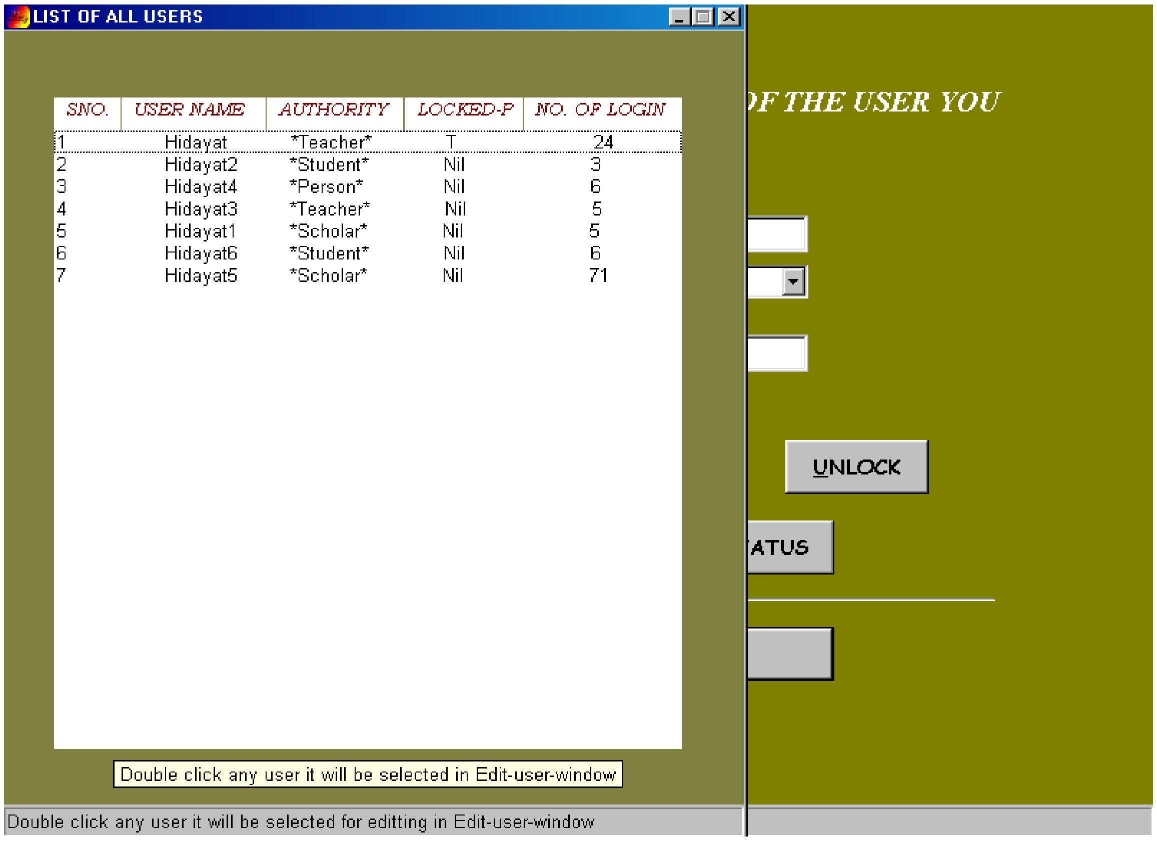
USER LOGIN NAME :

USER LEVEL : *Student*

PASSWORD (OPTIONAL) :

Figure A-4 Edit Users Window

This is the interface for editing the user accounts of the system. Administrator can add new user, delete any user or lock its account. Administrator can change the password or status of the other users.



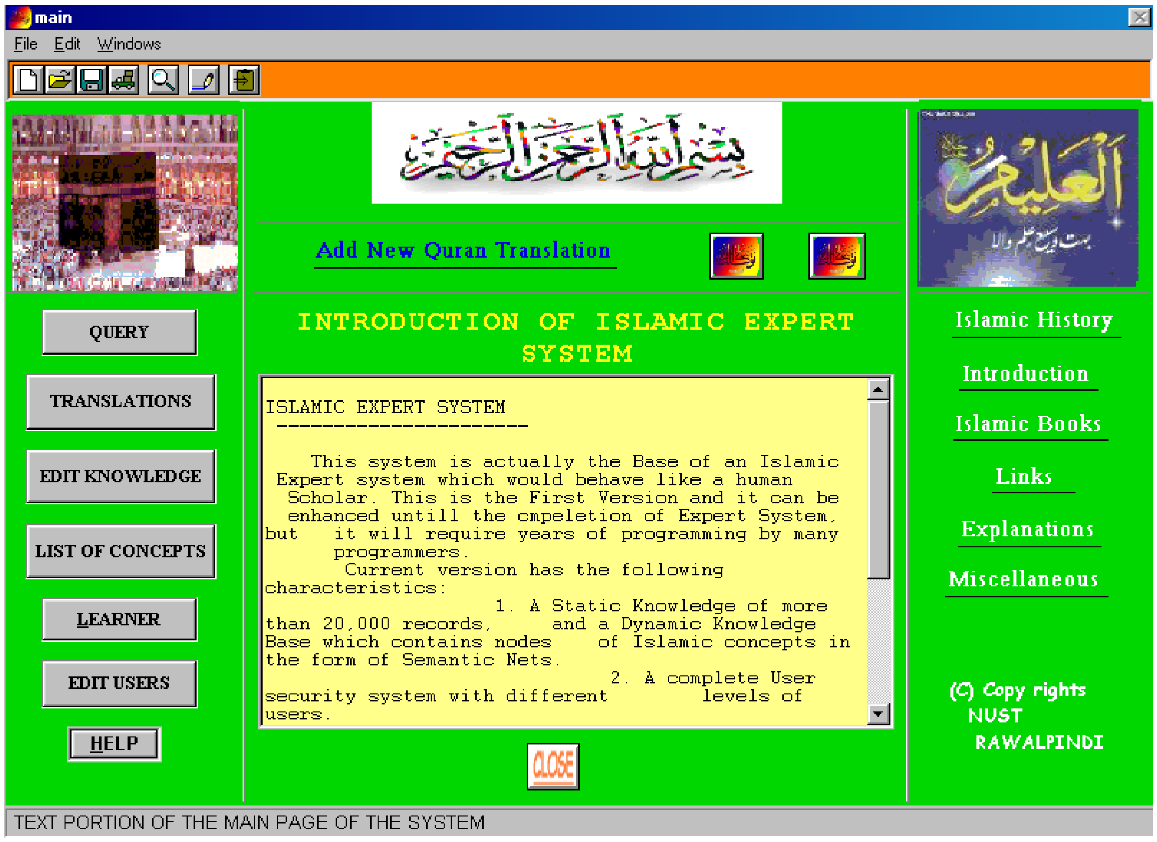
SNO.	USER NAME	AUTHORITY	LOCKED-P	NO. OF LOGIN
1	Hidayat	*Teacher*	T	24
2	Hidayat2	*Student*	Nil	3
3	Hidayat4	*Person*	Nil	6
4	Hidayat3	*Teacher*	Nil	5
5	Hidayat1	*Scholar*	Nil	5
6	Hidayat6	*Student*	Nil	6
7	Hidayat5	*Scholar*	Nil	71

Double click any user it will be selected in Edit-user-window

Double click any user it will be selected for editting in Edit-user-window

Figure A-5 All Users Window

Administrator can see the list of users of the system through this window. He may select any user for editing by double clicking the user name.

