UNIT – V (Expert System)

The first successful commercial expert system, R1, began operation at the Digital Equipment Corporation (McDermott, 1982). The Bayesian network formalism was invented to allow efficient representation of, and rigorous reasoning with, uncertain knowledge. This approach largely overcomes many problems of the probabilistic reasoning systems of the 1960s and 1970s; it now dominates AI research on uncertain reasoning and expert systems. The approach allows for learning from experience, and it combines the best of classical AI and neural nets. Work by Judea Pearl (l982a) and by Eric Horvitz and David Heckerman (Horvitz and Heckerman, 1986; Horvitz et al., 1986) promoted the idea of normative expert systems: ones that act rationally according to the laws of decision theory and do not try to imitate the thought steps of human experts. Early discussions of representation in AI tended to focus on "problem representation" rather than "knowledge representation". In the 1970s, AI emphasized the development of "expert systems" (also called "knowledge-based systems") that could, if given the appropriate domain knowledge, match or exceed the performance of human experts on narrowly defined tasks. For example, the first expert system, DENDRAL, interpreted the output of a mass spectrometer (a type of instrument used to analyze the structure of organic chemical compounds) as accurately as expert chemists. Although the success of DENDRAL was instrumental in convincing the AI research community of the importance of knowledge representation, the representational formalisms used in DENDRAL are highly specific to the domain of chemistry. Over time, researchers became interested in standardized knowledge representation formalisms and ontologies that could streamline the process of creating new expert systems. In so doing, they ventured into territory previously explored by philosophers of science and of language. The discipline imposed in AI by the need for one's theories to "work" has led to more rapid and deeper progress than was the case when these problems were the exclusive domain of philosophy.

KNOWLEDGE ACQUISITION: Knowledge acquisition is the process of adding new knowledge to a knowledge base and refining or otherwise improving knowledge that was previously acquired. Acquisition is usually associated with some purpose such as expanding the capabilities of a system or improving its performance at some specified task. It is goal oriented creation and refinement of knowledge. It may consist of facts, rules, concepts, procedures, heuristics, formulas, relationships, statistics or other useful information. Sources of this knowledge may include one or more of the following.

- Experts in the domain of interest
- Text Books
- Technical papers
- Databases Reports
- The environment

To be effective, the newly acquired knowledge should be integrated with existing knowledge in some meaningful way so that nontrivial inferences can be drawn from the resultant body of knowledge. the knowledge should, of course, be accurate, non redundant, consistent(non contradictory), and fairly complete in the sense that it is possible to reliably reason about many of the important conclusions for which the systems was intended. Types of learning:- Classification or taxonomy of learning types serves as a guide in studying or comparing a differences among them. One can develop learning taxonomies based on the type of knowledge representation used (predicate calculus , rules, frames), the type of knowledge learned (concepts, game playing, problem solving), or by the area of application(medical diagnosis , scheduling , prediction and so on). The classification is intuitively more appealing and is one which has become popular among machine learning researchers . it is independent of the knowledge domain and the representation scheme is used. It is based on the type of inference strategy employed or the methods used in the learning process. The five different learning methods under this taxonomy are: Memorization (rote learning) Direct instruction(by being told) Analogy Induction Deduction Learning by memorization is the simplest form of learning. It requires the least 5 amount of inference and is accomplished by simply copying the knowledge in the same form that it will be used directly into the knowledge base. We use this type of learning when we memorize multiplication tables, for example. A slightly more complex form of learning is by direct instruction. This type of learning requires more understanding and inference than role learning since the knowledge must be transformed into an operational form before being integrated into the knowledge base. We use this type of learning when a teacher presents a number of facts directly to us in a well organized manner. The third type listed, analogical learning, is the process of learning an ew concept or solution through the use of similar known concepts or solutions. We use this type of learning when solving problems on an examination where previously learned examples serve as a guide or when we learn to drive a truck using our knowledge of car driving. We make frequent use of analogical learning. This form of learning requires still more inferring than either of the previous forms, since difficult transformations must be made between the known and unknown situations. This is a kind of application of knowledge in a new situation. The fourth type of learning is also one that is used frequency by humans. It is a powerful form of learning which, like analogical learning, also requires more inferring than the first two methods. This form of learning requires the use of inductive inference, a form of invalid but useful inference. We use inductive learning when wed formulate a general concept after seeing a number of instance or examples of the concept. For example, we learn the concepts of color sweet taste after experiencing the sensation associated with several examples of colored objects or sweet foods. The final type of acquisition is deductive learning. It is accomplished through a sequence of deductive inference steps using known facts. From the known facts, new facts or relationships are logically derived. Deductive learning usually requires more inference than the other methods. The inference method used is, of course , a deductive type, which is a valid from of inference. In addition to the above classification, we will sometimes refer to learning methods as wither methods or knowledge-rich methods. Weak methods are general purpose methods in which little or no initial knowledge is available. These methods are more mechanical than the classical AI knowledge – rich methods. They often rely on a form of heuristics search in the learning process.

