Chapter 1
Expert Systems: Introduction

Chapter Abstract:
The chapter describes the definitions of the expert system, history of the expert systems, how expert systems have evolved over the last three decades, general architecture adopted by expert system and its characteristics, advantages and disadvantages of providing solution using expert systems, and description of the different categories of expert systems. The chapter also describes applications of expert systems along with the examples of some of the well-known expert systems in the world studied by us. While describing the applications of expert systems in general, the chapter focuses more on the application of expert system in the human resource domain, the area the thesis addresses. The chapter also provides a detailed discussion on the different types of expert systems. At the end, the chapter discusses the tools, shells and languages for the development of expert systems

1.1 Introduction
Expert System is a special branch of Artificial Intelligence. (Efraim Turban, 1998) AI is a study of human thought process and modeling it into a computerized system. It is considered a study of how to make computers do things at which, at the moment people are better. (Knight, 1991)

Expert systems are used to perform a variety of extremely complicated tasks that could only be performed by a limited number of highly trained human experts. (Rolston, 1988) It deals with real world problems and provides solution, which reflect human judgment and intuition.

An expert system is an interactive computer based decision making tool that uses both facts and heuristics to solve difficult decision making problems, based on knowledge acquired from an expert. An expert system is also known as knowledge based system.
The following example will illustrate why expert systems differ from conventional computerized applications.

When you are suffering from illness and visit the doctor. Doctor will ask you a few questions, he/she will check your pulses, examines eyes, measures heart beats, measure temperature of your body and after a series of interaction, he will finally conclude that you are suffering from viral fever. The doctor diagnoses your illness easily and so accurately because he has acquired expertise in his field after a lot of study and experience in generating heuristics about the domain. Similarly, if an application program interacts with the patient and provides a diagnosis, then it can be considered as an expert system in the medical field.

The conventional application program will not be able to provide the decision on the basis of guts, intuitions and experience, while the expert system does. The conventional system is made up of algorithms and databases, while an expert system consists of inference engine and knowledge.

The table 1.1 differentiates between a conventional system and expert systems.

<table>
<thead>
<tr>
<th>Table 1.1 Difference between conventional system and expert system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional System</td>
</tr>
<tr>
<td>Solves the generic numeric problems.</td>
</tr>
<tr>
<td>It is sequential program where information and processing are combined.</td>
</tr>
<tr>
<td>Tested program never makes mistakes</td>
</tr>
<tr>
<td>No explanation is provided for output</td>
</tr>
<tr>
<td>When incorrect information is provided, the system may not function.</td>
</tr>
</tbody>
</table>
Now we will explore the definition of the expert system.

1.2 Definition

The expert system is an application which mimics human intelligence. To elaborate more, it is an application that solves complicated problems that would otherwise require extensive human expertise. It simulates the human reasoning process by applying specific knowledge and inferences.

As per Gaschnig, Reboh and Reiter, “expert systems are interactive computer programs incorporating judgment rules of thumb, intuition and other expertise to provide knowledgeable advice about a variety of tasks.” (Akerker, 2007)

As per JM & Co, “an expert system is a program that emulates the interaction a user might have with human expert to solve a problem. As per KSR Anjaneylnu, “An expert system encodes human expertise in limited domain by representing if-then-else rules.”

Now let us look into the history of the expert system development.

1.3 History of expert system

The term expert system has been coined first time in the late 1950s. Lisp, which is the programming language for AI was developed by John McCarthy in 1958. A predecessor to expert system was GPS (General purpose problem solver), which was developed by Newell and Simon. The GPS was not successful, but it was the starting point in the direction of expert systems. The first expert system was Dendral in 1965 by Stanford. Later on, many expert systems have been developed in different areas and domains.
In a prestigious international conference on AI, 1977, Professor Feigenbaum presented that “the power of expert system is derived from the knowledge it possesses not from the particular formalisms and inference schemes it employs.”

The table 1.2 lists the some of the early expert systems developed before 1980s. (Rolston, 1988)

<table>
<thead>
<tr>
<th>System</th>
<th>Domain Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dendral</td>
<td>Infers information about chemical structure</td>
</tr>
<tr>
<td>Macsyma</td>
<td>Performs complex mathematical analysis</td>
</tr>
<tr>
<td>Hearsay</td>
<td>Natural- language interpretation for subset language</td>
</tr>
<tr>
<td>Mycin</td>
<td>Diagnosis of blood disease</td>
</tr>
<tr>
<td>Teiresias</td>
<td>Knowledge transformation tool</td>
</tr>
<tr>
<td>Prospector</td>
<td>Mineral exploration and identification tool</td>
</tr>
<tr>
<td>R1(XCON)</td>
<td>Configuration of DEC computer equipment</td>
</tr>
<tr>
<td>Caduceus</td>
<td>Diagnostic tools for internal medicine</td>
</tr>
<tr>
<td>PUFF</td>
<td>Medical system for diagnosis of respiratory conditions</td>
</tr>
<tr>
<td>Prospector</td>
<td>Used by geologists for identifying sites for mining and drilling</td>
</tr>
<tr>
<td>Design Advisor</td>
<td>Gives advices to the designers of processor chips</td>
</tr>
<tr>
<td>Lithian</td>
<td>Gives advice to archaeologists examining stone tools</td>
</tr>
</tbody>
</table>

It will be interesting to note that out of the above mentioned system, 1/3 of them are very successful and were in used till 21st century. (Efraim Turban, 1998)

**1.4 Architecture of expert system**

The figure 1.1 shows a general architecture with the typical components.
Figure 1.1 General architecture of expert system

User:
The user can be the end user who will use the system in the absence of expert or the customer. The user can be knowledge engineer who will generate knowledge using heuristics and existing knowledge base. The real life expert can also be user to verify and validate the results of the expert system.

The roles of users interacting with expert systems can be classified mainly into the following categories:

1. **Domain expert**: An individual who is an expert in the field to solve the problem. The expertise is to be captured in the expert system.
2. **Knowledge Engineer**: An individual who encodes the expert’s knowledge into the knowledge base. He is a person who is capable of designing, building and testing an expert system.

3. **Programmer**: A person who is responsible for the actual programming for the expert system.

4. **Project Manager**: He is a leader of the expert system development team, responsible for keeping projects on track. (Negnevitsky, 2008)

5. **The End user**: A person who will use the expert system when it is deployed.

**User interface (UI) facility**:  
The UI facility must accept information from the user and translate it into a form acceptable to the system and accept information from the system and convert it to a form that can be understood by the user. (Rolston, 1988)

The UI consists of a natural language processing system, which accepts and returns information in the same form as accepted by a human expert. The user interface can consist of the following styles: Question and answer, menu driven, command prompt or graphical user interface.

**Knowledge base (KB)**:  
It represents a storehouse of the knowledge primitives (facts, procedure rules and heuristics) available to the system. In general, knowledge is stored in the form of facts and rules, but knowledge schema can vary for different knowledge base of expert systems. Knowledge can be classified as declarative (facts) and procedural. The design of knowledge schema affects the design of an inference engine, knowledge update process and explanation facility.

**Knowledge Engineering**:  
It is the process of acquiring specific domain knowledge and building it into the knowledge base. A knowledge engineer is the person who acquires the knowledge from the domain expert and transports it to the knowledge base. (Rolston, 1988)
Inference Engine:

It is considered as a brain of expert systems. An Expert system must be flexible to cope with different situations. An ES must be able to infer new knowledge from the existing knowledge base depending upon the different situations. For example, consider the two basic facts:

1. All cricketers have good income.
2. Sachin is a cricketer.

From the above two facts, new fact “Sachin has good income” can be inferred.

Knowledge Update:

Over a period of time, in certain domains knowledge change. In the complex domain like medical, pharmacy it expands. The knowledge base in expert system should be modified according to that. The knowledge update facility is used to perform such updates. The most advanced stage of knowledge update facility is automation of knowledge update where the system automatically generates new knowledge based on the generalization of past experience.