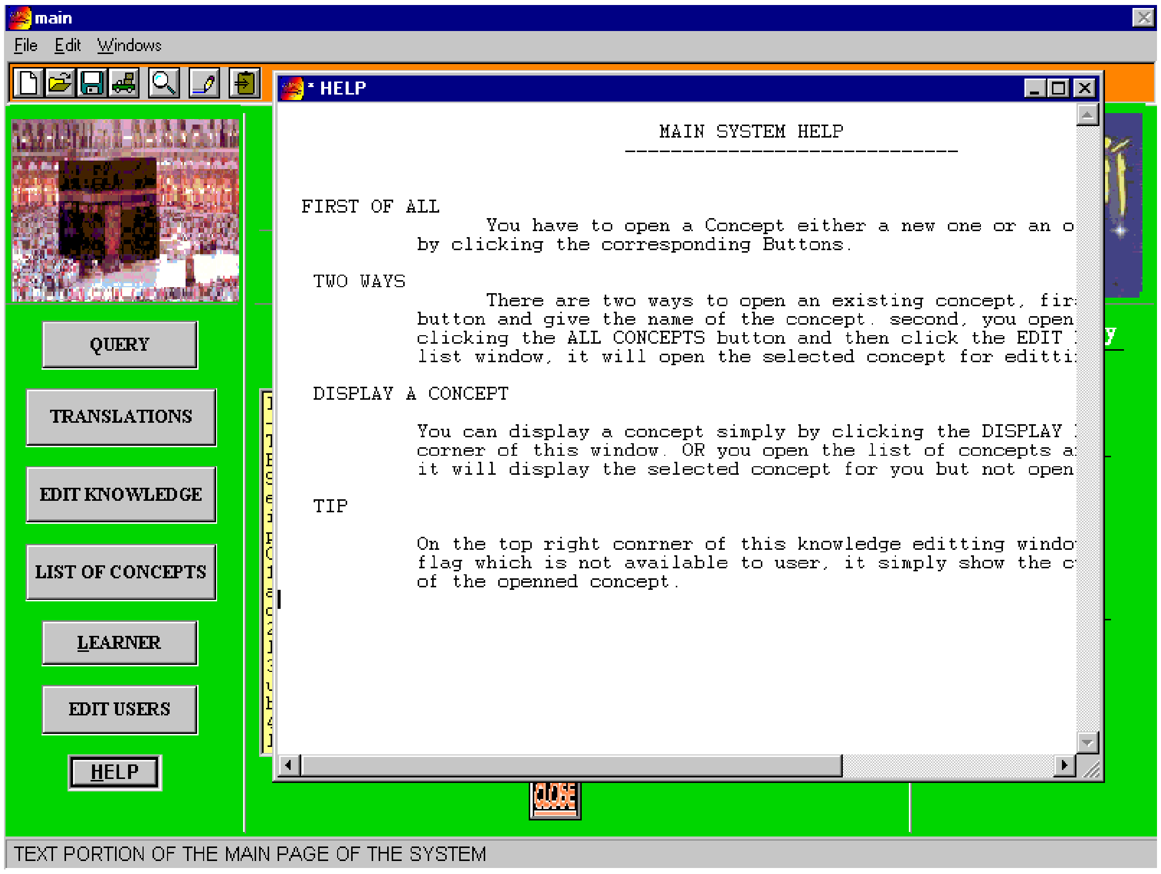


```
ISLAMIC EXPERT SYSTEM
 ----------------------

    This system is actually the Base of an Islamic
 Expert system which would behave like a human
  Scholar. This is the First Version and it can be
  enhanced untill the cmpeletion of Expert System,
but   it will require years of programming by many
      programmers.
       Current version has the following
characteristics:
                    1. A Static Knowledge of more
than 20,000 records,     and a Dynamic Knowledge
Base which contains nodes    of Islamic concepts in
the form of Semantic Nets.
                              2. A complete User
security system with different        levels of
users.
```

Figure A-6 Main Iexpert Window

This is the main window of iexpert. User can user any feature of iexpert through this system. This image is taken while scholar was login. Users with other authority level may have some features turned off.



```
                    MAIN SYSTEM HELP
              ----------------------------


FIRST OF ALL
               You have to open a Concept either a new one or an o
          by clicking the corresponding Buttons.

 TWO WAYS
               There are two ways to open an existing concept, fir
          button and give the name of the concept. second, you open
          clicking the ALL CONCEPTS button and then click the EDIT
          list window, it will open the selected concept for editti

 DISPLAY A CONCEPT

          You can display a concept simply by clicking the DISPLAY
          corner of this window. OR you open the list of concepts a
          it will display the selected concept for you but not open

 TIP

          On the top right conrner of this knowledge editting windo
          flag which is not available to user, it simply show the c
          of the openned concept.
```

Figure A-7 System Help Window

This is Help widow for helping the user in the use of this system. Help is displayed in the form of simple text.