The success of knowledge based systems lies in the quality and extent of the knowledge available to the system. Acquiring and validating a large groups of consistent, correlated knowledge is not a trivial problem. This has given the acquisition process an especially important role in the design and implementation of these systems. Consequently, effective acquisition methods have become one of the principal challenges for the AI researches. The goals of this branch of AI are the discovery and development of efficient, cost effective methods of acquisition. Some important progress has recently been made in this area with the development of sophisticated editors and some general concepts related to acquisition and learning.

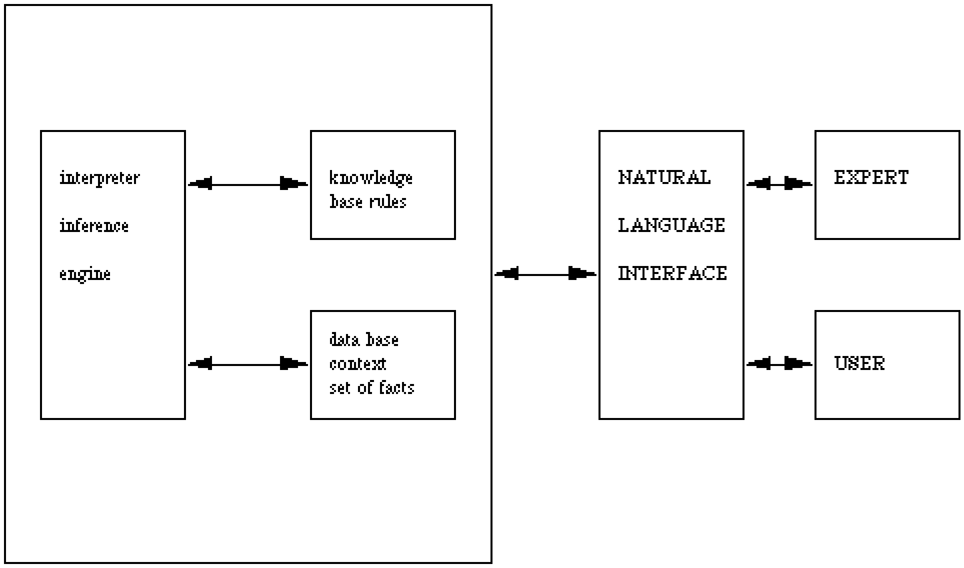
Expert Systems: The Past, Present and Future of Knowledge-based Systems:

Expert Systems were invented as a way to decrease the reliance by corporations on human "experts" -- people who apply reasoning and experience to make judgements in a specific field, such as medicine, insurance underwriting or the operation of a power-plant. Hence, an expert system should include a database of facts and a way of reasoning about them. In many, but not all, applications it is also helpful to have a way for the system to reason with probabilities or non-Boolean truth values. Expert systems are also sometimes referred to a "knowledge-based systems". In classical AI many different reasoning methods have been tried. A few of the common ones are "forward chaining", in which conclusions are drawn from a set of facts by using modus ponens, syllogism, and other simple tools of logic; "backward chaining", which uses trickier logic, such as modus tollens; and neural nets. Most expert systems simply use forward chaining and backward chaining, with some non-Boolean component such as Fuzzy Logic only where necessary. Expert systems tend to be more practical than AI in general. It is quite possible to build an expert system in a conventional programming-language, such as COBOL, C or Java. However, much of the machinery inside an expert system can be abstracted away from any specific domain, and the main criterion in the success of an expert system is its ease of use and maintenance, not it's ability to make decisions in a fraction of a second. Therefore, it is possible to build a "knowledge system shell" which can then be prepared for almost any domain simply by listing rules and data in a standard form. Few expert systems are written in LISP, because most LISP implementations lack robust user-friendly input-output routines. A good knowledge system shell includes I/O routines, a way to accurately and generally represent facts, and an easy, efficient, accurate way to give the system inference-rules.