Explanation system:

An expert provides reasoning that led to the conclusion. This enables the expert system to review its own reasoning and explain its decision. An expert system should have that facility and that will differentiate expert system from traditional computer systems.

1.4.1 Role of knowledge in expert system

An expert system derives its power from extensive domain specific knowledge rather than normal understanding of general functionality. For example, we all know that when the body temperature is high, we should take paracetamol tablet to control the body temperature, but an expert physician knows that whether paracetamol will be effective and should be given to the patient or not.

The following are the components of the knowledge: Facts, Procedural rules and Heuristics. Facts are truth regarding the subject domain. Procedural rules define
sequence of events and relations among them to a particular domain and heuristics are rules of thumb to be followed when invariant procedural rules are not available.

Now from the definition and architecture of the expert system, we can derive the characteristics of the expert systems.

### 1.5 Characteristics of expert system

The following are characteristics of an expert system.

- It solves a difficult problem in a domain as good as or better than human experts. (V S Janakiraman K. S., 2005)
- It should have a huge domain specific knowledge to the minute details.
- It should permit the use of heuristic search process.
- It should be able to justify the answers and conclusions.
- It should be able to accept advice, modify, update and expand.
- It should be able to deal with uncertain and irrelevant data like human experts.
- It should provide user interface in the natural language to the user.
- It should respond to the problem in a reasonable amount of time in case of real time systems. (Padhy, 2005)
- It should operate as an interactive system. It means it should answer the question, provide clarification, and help in decision making by recommending solutions.
- It should have a mechanism for filtering and updating knowledge on a continuous basis.
- It should be able to make inferences on the basis of knowledge stored in the knowledge base.
- It should have high performance standard in terms of accuracy and consistency.
- It should be able to manipulate symbolic information and draw conclusion from it.

Now let’s explore the advantages and limitations of expert systems.
1.6 Advantages of using an expert system

The following are the advantages of the expert systems.

- Expert system serves as an archive of knowledge and also acts as a knowledge disseminator.
- Expert system provides answers to all ‘Wh’ questions, so users can understand reasoning process.
- The Expert system accepts commands in natural language of the user.
- Expert system never relies on mood and stress, so it cannot make the wrong decision. It provides greater reliability. (V S Janakiraman K. S., 2005)
- It is available 24*7 unlike human experts.
- The knowledge transfer and expertise can be made available in remote locations as well.
- The duplication of another expert system is easy.
- Expert systems are assets in the interdisciplinary domains where multiple experts are needed.
- The answers and reasoning are consistent with an expert system.
- It reduces decision making time.
- It captures the scarce expertise.
- In the long run, expert systems are cost effective.

Now we will explore the limitations of the expert system.

1.7 Limitations of expert system

The following are the basic limitations of the expert system.

- Expert systems do not have common sense.
- It is hard to extract knowledge and code them for vast areas of domains. So it has only limited applicability.
- There may be a different approach adopted by different experts for the same problem in the same environment and all approaches are correct. It is hard to decide which is the best approach.
- If knowledge representation is faulty, the expert system will provide wrong results.
• Expert system answers ‘Why’ with reasoning, but doesn’t answer ‘Why not’.
• The cost for developing expert systems is very high.
• In case of human experts, performance degradation is gradual, while in the case of expert system, it is steep.
• Expert system works only for very specific limited areas of the domain.
• The requirement of knowledge engineer can be a bottleneck for the expert system.
• Users of expert systems have natural cognitive limits.

So, now the next issue is to explore where an expert system can be an effective alternative for the human expert.

1.7.1 Situations where expert systems are effective alternative

In the following situations, an expert system can be considered as an effective alternative for the human expert.

• When multiple experts are needed at different locations simultaneously.
• Experts have to work in the environment, which is hazardous for health, i.e. mining, drilling etc.
• When firm doesn’t want its expertise to get retired.
• When work is monotonous and of repetitive nature.
• When knowledge and expertise can be obtained through experience.
• The legal rules and code of conduct changes rapidly.

1.7.2 Reasons why expert system fails

Following are the reasons where design and development of an expert system fails.

• Lack of acceptance by the end user.
• Inability to retain current developers.
• Problems manifolds during maintenance of the system.
• Lack of top management support in organizing and funding.
• Error in expert knowledge can propagate throughout the entire system.
1.7.3 Reasons why expert system success
Following are the reasons where design and development of an expert system successful.

- Required in-house champion
- User involvement and training
- An organizational environment for new technology

1.8 Categories of expert system
Expert systems are categorized into three general classes: (Davis, 1985)

1. Assistant – Here expert system act as a subsystem and the role of expert system is more or less similar to Decision Support System, which assists in the decision making process. The system doesn’t have its own capacity to take intellectual decision.

2. Colleague – The system acts as a peer and provides its expertise in a small domain of decision making problems. E.g. Mapping user queries to FAQs

3. Expert – This is a true category of expert, and system acts as a real world expert and takes intelligent decisions in the absence of human experts.

1.9 Examples of expert system
Before we discuss the applications of expert system in the human resource domain, we will discuss some very well-known examples of expert system. The following examples just represent few domains to get an idea on what are different applications of expert system in different domains.

ELIZA: - It allows the user to type a sentence through the keyboard and the computer would respond with a sentence of its own.

PARRRY – The Artificial Paranoid: - It simulates the responses of a young man suffering from paranoid schizophrenia. The user will type queries or comments and responses are printed by the program.
Design of expert system prototype for analysing and structuring motivational strategies on ICT human resources to reduce employee turnover ratio

**Internist:** - It is used for medical diagnosis. It is a diagnostic aid for all internal medicine.

**NAVEX:** - It is used for monitoring signals. It monitors, radar data and estimated the velocity and position of the space shuttle.

**MOLGEN:** - It planned chemical processes whose purpose were to analyze and synthesize DNA.

**SOPHIE:** - It is used to teach students a body of knowledge and varying teaching based on assessment of learning ability of the student. It instructs the student on the repair of an electronic power-pack.

**PLANT:** - It is used to forecast future events using a model based on past events. It predicts the damage to be expected when a corn crop was invaded by black cutworm.

**COOKER Adviser:** - It is used for generating remedies for system faulty. It provides repair advice with respect to canned soup sterilizing machines.

**Ventilator Management Assistant:** - It governs the behavior of the system by anticipating problems, planning solution, and monitoring actions. It is used to scrutinizing data from hospital breathing support machines, and provides accounts of patient conditions.

**Crop Advisor:** - It advises cereal grain farmers on appropriate fertilizers and pesticides for their farms. Farmers can access the system via internet.

**Optimum-AIV:** - It is a planner used by the European space agency to help in the assembly, integration and verification of space craft.

**KISAN:** - It is used for soil nutrient management. It uses the knowledge of visual deficiency symptoms evident in the plants to diagnose the nutrient
Now we will discuss different application areas of an expert system.

1.10 Applications of expert systems

Expert systems are used in playing chess, making investment decisions, diagnosis of illness, configure computers, monitor performance of complex systems, underwriting insurance policies, and career counseling, etc. (Chakraborty R., 2010) In a brief, the application areas of expert systems are classified into the following categories.

1. Analysis
2. Control
3. Designing
4. Diagnosis
5. Instruction
6. Monitoring
7. Planning
8. Prediction
9. Repair
10. Interpretation

1.11 Applications of expert system in human resources domain

Now we will try to figure out applications of expert system in different subdomains of the human resources domain. Basically, human resource is a domain which requires human expertise to solve the problems like recruitment, selection, performance appraisal, training and development, motivation and leadership, and employees’ welfare. The reason behind doing the study is to list out different areas where expert system can be design and developed or there is scope of development of an expert system.

**Recruitment:** An expert system can be developed for planning the gap between human resources supply and demand. It can also help with strategic planning of recruitment of key personnel and succession planning of retiring employees. The system can deal with employee layoffs as well dismissal of an employee. It can also help in managing different sources of recruitment like campus, referrals, walk-ins etc.
Job Analysis: An expert system can be developed for job analysis for different designation and fitting each job in the organisational hierarchy. It can also help in preparing an organisation chart based on job description and job specification. It can also list job characteristics and match employee characteristics to find the suitable employee for each job.