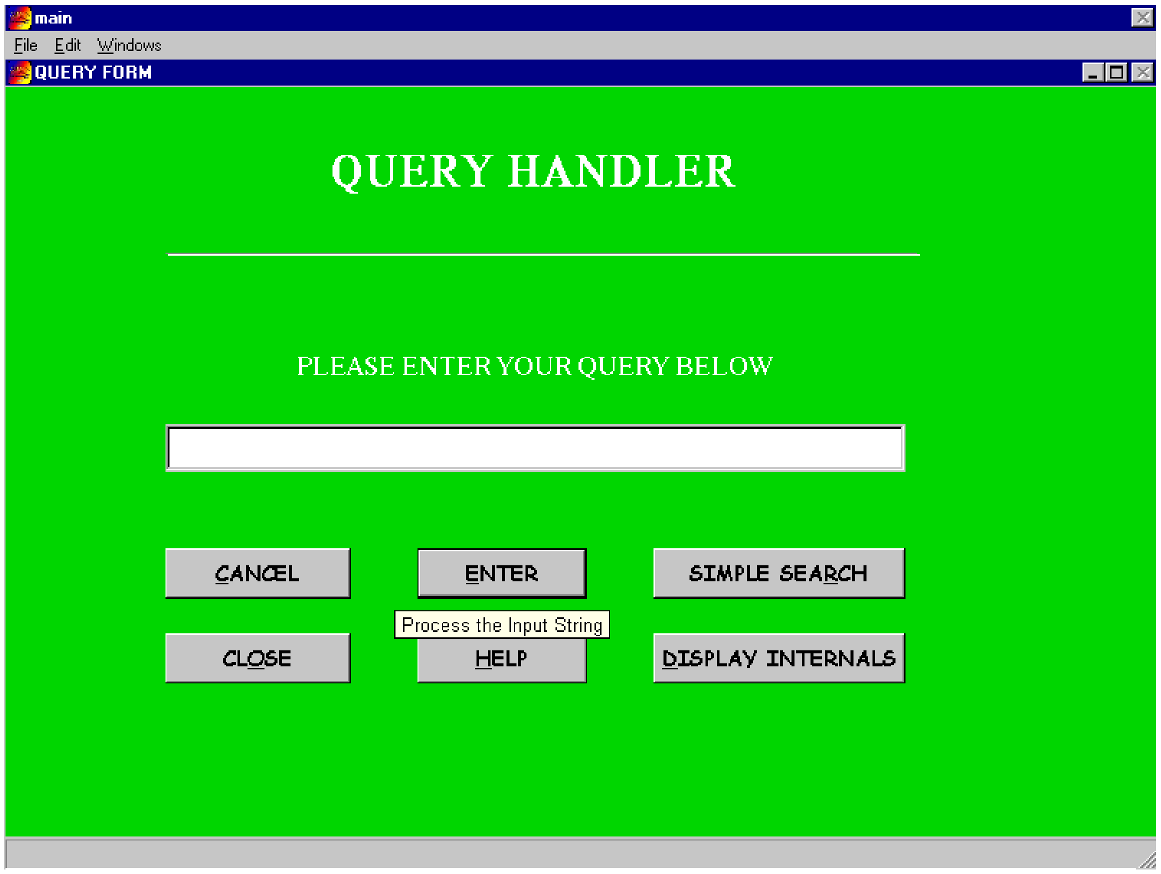
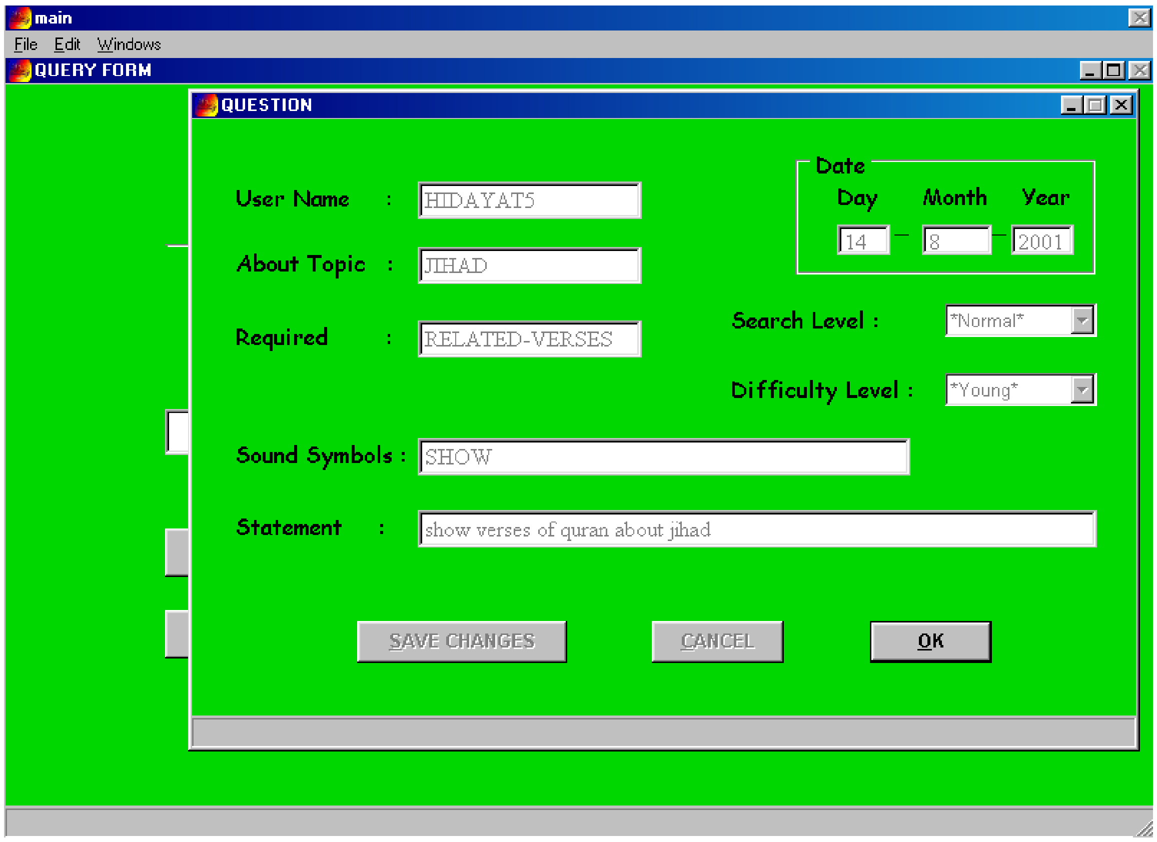