Early applications the DENDRAL program dealt with the structure of chemical compounds. The MYCIN program " physicians choose appropriate antibiotics for patients with severe bacterial infections.". Another early, commercially successful expert system was PROSPECTOR. It analyzed data that might have to do with the location of oil wells and suggested a most-profitable place to drill. Perhaps these early systems were easy to construct because the principles involved were widely known by academic experts in the relevant fields and available in published literature. One system that was widely mentioned in subsequent papers was an application developed by Digital Equipment Corporation. Mid-1980s By 1985 or so, a significant number of expert systems had been written and deployed, and the industry seems to have had the critical mass to enter a phase of wild growth. These efforts in more specialized domains required the cooperation of one or a few human experts, generally experienced workers close to retirement. This caused public discussion of ethical and labor-relations issues. Used as a training tool, an expert system provides the knowledge of the seasoned professional to the more junior employees. People expressed concern that the experts' knowledge would be drained from them and they would then be out of work; however, it was generally said that the experts were at the point of retirement anyway. Redefining Expert Systems If the early expert-system applications were truly limited by the computer hardware of the time, advances at the end of the 1980s should have made them much more widespread and really opened up the market as had been predicted. A few companies still sell what appear to be traditional expert-systems software and services. One is Acquired Intelligence (Acquired). They sell a framework for building expert systems and also offer consulting-services to build custom ones. Another company is Knowledge System Design, Inc. While they claim to have been in business since 1992, their list of clients is much smaller than that of Acquired Intelligence. They mention a tool for building expert systems on their website, but it is not clear whether they have developed it or simply prefer to use it. Research continues to be done in Expert Systems. For instance, at this university in 2001 a system was developed to organize the manufacturing-processes for composites (Martínez). An example of a something made from a composite is a Fiberglas canoe. There are consumer-visible systems out there that seem to be influenced by expert-system principles. On would be the IBM laptop-selector, mentioned above. Future One fear that has been expressed occasionally in science fiction is that human society will become decadent and infantile; that we will leave off the pursuit of knowledge as soon as it seems like a machine can do it better, and let the machines do all our thinking for us. Traditional AI promises to make a self-training expert-system, but has not delivered yet. The fusion of the two, should it ever occur, could cause major changes in the world.

One of the largest area of applications of artificial intelligence is in expert sytems, or knowledge based systems as they are often known. This type of system seeks to exploit the specialised skills or information held by of a group of people on specific areas. It can be thought of as a computerised consulting service. It can also be called an information guidance sytem. Such systems are used for prospecting medical diagnosis or as educational aids. They are also used in engineering and manufacture in the control of robots where they inter-relate with vision systems. The initial attempts to apply artificial intelligence to generalised problems made limited progress as we have seen but it was soon realised that more significant progress could be made if the field of interest was restricted.

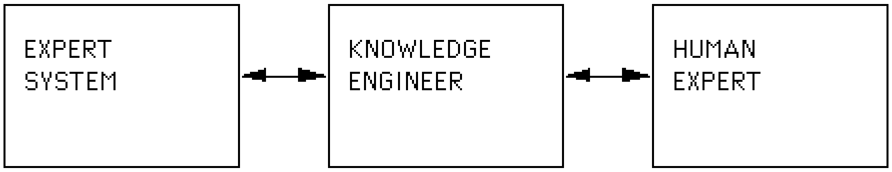
STRUCTURE: The internal structure of an expert system can be considered to consist of three parts: the knowledge base; the database; the rule interpreter. This is analagous to the production system where we have the set of productions; the set of facts held as working memory and a rule interpreter.