Selection: An expert system in this area can deal with application blank, aptitude, personality and performance test, applying selection parameters on potential candidates, and help organisations to have a successful selection of the right candidates. It can also help organisations in improving the selection ratio as unskilled and unqualified candidates are screened in the first round of the selection process.

Performance evaluation and appraisal: An expert system in this area can deal with measuring the employees’ performance using the right tools and methods and based on the result, decide the best performance appraisal for each employee individually. It can help in deciding the best payment methods, pay structure, increment range keeping in mind the organisation total profile amount. The expert system here can help in deciding the evaluation policies which are acceptable to both employees and employers.

Training and development: An expert system in this area can help organisation in identifying training needs by finding out the gap between skills demanded by the job and skills possessed by employees. The main objective for the system can be to match the organisational objectives with firms’ human talent and develop human talent for future objectives.

Labour relationship: An expert system in this area can help organisation to negotiate between management and employee unions about the labour laws and issues pertaining to lock-outs and strikes.

Motivation and leadership: An expert system in this area can help organisation by identifying motivational strategies to be implemented to motivate employees. For
Design of expert system prototype for analysing and structuring motivational strategies on ICT human resources to reduce employee turnover ratio

Viral Nagori
Computer Science
Registration Number – 0980150

each employee, a customized set of motivational strategies can be implemented based on his demographic factors. Based on motivational level and employee skills, an expert system can also help in identifying future leaders for the organisation.

**Career planning:** An expert system in this area can help students in identifying the potential careers based on their skills. It can take students’ psychometric test and based on the analysis, it will provide them with career counselling to the students. The expert system in this area can also be extended to identifying whether the person is allocated for the right job or not and which senior level employee will be the best mentor for the junior level employee.

### 1.12 Types of expert systems

The expert system is a very special branch of Artificial intelligence that makes extensive use of specialised knowledge to solve problems at the level of human experts. There are different types of expert systems. They are rule based expert system, fuzzy expert system, frame based expert system, and hybrid expert systems. The Hybrid expert system is the combination of two or more types of intelligent systems. Prominently, there are two types of hybrid expert systems. The first one is neural expert systems and the second one is Neuro-fuzzy systems. The neural expert system combines the features of a rule based expert system along with neural network features. While Neuro-fuzzy expert system combines the features of fuzzy logic along with the features of neural network. Here we are going to study and provide analysis of different types of expert systems.

#### 1.12.1 Rule base expert system

As the name suggests, a rule based expert system consists of a set of rules. The rule is an expressive, straightforward and flexible way of expressing knowledge. In a rule based expert system, knowledge is represented as a set of rules. Knowledge is a theoretical or practical understanding of a subject or a domain. (Negnevitsky, 2008) Expert possesses deep knowledge and practical experience over the years which results into expertise. An expert has an ability to code the knowledge in form of rules. Any rule consists of two parts: The IF part, called and antecedent (premise or
Design of expert system prototype for analysing and structuring motivational strategies on ICT human resources to reduce employee turnover ratio

condition) and THEN part, called the consequent (conclusion or action). The basic syntax or rule base is:

IF {antecedent}
   THEN {Consequent}

A rule can have multiple antecedents joined by the keywords AND, OR, or a combination of both. The antecedent of a rule consists of two parts. They are objects and its value. Object and value are linked by an operator. The operator can be mathematically or may be logical.

**Structure of rule based expert system:**

A rule based expert system has five components: The knowledge base, the database, the inference engine, the explanation facility and the user interface. The knowledge base contains the knowledge about the domain. The database has set of facts, which is used to match the against the IF- THEN rules. The inference engine provides reasoning, so that expert system can reach a solution. The explanation facility provides the answer to user about why the particular solution is reached. The user interface enables user to interact with the other components of the expert systems.

The other additional components include the external interface, the developer interface, text editor, book keeping facilities, debugging aids, and run time knowledge acquisition. The external interface allows an expert system to interact with other database and programs. The developer interface allows developers to edit with knowledge base, rules and facts. Text editor provides notepad kind facility to enter inputs. The book keeping facility is provided to monitor the changes made by the knowledge engineer in the knowledge base or inference engine. Debugging aids provide tracing of all rules fired during the program execution. Run time knowledge acquisition facility enables to add knowledge or facts, which are not available in the knowledge base or database.

The figure 1.2 represents the structure of a rule based expert system. (Negnevitsky, 2008)
Design of expert system prototype for analysing and structuring motivational strategies on ICT human resources to reduce employee turnover ratio

Figure 1.2 Architecture of rule base expert system
Inference Techniques:
The rule based expert system has set off IF-THEN rules to represent the domain knowledge and set of facts which will represent the current situation. Inference engine compares each rule stored in knowledge base with the facts in the database. When the IF rule is matched with a fact, the rule is fired and its action part stated in THEN is executed. The fired rule may change the set of facts by adding new facts in a database. The entire above process produces inference chains. The process is shown in the figure 1.3. (Negnevitsky, 2008)

Figure 1.3 Inference chain for rule base expert system
There are two types of inference techniques. Forward chaining and backward chaining.

**Forward Chaining:**
Forward chaining is the data driven reasoning. The reasoning starts from the known data and proceeds. It will look for the rules which will move the current state of problem solution closer to the final solution. When a rule is fired, the new facts are added to the database. (Chakraborty R. C., 2010)

Let us consider the following example.
A rule base consisting of the following rule set.
Rule 1: If A and C then F
Rule 2: If A and E then G
Rule 3: If B then E
Rule 4: If G then D.

**Problem:** Prove that If A and B is true, then D is true.

**Solution:** Here we will start with rule 1 and proceed towards the next rule. First rule 3 will be fired and then rule 2 will be fired and then at the end rule 4 will be fired. Hence we can reach to the desired goal to prove that if A and B then D is true. This is how inference engine will process inference chain in a forward manner.

The only problem with the forward chaining is that if the rule base has thousands rule, then many a times, unnecessary rules are also fired, which will increase the execution time and we will generate many facts which are unrelated to the goal. To overcome this problem, backward chaining is used.

**Backward Chaining:**
Backward chaining is the goal driven reasoning. In backward chaining, an expert system has the goal and the inference engine attempts to find the evidence to prove it. First, the inference engine will search the knowledge base to find rules that have required a solution and such rule will have goal in their action (THEN) parts. If such
a rule is found and if its condition (IF) part matches the data in the database, then the rule is fired and then the goal is proven.

Let us see how backward chaining works from the above example.

A rule base consisting of the following rule set.

Rule 1: If A and C then F
Rule 2: If A and E then G
Rule 3: If B then E
Rule 4: If G then D.

**Problem:** Prove that If A and B is true, then D is true.

**Solution:** Here we will start with the goal D is true and move backwards till a rule is fired. So rule 4 will be fired and we have a new goal to prove that G is true. When G is true, it means both A and E is true as per rule 2. While we are given the fact that A is true, hence we have got a new goal which states that E is true. If E is true, moving backwards, we ascertain that B is true using rule 3. Hence now we have ascertained both A and B is true. Hence the goal is proved.

**Difference between forward chaining and backward chaining:**

The table 1.3 will differentiate between forward chaining and backward chaining.

<table>
<thead>
<tr>
<th></th>
<th>Forward chaining</th>
<th>Backward chaining</th>
</tr>
</thead>
<tbody>
<tr>
<td>The data is known at the beginning of the inference process, and the user is not required to input additional facts.</td>
<td>The goal is set up and the only needed data from the database is used for reasoning. The user is only required to input facts which are not in the database.</td>
<td></td>
</tr>
<tr>
<td>Developers should choose the forward chaining when they need to gather some information first and then wants to infer something from that.</td>
<td>Developers should choose backward chaining when they begin with hypothetical solution and then search for facts to prove it.</td>
<td></td>
</tr>
<tr>
<td>Dendral, an expert system for</td>
<td>MYCIN, an expert system for diagnosis</td>
<td></td>
</tr>
</tbody>
</table>
determining molecular structure of unknown soil uses forward chaining. infectious blood disease uses backward chaining.