Figure A-8 Query Handler Window

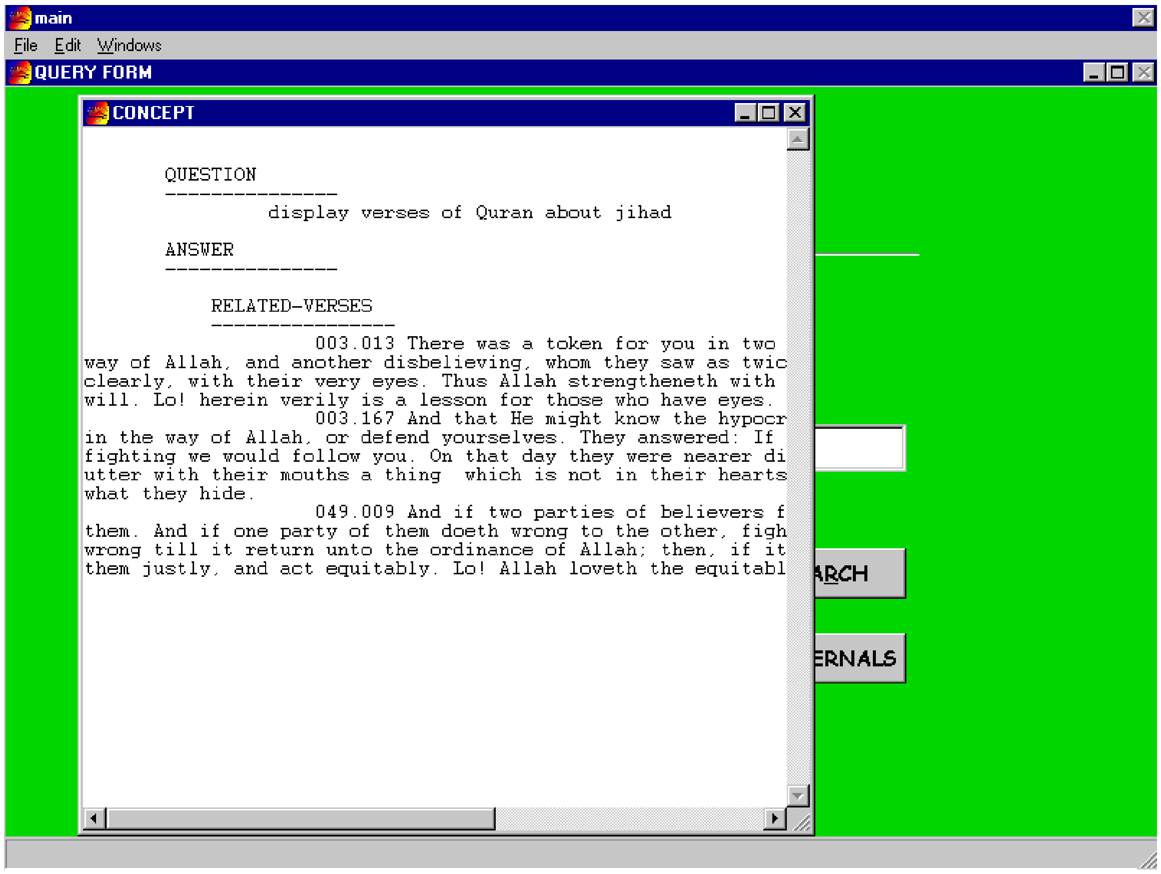
This is interface of the iexpert query handler. User can enter query in the form of simple English sentence in the editable box. Query Handler give its message in the static text, present above of the user editable box.



User Name :	HIDAYAT5
About Topic :	JIHAD
Required :	RELATED-VERSES
Sound Symbols :	SHOW
Statement :	show verses of quran about jihad
Date Day	14
Date Month	8
Date Year	2001
Search Level :	*Normal*
Difficulty Level :	*Young*

Figure A-9 Question Window

This window is showing the internal representation of the question entered by the user. User can only see if the system has comprehended the input string correctly. This window is un-editable and is for explanation purpose.



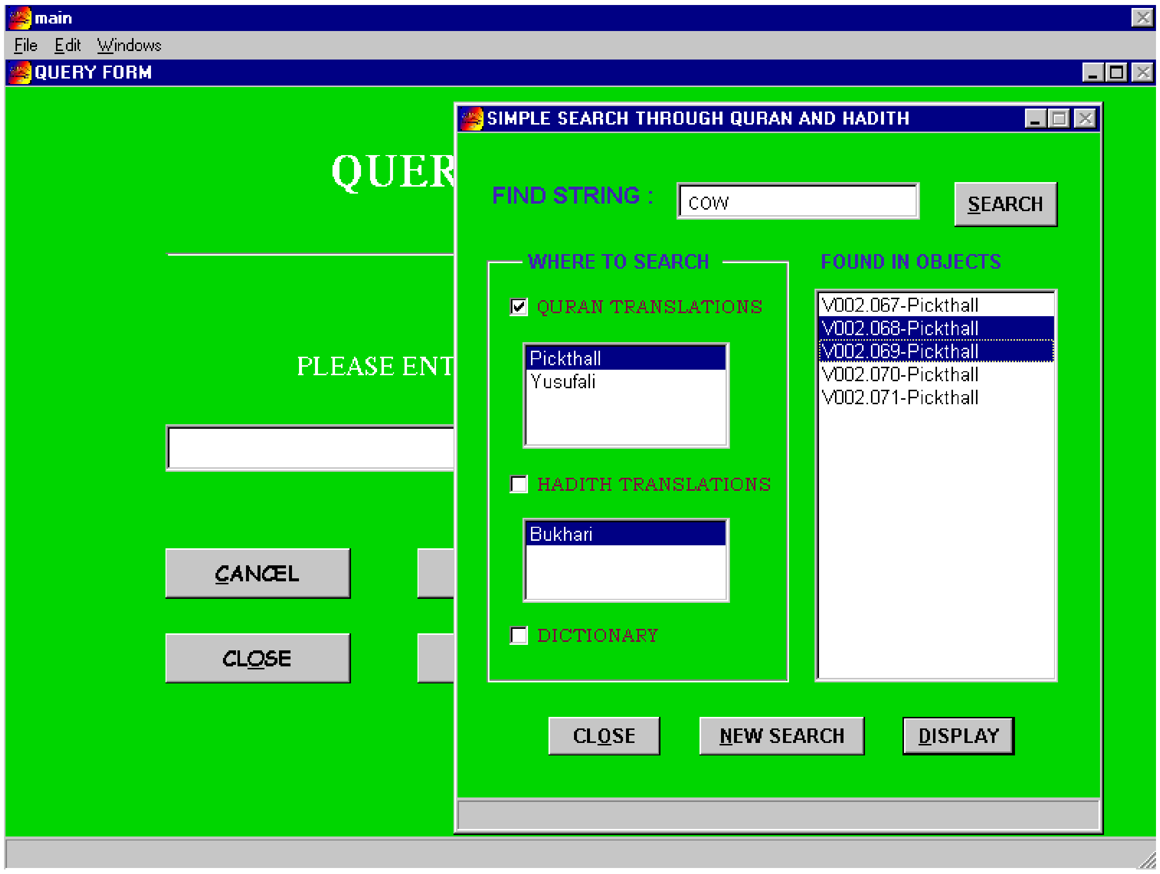
```
       QUESTION
       ---------------
                display verses of Quran about jihad

       ANSWER
       ---------------

           RELATED-VERSES
           ----------------
                    003.013 There was a token for you in two
way of Allah, and another disbelieving, whom they saw as twic
clearly, with their very eyes. Thus Allah strengtheneth with
will. Lo! herein verily is a lesson for those who have eyes.
                    003.167 And that He might know the hypocr
in the way of Allah, or defend yourselves. They answered: If
fighting we would follow you. On that day they were nearer di
utter with their mouths a thing  which is not in their hearts
what they hide.
                    049.009 And if two parties of believers f
them. And if one party of them doeth wrong to the other, figh
wrong till it return unto the ordinance of Allah; then, if it
them justly, and act equitably. Lo! Allah loveth the equitabl
```

Figure A-10 Answer Window

This window is for display the answer of the user query. Answer is displayed in the form of simple formatted text.