The knowledge base holds the set of rules of inference that are used in reasoning. Most of these systems use IF-THEN rules to represent knowledge. Typically systems can have from a few hundred to a few thousand rules. The database gives the context of the problem domain and is generally considered to be a set of useful facts. These are the facts that satisfy the condition part of the condition action rules as the IF THEN rules can be thought of. The rule interpreter is often known as an inference engine and controls the knowledge base using the set of facts to produce even more facts. Communication with the sytem is ideally provided by a natural language interface. This enables a user to interact indelendently of the expert with the intelligent system.

OPERATION OF THE SYSTEM: Again there are three modes to this: the knowledge acquisition mode; the consultation mode; and the explanation mode. We shall consider each in turn.

KNOWLEDGE ACQUISITION:



The system must liaise with people in order to gain knowledge and the people must be specialised in the appropriate area of activity. For example medical doctors, geologists or chemists. The knowledge engineer acts as an intermediary between the specialist and the expert system. Typical of the information that must be gleaned is vocabulary or jargon, general concepts and facts, problems that commonly arise, the solutions to the problems that occur and skills for solving particular problems. This process of picking the brain of an expert is a specialised form of data capture and makes use of interview techniques. The knowledge engineer is also responsible for the self consistency of the data loaded. Thus a number of specific tests have to be performed to ensure that the conclusions reached are sensible.

CONSULTATION: The system is in this mode when a user is interacting with it. The user interacts by entering data in English and the sytem responds by using its inference engine to carry out the reasoning involved in deriving an answer to the questions posed by the user. The role of the inference engine is to search for facts that match the condition part of the productions that match the action part of the question. There is scope for the use of heuristics in this role.

The following example is hypothetical but illustrates the action of typical systems.

System: What is the patient's name?

User: George Washington

System: What is the patient's age?

User: 252 years

System: Is the patient male or female?

User: Male

System: What are the symptoms?

User: Headache dizziness upset stomach high fever

System: When did these symptoms first appear?

User: December 13,1799

System: I conclude that the patient has a viral infection, my recommendation is that he should take two aspirin, drink plenty of fluid, get lots of rest and call me in the morning. George Washington died the next day of a viral infection no antibiotics

EXPLANATION: This mode allows the system to explain its conclusions and its reasoning process. This ability comes from the AND/OR trees created during the production system reasoning process. As a result most expert systems can answer the following why and how questions

why was a given fact used?

why was a given fact not used?

how was a given conclusion reached?

how was it that another conclusion was not reached?

Let us illustrate the action of an expert system by considering the behaviour of an actual one MYCIN a medical expert system.

Mycin has been informed that a patient has contracted meningitis while in hospital. Doctors wish to interrogate Mycin to check this diagnosis.

Mycin then asks the doctors this question

Mycin: Is Meningitis a hospital acquired illness?

Doctor: why?

Mycin: [2.0] this will aid in determining the orgamisms that might be causing the infection

It has already established

[2.1] the infection that requires therapy is meningitis

[2.2] organisms were not seen on the stain of this pending csf culture

[2.3] the type of infection is bacterial

[2.4] neurosurgery is not one of the relevant items of patient 100

[2.5] the age of patient 100 is greater than 25

therefore if [2.6] the infection was acquired while the patient was in hospital then there is evidence that the organisms that might be causing the infection are E coli(0.75) staphylococcus-coag-pos(0.3) pseudomonas-aeruginosa(0.3)

klebsiella-pneumoniae(0.5)

If the doctors wanted to know how something was concluded they might ask

Doctor: How 2.3

The system then reponds with:

Mycin: The following rules concluded about the likelihood that the type of the infection is bacterial:

[2.7] RULE148 (0.19)

[2.8] RULE500 (0.51)

[2.9] RULE501 (0.95)

[3.0] RULE502 (0.97)

[3.1] RULE526 (0.98)

[3.2] RULE504 (0.97)

[3.3] RULE524 (0.98)