In conclusion, we can say that a user can use a combination of forward and backward chaining both depends on the type of domain, its requirement and the shell used for development of an expert system.

**Advantages of rule based expert systems:**

- It provides a representation of knowledge in natural ways of if-then rules.
- It has a uniform structure of rule base. Each rule is an independent piece of knowledge.
- It separates the knowledge base from the inference engine, i.e., knowledge base can be updated without intervening of processing.
- The rule base expert system can perform even with incomplete and uncertain knowledge by associating certainty factors with it.
- It is easy to understand and rules can be self-documented without the help of explicit translation.
- The rule base system can be developed in phase by manner. A quick prototype can be developed with fewer rules and if it achieves the desired result, then new rules can be added to improve the performance and efficiency.

**Disadvantages of rule base expert systems:**

- It lacks hierarchical structure of knowledge representation; hence we cannot understand the logical interdependence of rules.
- It goes through exhaustive search; hence it makes the system very slow and cannot be used for real-time applications.
- It cannot learn from past experience and cannot break the rules in case if an exception occurs.
- It is built on experts past experience, guts and intuitions and trial and error approach.
Domain areas where a rule based expert system are useful:

- When knowledge is diffuse, i.e. there are large numbers of facts, which are more or less independent of each other. For example, we cannot use a rule base in mathematics domain where there are set of interrelated principles. But rule base can be applied to clinical trial domain, where comparatively there are a large number of interdependencies among rules.

- Where knowledge can be easily separated from its use, i.e. when there is no dependency on how to use knowledge. For example, we can use a rule base expert system where we need to decide the ingredients of an item, but we cannot use a rule base expert system where we also need to decide how much of each ingredients, we need to mix up for a tasteful recipe.

1.12.2 Fuzzy expert system

When we want to express expert knowledge that uses vague and unclear words like ‘slightly overburden’, ‘heavily reduced’, ‘moderately difficult’, ‘not so old’, ‘very tall’, we can use fuzzy set theory. Fuzzy logic is not logic that is fuzzy, but the logic that is used to describe fuzziness. (Negnevitsky, Artificial Intelligence, 2008) Fuzzy logic describes vagueness. But then the problem comes, how you will differentiate between members of a class from non-members. Fuzzy logic is based on the idea that all things are described on a sliding scale, which helps us to differentiate between members of the class from non-members.

Fuzzy logic is also known as multi valued logic. The classical logic operates on two truth values. They are True (1) and False (0). In fuzzy logic, all the truth values are expressed in real numbers from the interval between 0 to 1. A number in the interval is used to represent the possibility that a given statement is true or false. The figure 1.4 explains the above logic.
Range of logical values in Boolean and fuzzy logic

(a) Boolean Logic.       (b) Multi-valued Logic.

Figure 1.4 Difference between fuzzy logic and Boolean logic

Fuzzy logic is determined for knowledge representation based on degree of membership rather than on crisp membership of classical binary logic. (Zadeh, 1965)

Fuzzy sets:
Fuzzy sets can be simply defined as a set with fuzzy boundaries. A fuzzy set is capable of providing a graceful transition across a boundary.

Let $x$ can be universe of discourse and its elements be denoted as $x$. In classical set theory, crisp set $A$ of $X$ defined as function $f_A(x)$ called the characteristic function of $A$.

$$f_A(x) : X \rightarrow \{0, 1\},$$
Where

$$f_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases}$$

The set map inverse $X$ is a set of two elements. For any element $x$ of universe of $X$, characteristic function $f_A(x)$ is equal to 1, if $x$ is an element of set $A$ and is equal to 0 if $x$ is not an element of $A$.

In fuzzy set $A$ of universe $X$ is defined by function $\mu_A(x)$ called the membership function of set $a$. 
Design of expert system prototype for analysing and structuring motivational strategies on ICT human resources to reduce employee turnover ratio

\( \mu_A(x): \ X \rightarrow [0,1], \)

Where

\( \mu_A(x) = 1 \) if \( x \) is totally in \( A; \)
\( \mu_A(x) = 0 \) if \( x \) is not in \( A; \)
\( 0 < \mu_A(x) < 1 \) if \( x \) is partly in \( A. \)

This allows a continuum of possible choices. For any element \( x \) of universe \( X, \) membership function \( \mu_A(x) \) equals to the degree to which \( x \) is an element of set \( A. \) This degree, a value between 0 and 1 represents the degree of membership, also called membership value, of element \( x \) in set \( A. \) (Negnevitsky, 2008) For example, the fuzzy set young can be represented in the figure 1.5

<table>
<thead>
<tr>
<th>Fuzzy Term young</th>
<th>Age</th>
<th>Grade of Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>25</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Figure 1.5 Representation of fuzzy set

Linguistic variables and hedges:
A linguistic variable is a fuzzy variable. When we say, \( X \) is tall, so here \( X \) takes a linguistic value tall.
For example,
IF the house is old
THEN resale value is low.
IF speed is fast
THEN time to travel is less.

The range of possible values of a linguistic variable represents the universe of discourse of that variable. For variable speed, the values can be very slow, slow, medium, fast and very fast. A linguistic variable carries with it the concept of fuzzy set qualifiers, called hedges. Hedges are terms that modify the shape of fuzzy sets. They include adverbs like very, somewhat, quite, more or less and slightly.

Hedge can modify verbs, adjective, adverbs or even whole sentences. They are used as

- All-purpose modifiers, such as very, quite or extremely.
- Truth-values, such as quite true or mostly false.
- Probabilities, such as likely or not very likely.
- Quantifiers, such as most, several or few.
- Possibilities such as almost impossible or quite possible.

**Fuzzy rules:**
A fuzzy rule can be defined as a conditional statement in the form:
IF x is A

THEN y is B

Where x and y are linguistic variable and A and b are Linguistic Values determined by fuzzy sets on the universe of discourse X and Y, respectively. (Negnevitsky, 2008)

**Difference between classical and fuzzy rules:**
A classical IF THEN rule uses binary logic, for example,

Rule: 1

IF speed is >100

THEN stopping distance is longer

Rule: 2

IF speed is < 40

THEN stopping distance is short
A IF-THEN rule using fuzzy logic,

Rule 1:
 IF speed is fast
 THEN stopping distance is longer

Rule 2:
 IF speed is slow
 THEN stopping distance is short

The most important advantage with fuzzy expert system is that it merges the rules and consequently cut the number of rules significantly.

Fuzzy set includes two distinct parts: evaluating the rule antecedent (the IF part of the rule) and implication or applying the result of the consequent (Then part of the rule). In fuzzy systems, where the antecedent is a fuzzy statement, all rules fire to some extent, or in other words they fire partially. If the antecedent is true to some degree of membership, then the consequent is also true to that same degree.

**Fuzzy inference:**

Fuzzy inference can be defined as a process of mapping from a given input to an output, using the theory of fuzzy sets. (Negnevitsky, 2008) The fuzzy inference process is performed in four steps. They are fuzzification of the input variable, rule evaluation, aggregation of the rule output and finally defuzzification. (Assilian, 1975)

**Step 1: Fuzzification**

In this step, we take the crisp inputs, x1 and y1 and determine the degree to which these inputs belong to each of the appropriate fuzzy sets.

**Step 2: Rule evaluation**

In this step, we take fuzzified inputs and apply them to the antecedents.
Design of expert system prototype for analysing and structuring motivational strategies on ICT human resources to reduce employee turnover ratio

**Step 3: Aggregation of the rule outputs**
In this step, we take the membership functions of all rule consequents previously clipped or scaled and combine them into a single fuzzy set. Aggregation is the process of unification of the outputs of all rules. Thus the input of the aggregation process is the list of clipped or scaled consequent membership functions, and the output is one fuzzy set for each output variable.

**Step 4: Defuzification**
Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number. The above process of converting the output to crisp number is called defuzzification. The input for the defuzzification process is the aggregate output fuzzy set and it is a single number.