FIND STRING : cow

WHERE TO SEARCH

[✓] QURAN TRANSLATIONS

Pickthall
Yusufali

[ ] HADITH TRANSLATIONS

Bukhari

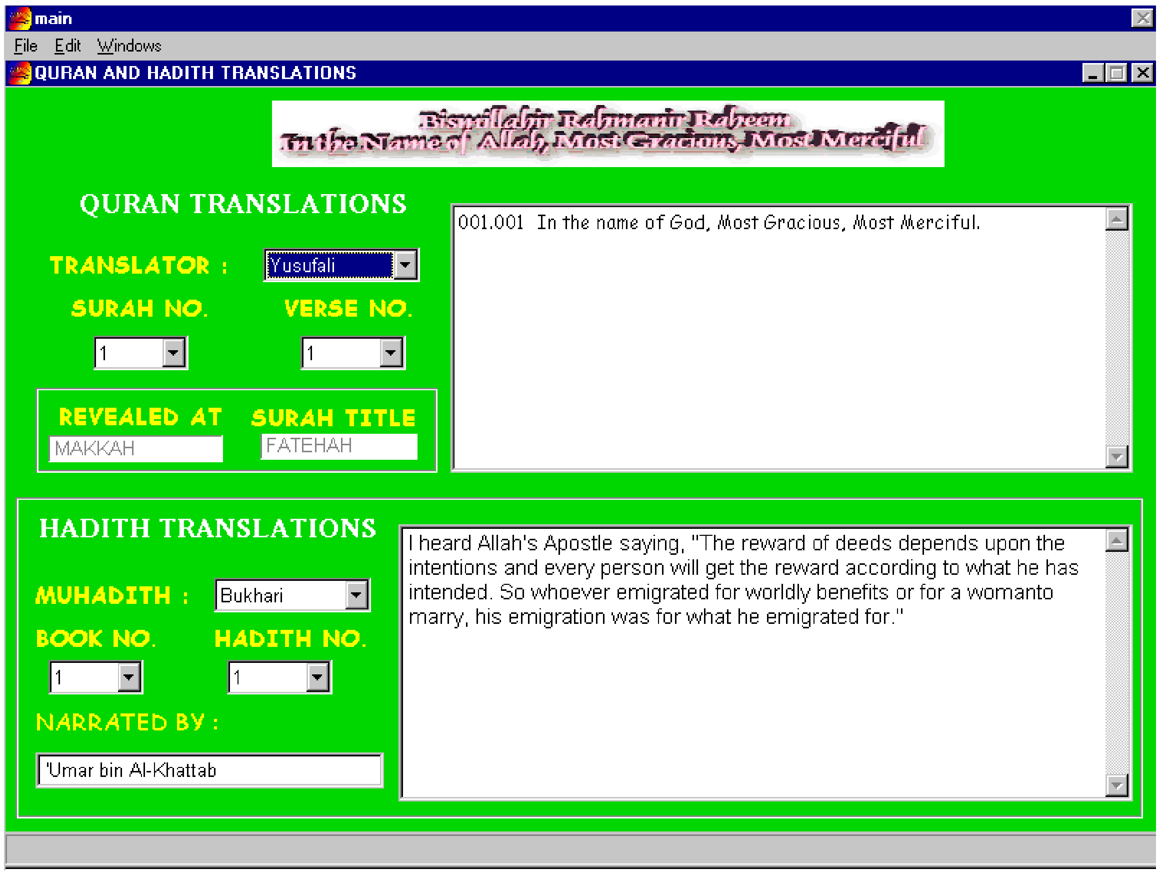
[ ] DICTIONARY

FOUND IN OBJECTS

V002.067-Pickthall
V002.068-Pickthall
V002.069-Pickthall
V002.070-Pickthall
V002.071-Pickthall

Figure A-11 Simple Search Window

User can search any string from translations of Quran and Hadith through this window. Dictionary option is currently not working.


main
File Edit Windows
QURAN AND HADITH TRANSLATIONS
Bismillahir Rahmanir Raheem
In the Name of Allah, Most Gracious, Most Merciful

QURAN TRANSLATIONS

TRANSLATOR : Yusufali

SURAH NO. 1

VERSE NO. 1

REVEALED AT MAKKAH

SURAH TITLE FATEHAH

001.001 In the name of God, Most Gracious, Most Merciful.

HADITH TRANSLATIONS

MUHADITH : Bukhari

BOOK NO. 1

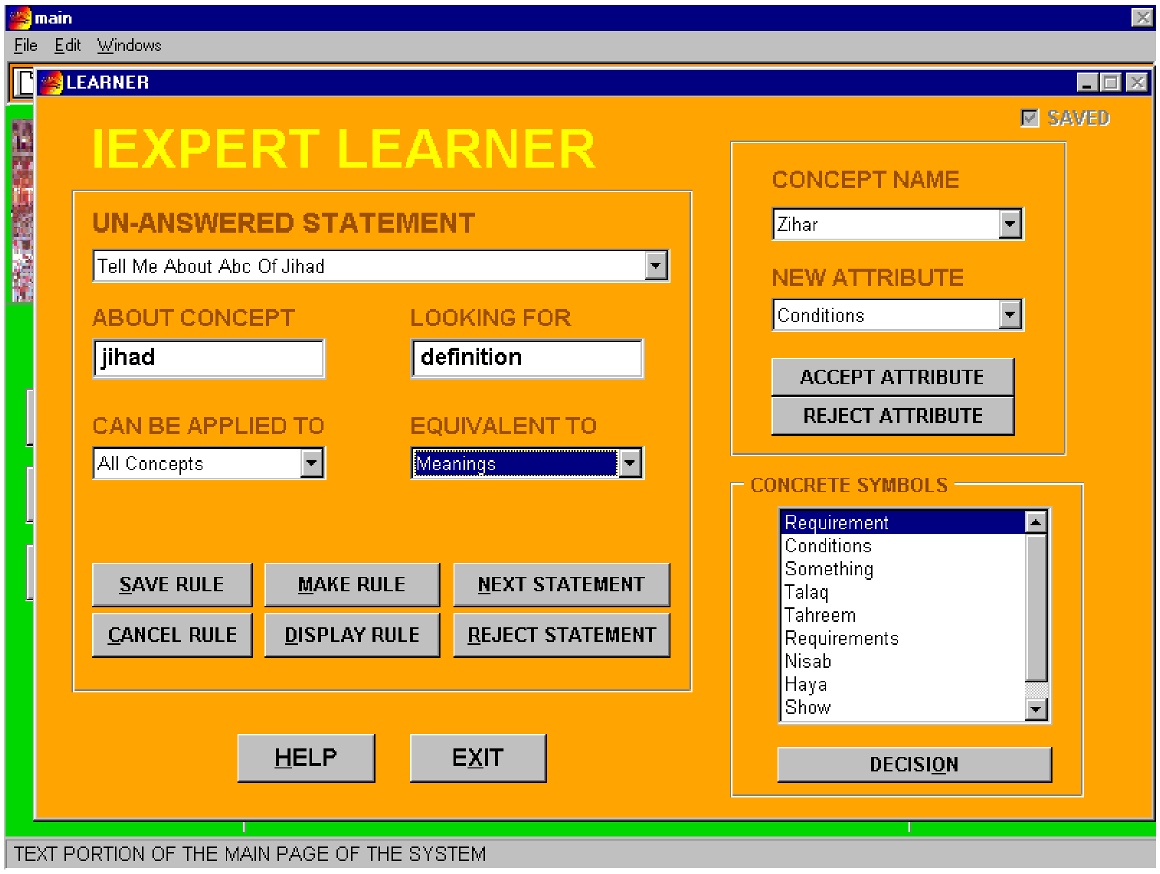
HADITH NO. 1

NARRATED BY : 'Umar bin Al-Khattab

I heard Allah's Apostle saying, "The reward of deeds depends upon the intentions and every person will get the reward according to what he has intended. So whoever emigrated for worldly benefits or for a womanto marry, his emigration was for what he emigrated for."

Figure A-12 Translation Window

This window displays the translations of Quran and Hadith. User can read these translation. Arabic text and recitation of Arabic text is unavailable.



UN-ANSWERED STATEMENT: Tell Me About Abc Of Jihad

ABOUT CONCEPT: jihad

LOOKING FOR: definition

CAN BE APPLIED TO: All Concepts

EQUIVALENT TO: Meanings

CONCEPT NAME: Zihar

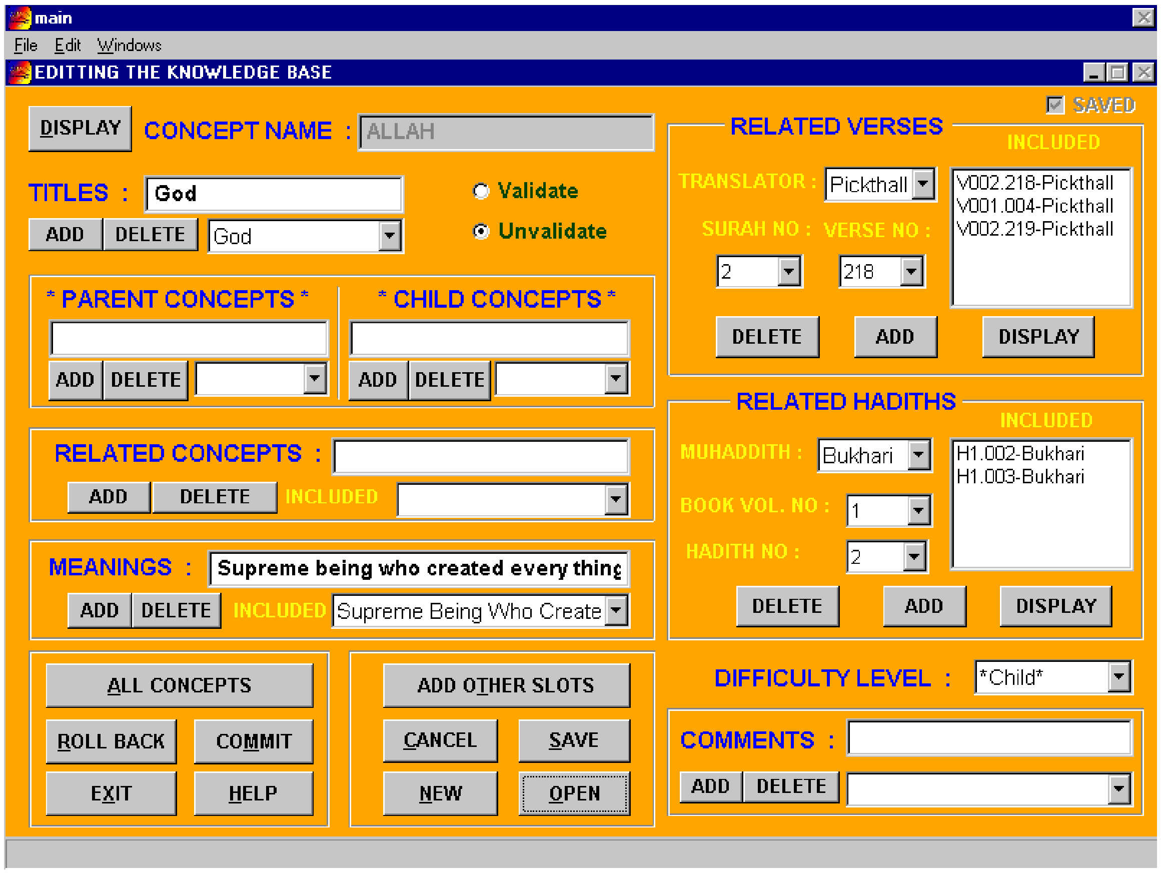
NEW ATTRIBUTE: Conditions

CONCRETE SYMBOLS: Requirement, Conditions, Something, Talaq, Tahreem, Requirements, Nisab, Haya, Show

TEXT PORTION OF THE MAIN PAGE OF THE SYSTEM

Figure A-13 Iexpert Learner Window

System learns from the scholar through this window. It learns the comprehension of un-answered statements. It verifies the learned concepts, slots and their values from human scholar.



[✓ SAVED]

CONCEPT NAME : ALLAH

TITLES : God

God

Validate [✓ Unvalidate]

* PARENT CONCEPTS *

* CHILD CONCEPTS *

RELATED CONCEPTS :

INCLUDED

MEANINGS : Supreme being who created every thing

INCLUDED Supreme Being Who Create

RELATED VERSES

TRANSLATOR : Pickthall

SURAH NO : 2

VERSE NO : 218

INCLUDED

V002.218-Pickthall
V001.004-Pickthall
V002.219-Pickthall

RELATED HADITHS

MUHADDITH : Bukhari

BOOK VOL. NO : 1

HADITH NO : 2

INCLUDED

H1.002-Bukhari
H1.003-Bukhari

DIFFICULTY LEVEL : *Child*

COMMENTS :

Figure A-14 Knowledge Editing Window

This is the knowledge-editing window of iexpert. Scholar edits the knowledge base of iexpert through this window.