**Fuzzy expert system development process:**
The process has the following steps.
1. Specify the problem and define linguistic variables.
2. Determine fuzzy sets.
3. Elicit and construct fuzzy rules.
4. Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system.
5. Evaluate and tune the system.

**1.12.3 Frame based expert system**
In rule base expert system and fuzzy expert system, IF- THEN rules are used to represent knowledge. In frame based expert system, frames are used to represent knowledge. Now we will try to explore the concept of frames.

**Frame:**
A frame is a data structure with typical knowledge about a particular object or concept. Frames are used to capture and represent knowledge in a frame based expert system. (Minsky, 1975) Each frame has its own name and set of attributes or slots associated with it.
The frame provides a natural way for the structured and concise representation of knowledge. We can combine all necessary knowledge about a particular object or concept in a single entity. In general frames are an application of object-oriented programming for expert systems. The advantage what frames offer over the rule base is that we just need to search through frames only to execute rules, unlike of the rule base expert system, where it goes through a systematic search through all rules for execution.

Frame as a knowledge representation technique:
The concept of a frame is defined by a collection of slots. Each slot describes a particular attribute or operation of the frame. Slots are used to store values. A slot may contain a default value or a pointer to another frame, a set of rules or procedure by which the slot value is obtained. (Negnevitsky, 2008)

In general, a slot may include the following information:

1) Frame name
2) Relationship of the frame to the other frames.
3) Slot value
4) Default slot value
5) Range of slot value
6) Procedural information: A procedure is executed if the slot value is changed or needed. There are two types of procedures attached to slots.
   a. When changed procedure is executed when new information is placed in the slot.
   b. When the needed procedure is executed when information is needed for the problem solving, but the slot value is unspecified. Such procedural attachments are called demons.

Frame based expert systems also provide an extension to the slot value structure through the application of facets.
Facet:
A facet is a means of providing extended knowledge about an attribute of a frame. Facets are used to establish the attribute value, end-user queries, and tell the inference engine how to process the attribute.

There are three kinds of facets which can be attached to the frame based expert system. They are value facets, prompt facets and inference facets. Value facet specifies default and initial values of an attribute. Prompt facets enable the end-user to enter the attribute value on-line during a session with the expert system. And finally inference facet allows us to stop the inference process when the values of a specified attribute changes.

Methods and demons:
Frames provide us structured way of representing knowledge. But what should we do when we want to validate and manipulate the knowledge. The answer to that lies in the methods and demons.

Method is a procedure associated with a frame attribute that is executed whenever requested. (Durkin, 1994) Method is represented by a series of commands similar to a macro in Microsoft excel.

In general, demons have an IF-THEN structure. It is executed whenever an attribute in the demon’s IF statement changes its value.

Inference engine:
In a frame based expert system, the inference engine searches for the goal or a specific attribute. In a frame based expert system, rule plays a secondary role. Knowledge is stored in frames and both methods and demons are used to add action to the frames. (Negnevitsky, 2008) The inference engine finds those rules whose consequents contain the goal of interest and examine them one by one in order of rule base. If all the rules are valid, then inference engine will conclude that goal is reached else if any of the antecedents are invalid, then it is concluded that goal is not reached.
Frame based expert system development process:
The steps in the frame based expert system development process are as follows: (Negnevitsky, 2008)

1) Specify the problem and define the scope of the system.
2) Determine the classes and their attributes.
3) Define instances.
4) Design displays.
5) Define when changed and when needed methods, and demons.
6) Define rules.
7) Evaluate and expand the system.

1.12.4 Hybrid expert system
The above discuss technologies, i.e. rule based expert system, fuzzy expert system, and frame based expert system have positive and negative points. Let say, in rule based expert system, the time to search and execute rule is longer, while Fuzzy logic allows you deal with imprecise knowledge. The frame based expert system allows us to represent knowledge in a hierarchical way, while in rule based expert system is most easy to construct because of it simple structure. When we want to combine the advantages of two components, we can create hybrid technology. But the hybrid technology can be good or bad depending on which two technologies are combined. Here we are exploring two hybrid expert systems.

i. Neural expert systems

ii. Neuro-fuzzy systems

It is important to note that, in both of hybrid system; neural network is one of the important components.

1.12.4.1 Neural expert system
Neural network and expert systems both have a common goal of imitating the human intelligence. A hybrid expert system, which combines a neural network and a rule based expert system, is called a neural expert system (Connectionist expert system). Here with the obvious advantages of a rule based expert system, we also combine the
advantages of neural network like learning, generalisation, robustness and parallel information processing.

**Structure of neural expert system:**

The figure 1.6 will show the basic structure of a neural expert system. (Negnevitsky, 2008)
Figure 1.6 Structure of neural expert system

A Neural expert system has a neural knowledge base in place of traditional knowledge base, where knowledge is stored as weights in neurons. The combination of rule extraction component along with neural network enables neural expert system to justify and provide explanation facility for its conclusion.

Neural network also allows dealing with noisy and incomplete data because of its capability of generalisation. Hence it allows approximate reasoning. The rule extraction unit examines the neural knowledge base and produces the rules implicitly buried in the trained neural network. The user interface provides the interaction between the user and the neural expert system.

Rule extraction in neural expert system:

Neurons in the network are connected by links, each of which has a numerical weight attached to it. The weights in a trained neural network determine the strength or importance of the associated neuron inputs. There are three layers in neural networks. The first layer is input layer. Neurons in the input layer transmit external signals to the next layer. The middle layer is the conjunction layer. It is also known as hidden layer. Here activation function is used to calculate weights for neurons. There are different types of activation functions. The number of hidden layers depends on the types of input, type of domain and type of problem to be solved. The last layer is the output layer. Each output neuron receives an input from a single conjunction neuron. The weights between the second and the third layers are set to unity.

IF-THEN rules are mapped into multi-layer neural network where the last layer represents the action parts of the rules. We can train a neural network according to a given set of training data using back propagation training algorithm. Once the initial training phase is completed, we can examine the neural network knowledge base, extract and refine the set of initial IF-THEN rules. Hence a neural expert system provides a bi-directional link between neural networks and rule based systems.
The only problem with neural expert system is that it cannot represent continuous input variable as it may lead to the creation and execution of infinite rules. The problem can be overcome by Neuro-fuzzy systems.

**1.12.4.2 Neuro-fuzzy system**

Fuzzy logic and neural networks are natural complementary tools in building intelligent systems. While neural networks are low-level computational structure that performs well when dealing with raw data, fuzzy logic deals with reasoning on a higher level, using linguistic information acquired from domain experts. (Negnevitsky, 2008)

However, fuzzy systems lack the ability to learn and cannot adjust themselves to a new environment. On the other hand, although neural networks can learn, they are opaque to the user, i.e. the process is a black box for the end users.

But when you merge a neural network with fuzzy expert system, then it provides you more powerful technology to design expert system. Integrated Neuro-fuzzy systems can combine the parallel computation and learning abilities of neural network with the human-like knowledge representation and explanation abilities of fuzzy systems. As a result, neural networks become more transparent, while fuzzy system becomes capable of learning.

A Neuro-fuzzy system is, in fact, a neural network that is functionally equivalent to a fuzzy inference model. It can be trained to develop IF-THEN fuzzy rules and determine membership functions for input and output variables of the system. At the same time, the connectionist structure avoids fuzzy inference, which entails a substantial computational burden.

**Structure of Neuro-fuzzy expert system:**

The general structure of Neuro-fuzzy expert system is given below in the figure 1.7. Here it has input and output layers and three hidden layers that represent membership functions and fuzzy rules.
Design of expert system prototype for analysing and structuring motivational strategies on ICT human resources to reduce employee turnover ratio

Figure 1.7 Structure of Neuro-fuzzy system
Neuro-Fuzzy Inference process:
Each layer in the Neuro-fuzzy system is associated with a particular step in the fuzzy inference process. (Negnevitsky, 2008)

Layer 1 is the input layer. Each neuron in this layer transmits external crisp signals directly to the next layer. That is,

$$y_i^{(1)} = x_i^{(1)}$$

Where $X_i$ is input and $Y_i$ is the output of input neuron $i$ in Layer 1.