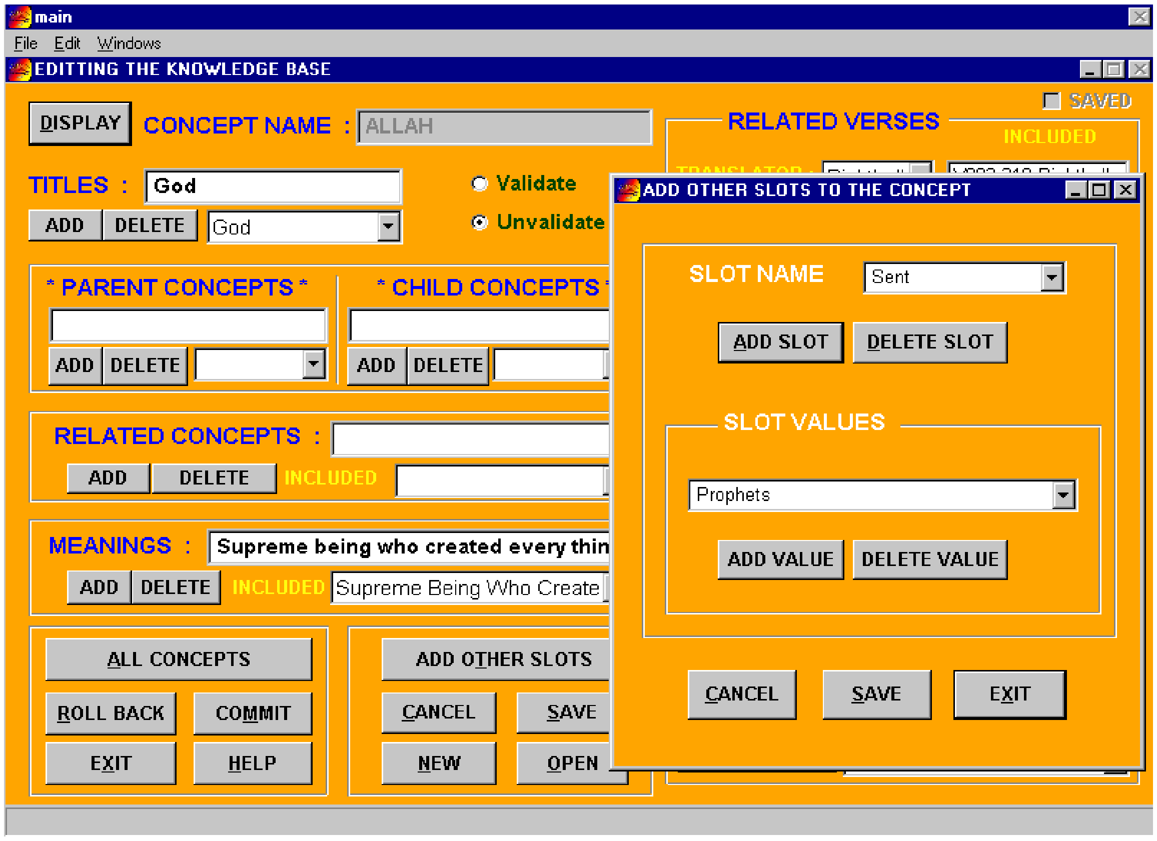
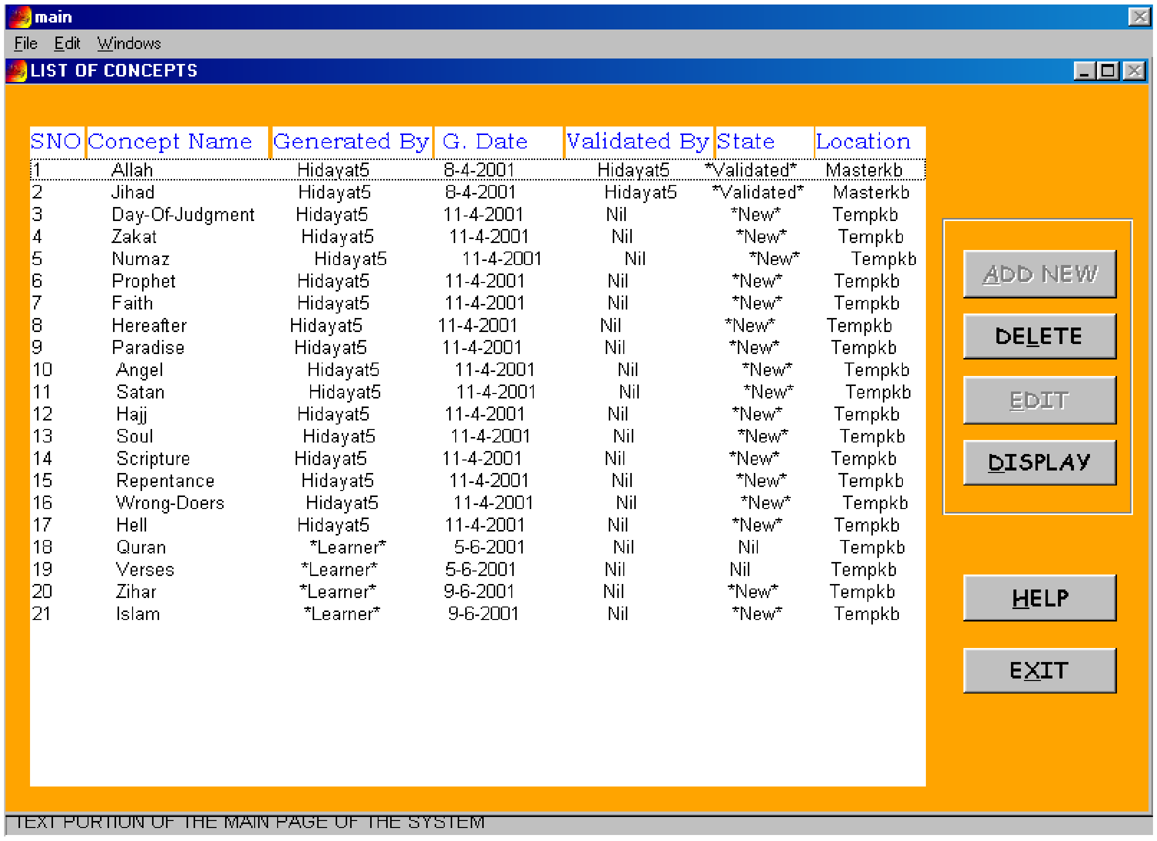


Figure A-15 Other Slots Window

You can see other slot window in the knowledge-editing window. Scholar can add new slots and their values at run time through this window.

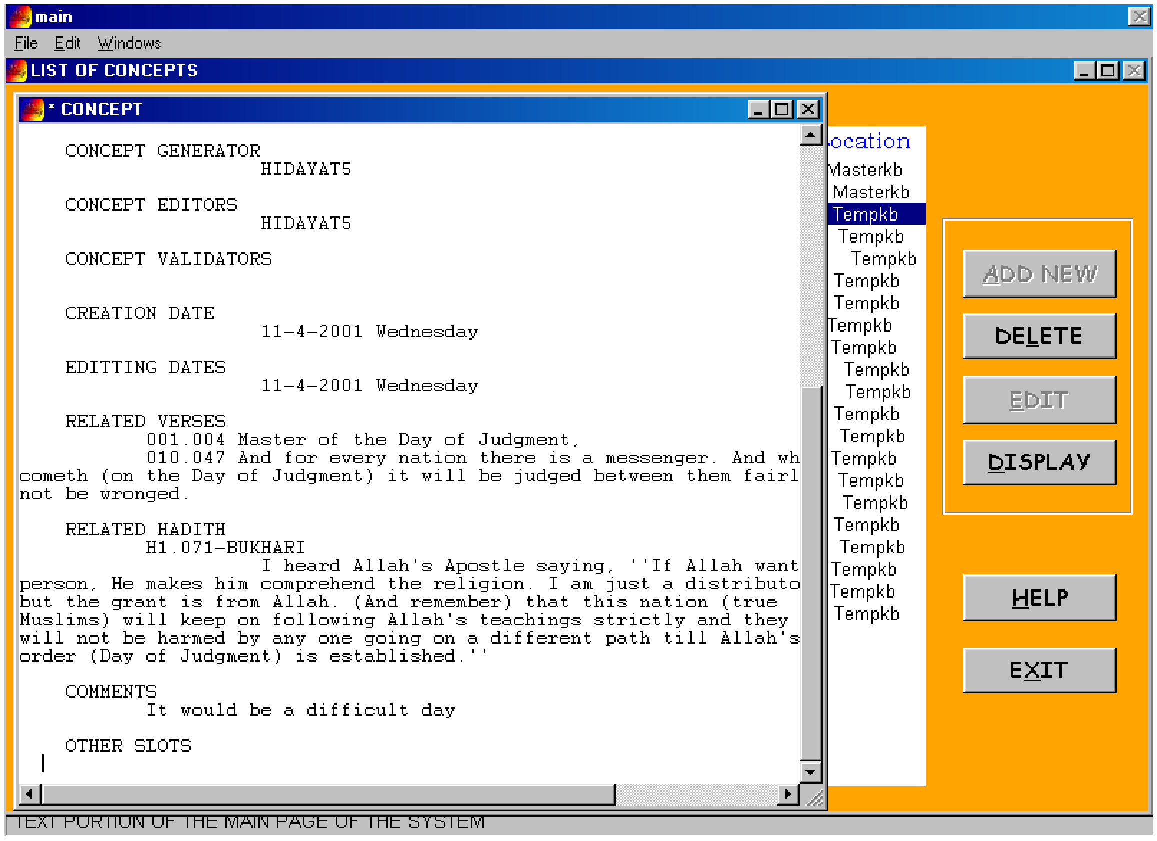


SNO	Concept Name	Generated By	G. Date	Validated By	State	Location
1	Allah	Hidayat5	8-4-2001	Hidayat5	*Validated*	Masterkb
2	Jihad	Hidayat5	8-4-2001	Hidayat5	*Validated*	Masterkb
3	Day-Of-Judgment	Hidayat5	11-4-2001	Nil	*New*	Tempkb
4	Zakat	Hidayat5	11-4-2001	Nil	*New*	Tempkb
5	Numaz	Hidayat5	11-4-2001	Nil	*New*	Tempkb
6	Prophet	Hidayat5	11-4-2001	Nil	*New*	Tempkb
7	Faith	Hidayat5	11-4-2001	Nil	*New*	Tempkb
8	Hereafter	Hidayat5	11-4-2001	Nil	*New*	Tempkb
9	Paradise	Hidayat5	11-4-2001	Nil	*New*	Tempkb
10	Angel	Hidayat5	11-4-2001	Nil	*New*	Tempkb
11	Satan	Hidayat5	11-4-2001	Nil	*New*	Tempkb
12	Hajj	Hidayat5	11-4-2001	Nil	*New*	Tempkb
13	Soul	Hidayat5	11-4-2001	Nil	*New*	Tempkb
14	Scripture	Hidayat5	11-4-2001	Nil	*New*	Tempkb
15	Repentance	Hidayat5	11-4-2001	Nil	*New*	Tempkb
16	Wrong-Doers	Hidayat5	11-4-2001	Nil	*New*	Tempkb
17	Hell	Hidayat5	11-4-2001	Nil	*New*	Tempkb
18	Quran	*Learner*	5-6-2001	Nil	Nil	Tempkb
19	Verses	*Learner*	5-6-2001	Nil	Nil	Tempkb
20	Zihar	*Learner*	9-6-2001	Nil	*New*	Tempkb
21	Islam	*Learner*	9-6-2001	Nil	*New*	Tempkb

TEXT PORTION OF THE MAIN PAGE OF THE SYSTEM

Figure A-16 Concept List Window

This window displays the concepts, which are present in the knowledge base of iexpert.



```
    CONCEPT GENERATOR
                     HIDAYAT5

    CONCEPT EDITORS
                     HIDAYAT5

    CONCEPT VALIDATORS


    CREATION DATE
                     11-4-2001 Wednesday

    EDITTING DATES
                     11-4-2001 Wednesday

    RELATED VERSES
           001.004 Master of the Day of Judgment,
           010.047 And for every nation there is a messenger. And wh
cometh (on the Day of Judgment) it will be judged between them fairl
not be wronged.

    RELATED HADITH
           H1.071-BUKHARI
                     I heard Allah's Apostle saying, ''If Allah want
person, He makes him comprehend the religion. I am just a distributo
but the grant is from Allah. (And remember) that this nation (true
Muslims) will keep on following Allah's teachings strictly and they
will not be harmed by any one going on a different path till Allah's
order (Day of Judgment) is established.''

    COMMENTS
           It would be a difficult day

    OTHER SLOTS
```

```
ocation
Masterkb
Masterkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
Tempkb
```

TEXT PORTION OF THE MAIN PAGE OF THE SYSTEM

Figure A-17 Display Concept Window

This window displays the contents of a concept in the form a formatted text.

SYSTEM STATISTIC AND CONVENTIONS

- There are eighty five (85) Files of system
- Approximately three hundred and twenty five (325) Programs of Lisp, Functions and Methods are included in the code of this system.
- It takes approximately two hundred (200) Seconds to load.
- Twelve (12) Classes have been defined in the system
- There is Database of approximately twenty thousand (20,000) records.
- System takes thirteen (13) seconds to search a string from translations of Quran and Hadith.
- Initially there are twenty (20) concepts in this system.
- There are twenty seven (27) global variables, out of which fifteen (15) are system variables and twelve (12) are learner variables, in the system code.
- System functions start with 'iexpert-' symbol.
- Files with names ending with'-iexpertfile.lsp' are being used by the system to write at runtime.
- Brown color is for security windows.
- Green color of windows is for those which are accessible by all authority levels of users.
- Orange color windows are only accessible by the Scholars.

My Thanks Again to

all of those who ever helped me

or taught me a single thing.

I am also thankful to YOU
for reading these pages.

PLEASE contact me with more ideas.

May God help us in learning more knowledge and wisdom.

(Hidayat-Ullah Khan)