Layer 2 is the fuzzification layer. Neurons in this layer represent fuzzy sets used in the antecedents of fuzzy rules. A fuzzification neuron receives a crisp input and determines the degree to which this input belongs to the neuron’s fuzzy set. The activation function of a membership neuron is set to the function that specifies the neuron’s fuzzy set.

Layer 3 is the fuzzy rule layer. Each neuron in this layer corresponds to a single fuzzy rule. A fuzzy rule neuron receives inputs from the fuzzification neurons that represent Fuzzy sets in the rule antecedents. For instance, neuron $R_1$, which corresponds to Rule 1, receives inputs from neurons $A_1$ and $B_1$.

The weights between layer 3 and layer 4 represent the normalised degrees of confidence of the corresponding fuzzy rules. These weights are adjusted during training of a Neuro-fuzzy system.

Layer 4 is the output membership layer. Neurons in this layer represent fuzzy sets used in the consequent of fuzzy rules. An output membership neuron combines all its inputs by using the fuzzy operation union.

Layer 5 is the defuzzification layer. Each neuron in this layer represents a single output of the Neuro-fuzzy system. It takes the output fuzzy sets clipped by the respective integrated firing strength and combines them into a single fuzzy. The
output of the Neuro-fuzzy system is crisp and combined output fuzzy set must be defuzzified. For that standard defuzzification method can be applied.

**Learning of Neuro-fuzzy system:**

Neuro-fuzzy system can learn using standard learning algorithms developed for neural networks for example, back propagation. (Altock, 1997)

When a training input-output of example is presented to the system, the back – propagation algorithm computes the system output and compares it with the desired output and compares it with the desired output of the training example. The difference is propagated backwards through the network from the output layer to the input layer.

The neuron activation functions are modified as the error is propagated. To determine the necessary modifications, the back-propagation algorithm differentiates the activation function of the neurons.

This is how Neuro-fuzzy system learns.

**1.12.5 Comparison of different types of expert systems**

Here we have discussed in total five expert systems. They are rule based expert system, fuzzy expert system, frame based expert system, neural expert system, and Neuro-fuzzy expert system. The parameters which we are using for comparison are: Knowledge representation, learning ability, uncertainty tolerance, imprecision tolerance, explanation facility, inference engine, knowledge update facility, maintenance, adaptability, knowledge structure and processing time of the system. The table 1.4 presents comparison.
Table 1.4 Comparison between different types of expert system

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Rule based expert system</th>
<th>Fuzzy expert system</th>
<th>Frame based expert system</th>
<th>Neural expert system</th>
<th>Neuro-fuzzy expert system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge Representation</td>
<td>IF-THEN rules in the knowledge base</td>
<td>Based on degree of membership using fuzzy logic</td>
<td>Knowledge in frames using a hierarchical structure</td>
<td>IF-THEN rules on the neural knowledge base</td>
<td>In linguistic variable and using IF-THEN rules in fuzzy structure</td>
</tr>
<tr>
<td>Learning ability</td>
<td>Cannot learn on its own, and update the existing knowledge base</td>
<td>Lacks ability to learn from the experience</td>
<td>Cannot learn and adjust to the new environment</td>
<td>Neural network can learn, but the learning is black box process for the user</td>
<td>Has learning ability because of neural network as one of the component</td>
</tr>
<tr>
<td>Uncertainty tolerance</td>
<td>Difficult to measure uncertainty</td>
<td>Probabilistic reasoning can deal with uncertainty</td>
<td>Not possible because of knowledge structure</td>
<td>Approximate reasoning</td>
<td>Using script value and probabilistic reasoning</td>
</tr>
<tr>
<td>Imprecision tolerance</td>
<td>Very low, required precise information</td>
<td>High, as fuzzy logic can deal with imprecision</td>
<td>Very low, imprecise data can lead to faulty output</td>
<td>Neural network component can deal with imprecise data</td>
<td>Very high because of combination of neural network and fuzzy logic</td>
</tr>
<tr>
<td>Explanation facility</td>
<td>Yes</td>
<td>Yes, using linguistic variable</td>
<td>Yes, good explanation for the output</td>
<td>Yes, because of rule based component</td>
<td>Yes, very effective</td>
</tr>
<tr>
<td>Inference Engine</td>
<td>Process rules and derive conclusion</td>
<td>Process rules using fuzzification and defuzzification</td>
<td>Search for goals using methods and demons</td>
<td>Three layer neural network combined with rule extraction</td>
<td>Fuzzy inference process using fuzzification and defuzzification</td>
</tr>
<tr>
<td>Knowledge update</td>
<td>Difficult to add new rules</td>
<td>Difficult to introduce new linguistic variables in existing structures</td>
<td>Not possible</td>
<td>Yes, With new knowledge, neural network component can learn</td>
<td>Yes, new linguistic variables can be added into existing knowledge structures</td>
</tr>
</tbody>
</table>
Design of expert system prototype for analysing and structuring motivational strategies on ICT human resources to reduce employee turnover ratio

<table>
<thead>
<tr>
<th>Maintainability</th>
<th>Moderately difficult</th>
<th>Very difficult</th>
<th>Easy</th>
<th>Easy</th>
<th>Difficult because of fuzzy component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptability</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Processing time</td>
<td>Very high due to each rule is processed</td>
<td>Processing time is reduced compared to rule based expert system</td>
<td>The knowledge stored in the frames can be processed very rapidly.</td>
<td>Learning for neural network takes time, but once knowledge is stored in neurons, then processing time for rule is fast.</td>
<td>Learning for neural network takes time, but then processing time is significantly reduced.</td>
</tr>
<tr>
<td>Knowledge Structure</td>
<td>Adhoc, cannot understand logical dependence of rules</td>
<td>Quite unstructured</td>
<td>Highly structured</td>
<td>Structured, but stored in neurons</td>
<td>Moderately structured</td>
</tr>
</tbody>
</table>

In conclusion, we can say that each type of expert system has its own merits, demerits and requirement to use based on domain analysis. Hence forth, we will explore the development environment and tools for development of an expert system. The development environment consist of programming languages, expert system tools and expert system shells.

1.13 Programming languages for expert system

There are basically two types of programming languages. They are algorithmic languages and symbolic languages. ‘Pascal’, ‘C’, ‘Basic’ and ‘Fortran’ are algorithmic languages. They are also known as procedural languages. But implementing logical inferences is difficult in algorithmic languages. Hence we need a specialised language which can implement logical inferences. Such languages are known as symbolic languages. The most common symbolic languages, which are used in the development of expert systems, are LISP and PROLOG.
1.13.1 LISP (LISt Processing)
John McCarthy developed LISP in 1950, which stands of LISt processor. It supports symbolic manipulation and interactive trial and error style of programming. The most prolific advantage of LISP is that it has a set of primitive operator for carrying out the deduction with sentences containing words representing predicates and their arguments and thus helps in implementing logical inferences. (Akerkar, 2007)

The major characteristic of LIPS is that all basic elements are treated as symbols irrespective of whether they are numeric or alphanumeric. (V S Janakiraman K. S., 2005) The basic data element of LISP is an atom, a single word or a number that stands of an object or value. An atom has a very special property. It is indivisible. LISP provides two basic types of atoms: numbers and symbols. Number stores numeric values. Numeric values can be positive or negative integers or floats. Symbol represents alphanumeric characters which represent alphabets and numeric.

LISP provides three types of basic functions. They are arithmetic, Boolean and list manipulation primitives. The developer can define its own functions as well. List manipulation primitives are basically used for creating a new list, modifying and existing list with new atom or extracting portions of a list. (V S Janakiraman K. S., 2005)

Another specialty of LISP is that it provides a facility for dynamic memory management by which we can reclaim used memory. The above facility is also known as garbage collection.

1.13.2 PROLOG
Alain Colmerauer and his colleagues in Marseilles developed PROLOG in 1970. PROLOG is a higher level language than LISP. The current standard used today is Edinburgh PROLOG and Turbo PROLOG. PROLOG is very useful for declarative programming where we declare a set of statements or axioms about the system that we wish to reason and then it deduces the desired additional facts with built in powers of deduction. (Akerkar, 2007) The programmer has to just specify what is
required rather than indicate how it is to be computed, hence freeing programmers from worrying about the implementation details.

PROLOG has the following characteristics. (V S Janakiraman K. S., 2005)

- The syntax and semantics are very close to formal logic structure.
- It has built in inference engine and automatic backtracking facility, which helps in implementing various search strategies.
- It is easy to maintain program written in PROLOG.
- In entire program, implementation dependencies are uniform.
- Program in PROLOG can be implemented on parallel machines.
- PROLOG as a programming language is very easy to understand as it has procedural and declarative programming capacity.
- In PROLOG, modular programming and testing is possible.
- In PROLOG, we can implement complex data structure.
- PROLOG supports quick prototype and incremental system development.
- Program tracing and debugging is easier in PROLOG.

The basis of PROLOG language is Horn clause and Robinson’s resolution. Horn clause consists of a set of statements joined by logical ANDs. The principle of Robinson’s resolution is that two clauses can be resolved with one another if one of them contains a positive literal and other contains a corresponding negative literal with same predicated symbols and the same numbers of arguments. (V S Janakiraman K. S., 2005) List is an important data structure in PROLOG. It is a collection of an ordered sequence of terms.

A PROLOG program consists of a set of clauses. A clause is either a fact of a rule, which is used to indicate a relationship between elements. PROLOG tries to match the arguments of the query with the facts in the database. This process is known as unification. If the unification succeeds, the variable is said to be instantiated. Here a description of domain is known as database. Variables in PROLOG are used to represent unspecified domain elements.
The orders of database entries are very important because the efficiency of a program depends on order as it can affect the amount of search required to satisfy the goal. Secondly, because of the exact nature of the search process, there might be cases where PROLOG may not find a solution even when it can be easily inferred from the given information.

The inference process for PROLOG programing is as follows. (Rolston, 1988)

1. Given a goal, PROLOG searches the database, starting at the top, for a fact that matches the goal.
2. When PROLOG finds a match and instantiates the appropriate variables, it leaves a pointer where the match occurred.
3. When a goal matches the head of a rule rather than a fact, the atoms within the body of the rule are treated as sub goals that must all be satisfied to prove that the head is satisfied.

1.14 Expert system tools

The main task of expert system development tools is making the development of an expert system much easier compared to programming language. Developers say that selecting the correct tool is a vital in the design and development of an expert system. The reasons are manifold. They are: (Rolston, 1988)

- They provide a rich software development environment like structure editors, powerful debugging and tracing facility, multi windows, graphics etc.
- Allows rapid prototyping because of incremental compilers, and automatic version control.
- Defining model, knowledge representation and inference design are built into the tools.
- Helps in maintaining the system and historical database.

In general, Expert system tools are ranging from low level general purpose programming languages to highest level hybrid development environments. There are many expert system development tools are available. Here we are trying to explore some of them in different categories.
1.14.1 Types of expert system tools

In general, there are five types of expert system tools.

1. Inductive tools
2. Simple rule based tools
3. Structured rule based tools
4. Hybrid tools
5. Domain specific tools

**Inductive tools:** Inductive tools generate rules from examples. Here a developer feeds in a large number of examples from the machine's information base. The tools use an algorithm to convert the examples into rules and determine the order the system will follow when questioning the user.

**Simple rule based tools:** They use IF-THEN rules to represent knowledge. They are useful for developing expert systems containing fewer than 500 rules. The only problem with these tools is that they lack high end editing facility for the design of tools.

**Structured rule based tools:** They offer context trees, multiple instantiation, confidence factors, and more powerful editors compared to simple rule based tools. Here IF-THEN rules are arranged into sets. These rules sets act as separate knowledge bases. One set of rules can inherit the information acquired when other rule sets are examined. These tools are most useful when we need to process large number of rules and rules can be subdivided into sets.

**Hybrid tools:** Hybrid tools enable complex expert system development. These tools use object oriented programming techniques to represent elements of every problem as objects. Here a graphical user interface can also be provided to users.

**Domain specific tools:** They are specially designed to be used only to develop expert systems for a particular domain. It provides special development and user interface that make it possible to develop an expert system faster. They are also referred as narrow tools.
1.14.2 Selection criteria for expert system tools

The following criteria will help us to select the right kind of tool for the design and development of expert systems.

- Type of knowledge representation
- Inference, and control
- Developer interface
- User interface
- System interface
- Training and support

The answers to the following questions will help us to reach to selection of a right tool for the design and development of expert systems. (V S Janakiraman K. S., 2005)

✓ Does the tool portable on variety of hardware platforms?
✓ What is the cost of the license?
✓ How frequently does the new version come, and with what enhancement?
✓ What types of applications are developed and deployed using these tools?
✓ What other resources are required for implementation along with tools?
✓ Does the tool provide the knowledge representation mechanism suited for the problem domain?
✓ Does the tool have the flexibility to adopt to different knowledge representation techniques?
✓ Does the tool provide a natural language interface for end users?
✓ What is the time taken to process rules, facts and execute it to display results?
✓ Does the tool provide user friendly interface for accepting input and displaying results?
✓ Does the tool provide an interface with database and spreadsheets?

There are many expert system development tools available in the market. Few of them are CLIPS (C Language Integrated Production system), Loops, KEE, AGE, ART, ESIP Advisor, EXSYS, Guru, Insight, Rule Master, VP- Expert and SRL. In recent times, MATLAB and SciLab are two tools, which are used widely all over the
world. MATLAB is a proprietary tool, while SciLab is an open source tool. But both enable developers to design and develop complex and hybrid expert system.

1.14.3 MATLAB

Matrix Laboratory is a computing environment. It is the 4th Generation programming language. The MATLAB is developed by Mathworks. The Current version of Matlab is R2013b released on 5th September 2013. MATLAB provides the solution for the following types of problems. (Matlab)

- Data acquisition
- Data analysis
- Mathematical Modeling
- Algorithm Development
- Parallel Computing
- Desktop and Web Deployment
- Machine Learning
- System Design and Simulation
- Physical Modeling
- Discrete-Event Simulation
- Rapid Prototyping
- Embedded Code Generation
- HDL Code Generation and Verification
- Verification, Validation, and Test
- Embedded Systems
- Control Systems
- Digital Signal Processing
- Communications Systems
- Image Processing and Computer Vision
- FPGA Design and Codesign
- Mechatronics
- Test and Measurement
- Computational Biology
• Computational Finance
• Robotics

The most obvious advantage of MATLAB is that it provides the interface to combine the programs written in other languages like C, C++, JAVA and FORTRAN.

1.14.4 SCILAB
Scilab is an open source alternative for MATLAB. Because of its cross platform numerical computation abilities, it is equivalent to MATLAB for usage. It is used in engineering and scientific problem solving. Scilab 5.4.1 is the latest version, released on 2nd April 2013. Scilab provides the solution for the following kind of problems. (About Scilab)
• Signal processing
• Statistical Analysis
• Image processing and enhancement
• Fluid dynamics simulations
• Numerical optimization and modeling
• Simulation of dynamic systems
• Symbolic manipulations

1.14.5 Comparison between MATLAB and Scilab
The most common similarity between MATLAB and Scilab is that they both operate on matrices. But apart from that both have many differences in the context of functions, variables, comments, strings, Boolean etc.

Scilab is free and open source, which can be easily downloadable. On the other hand, MATLAB being proprietary software, it is very expensive. But the evaluation copy of MATLAB can be obtained through request with limited functionalities.

The user interface of MATLAB is very much user friendly compared to Scilab. As far as computation speed is concerned, MATLAB has an advantage over Scilab.
For image processing solution, if computer memory is not the constraint, a user can choose MATLAB over Scilab as the matrices calculation time is much lower in MATLAB compared to Scilab. But if the problem is about signal processing, Scilab has an advantage, because it occupies lesser memory to generate and process the signals.

As far as a solution to the problems of artificial neural network is concerned, both MATLAB and Scilab provides Neural Network toolbox. A tool box provides functions to model the complex non-linear systems. The data fitting, clustering, data mining, time series prediction, and dynamic system modelling can be done using neural network toolbox.

The table 1.5 summarises the comparison between MATLAB and Scilab.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>MATLAB</th>
<th>Scilab</th>
</tr>
</thead>
<tbody>
<tr>
<td>License</td>
<td>Proprietary</td>
<td>Open source</td>
</tr>
<tr>
<td>Cost</td>
<td>Very high</td>
<td>Free</td>
</tr>
<tr>
<td>Plug-ins</td>
<td>Available</td>
<td>Not Available</td>
</tr>
<tr>
<td>User interface</td>
<td>User friendly</td>
<td>Not so user friendly</td>
</tr>
<tr>
<td>Memory Requirement</td>
<td>Very High</td>
<td>Not so high</td>
</tr>
<tr>
<td>Computation speed</td>
<td>Very high</td>
<td>Less compared to MATLAB</td>
</tr>
<tr>
<td>Software updates</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>Functions and toolbox</td>
<td>Many</td>
<td>Less compared to MATLAB</td>
</tr>
<tr>
<td>Tutorials and demos</td>
<td>Easily available</td>
<td>Not so easily available</td>
</tr>
</tbody>
</table>

In conclusion, we can say that the choice of tool depends on the requirement of the domain as well as computation resources and cost constraints. If licensing cost is not an issue, go for MATLAB. But if you want free software, Scilab is the better option. If you have high computing resources with no memory constraints, then go for
MATLAB. But if you are working with shared memory and less computing resources, Scilab is the better option.

1.15 Expert system shell

A shell is a piece of software which helps in designing the inference engine, a knowledge base and user interface for expert systems. It provides a user friendly software environment to the knowledge engineer for building an expert system. It contains all the generic expert system logic required to build an expert system. (Padhy, 2005)

A shell may be thought of as an expert system with all domain specific knowledge removed and a facility for entering a new knowledge base provided. (Akerkar, 2007)

In general, the shell provides the following facilities.

- A knowledge representation language
- Knowledge base editor
- Tracing and debugging facilities
- User interface facilities
- Facilities for uncertain reasoning
- Rule induction facilities

1.15.1 Advantages of expert system shell

The following are the advantages of using an expert system shell.

- The programming efforts that have gone into building the user interface and inference engine is reused.
- The level of programming skills needed to produce the finished system is much lower compared to developing a system from scratch using conventional programming methods.
- The expert system development project can be completed faster, cheaper and in a more efficient way.

1.15.2 Disadvantages of expert system shell

The following are the disadvantages of using an expert system shell.
• Expert system shells are end user tools. So resulting software package may have poor documentation, weak security and high maintenance.

• If the shell is a poor match for the type of knowledge in the domain concerned, it may not replicate the expertise of an expert.

1.15.3 Selection criteria for expert system shell

The following criteria will help us to decide the choice of the expert system shell.

• The characteristics of knowledge, and the style of inferences used by domain experts.

• The time and money available for the project.

• The programming capabilities available in house.

• The hardware platform on which the system is to be deployed.

• The performance expectation from the system.

1.15.4 Examples of expert system shell

The following are the examples of expert system shells.

• Ex sys

• Knowledge Pro

• K-Vision

• Age

• Emycin

• KAS

• Leonardo

• Xi Plus

• Savoir

• XpertRule

Here we are exploring two most famous expert system development shell: Vidwan and JESS (Java Expert system shell)

1.15.5 Vidwan

Vidwan is an expert system shell developed at the National Centre for Software Technology, Mumbai in 1993. It enables knowledge of a domain to be encoded in
the form of IF-THEN rules. It supports backward chaining as an inference procedure. It also allows users to associate uncertainty associated with domain knowledge using certainty factors. It also provides explanation facility in the form of questions why and how. It also provides an interactive text editor for creating rule base. (M Sasikumar, 2007) The most important characteristic of Vidwan is that it is easy to learn to use Vidwan.

Vidwan is a domain independent, customizable, multi-user, web based expert system shell which allows creation of the rule base expert system. The user can interact with Vidwan at two levels. They are rule base user and rule base owner.

**Rule base user:**
Here the user will seek advice from Vidwan for a particular domain where rule base is already developed. Vidwan allows user an interaction with rule base. At the end of interaction, user can store results and then continue interactions in the next session.

**Rule base owner:**
Here the user will act as a domain expert and will impart his expertise to the rule base. Here user can add, update and delete from and into the existing rule base. It also allows domain experts to add comments into multiple templates for a given rule base.

So far, varieties of applications have been developed including medical advisors, troubleshooting systems and financial advisors. The basic components of a rule base for Vidwan are the parameters of interest in domain are called attributes of the domain. Vidwan also allows us to produce different reports. Vidwan allows user to record user interaction in a file to obtain a log of the session. Vidwan also provides limited arithmetic functionalities along with user friendly interface with windows, cursor selection of commands, and online help.

Vidwan also provides a feature called ‘revoke’, where user can review the responses he had given so far and if he feels, he can clear the values of attributes. Vidwan also provides support for database access, graphics and automatic data acquisition.
Vidwan applications can be embedded in programs developed in other languages. (M Sasikumar, 2007)

1.15.6 JESS (JAVA Expert System Shell)

JESS was developed at Sandia National Laboratories in late 1990s by Dr. Ernest J Friendman-Hill. The concept of JESS was inspired by AI production rule language CLIPS. It provides fully developed Java API for creating rule base expert system. Its syntax is similar to LISP. It can also be used to access JavaBeans. JESS is very simple and easy to learn. JESS is small, light and fastest rule engine available.

JESS matches facts in the fact base to rules in the rule base. JESS rules will form a knowledge base of the expert system. JESS uses Rete algorithm to match patterns. The figure 1.8 will represent JESS architecture. (Morris, 2003)

![JESS Architecture](image)

JESS provides features like backward chaining and working memory queries. JESS is a powerful java scripting environment where you can create java objects, java
methods and java interfaces. There are three ways to represent knowledge in JESS: Rules, functions and objects.

Before we conclude, we are discussing the pertaining issues which are raised by the legal practitioners on the use of an expert system.

1.16 Legal and Ethical issues with an expert system

Expert systems have been developed in domains like marketing, medical, pharmacy, Human resources, production, automobiles, engineering and others. In many places, expert systems have been successful in replacing human experts as well. In consideration of the current technology and development in the field of expert systems, the legal practitioners have raised the significant debate.

Who will be responsible if the advice given by expert system turn out to be wrong?

- User?
- Domain expert?
- Knowledge Engineer?
- Developer?
- Company?

Let say expert system has diagnosed the problem of the patient and provided a treatment. If that treatment or diagnosis turns out to be wrong and if patient suffers major loss to its health or if the patient dies, who will be responsible? Will you say a user, who uses expert system at his/her own risk? Or will it be a domain expert, who has provided erroneous knowledge and expertise to the system? Or will it be knowledge engineer, who has translated expert’s expertise into knowledge base? Or will it be a developer, who has introduced a bug in the system or the IT Company, who has taken a project of expert system development?

The IT practitioners and experts are still searching for the answer of the above problem. But to conclude, we can say that, expert system if it is used with care and diligence, then it can be a good alternative of human experts in some cases.
Chapter Conclusion

The chapter described the fundamentals of expert systems and its architecture. As our aim was to develop an expert system in the human resource domain, we carried out a study of different application areas especially the human resource domain. We also explored the different types of an expert system and the tools, languages and shells which help us in choosing the method or tool in the development of our expert system. In short, this chapter has built a platform to understand, design and develop an expert system.

1.17 References


Design of expert system prototype for analysing and structuring motivational strategies on ICT human resources to reduce employee turnover ratio