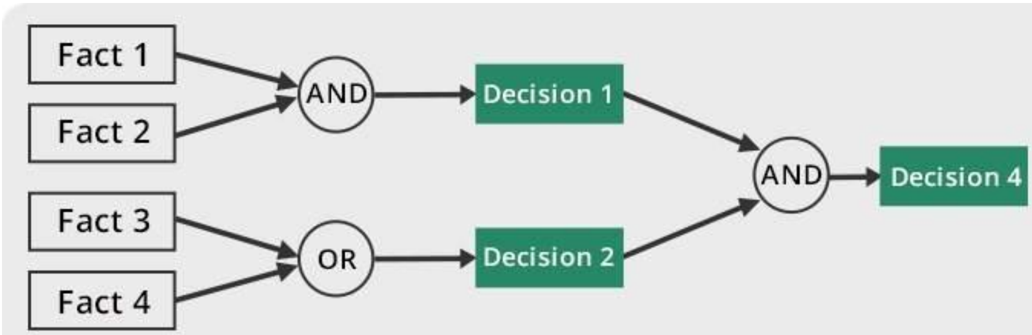
The system takes the doctors through the production system AND/OR tree to explain its conclusions. The numbers in brackets are called certainty factors because little can be absolutely certain. The certainty factors range from -1 to +1 where -1 means absolutely not and +1 absolutely so and 0 undecided either way.

Inference engine: Complete inference engine can be divided into following functional elements: Control system - determines the order of testing the knowledge base rules Rules interpreter - defines a boolean (true, not true uncertainty factor) applications rules Explanation mechanism – justifies user the process of reasoning and generates report A very important element of the expert system is also inference engine. Knowledge of the science must always be stored in the knowledge base, in formalized form, understandable to the inference engine. Using symbols you can easily determine how to handle the system to solve and analyze the correctness of the knowledge base. Since the inference engine is separated from the knowledge base it can be used in skeletal expert systems. Reasoning comes with aplly of certain pattern, which allows tasks to be based on the veracity of premises, request that the truth of a different opinion being requested. It is an attempt to determine the truth of the hypothesis targeted by inference engine. It is based on the assumption that between sentences there is an objective inference ratio, or the ratio of probabilities. Inference requires the ability to make decisions based on their knowledge. The inference is divided into reliable and unreliable..

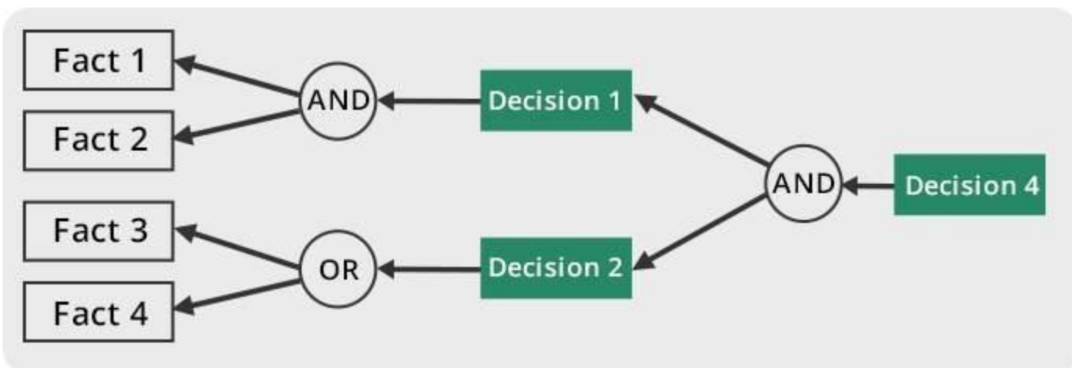
An example of a reliable inference is deductive reasoning. A special variant thereof is syllogistic reasoning from two premises. Syllogism model contains major and minor premise, which show the application. The inference unreliable evidence does not warrant the truthfulness truth of the conclusion. The direction of this inference is considered to be inconsistent with the direction of logical consequence. Such inference is unreliable.

Reductive inference. It is the inference method consisting in choosing the sentence recognized as true (the consequence) of such opinion (the reason), from which it follows logically first. Other examples of inference are unreliable: inductive reasoning by analogy. In general inference can be written as a formula: P1^P2^ ^Pn -> W P1,P2,Pn - evidence W - deduction There are three basic types of reasoning: back (regressive), forward (progressive), and mixed. There are also methods of inference that use the uncertain knowledge . An example of this technique is called fuzzy logic. The most commonly used and most important methods of inference in expert systems are forward and backward chaining.

To recommend a solution, the interface engine uses the following strategies − Forward Chaining, Backward Chaining.



Forward Chaining: It is a strategy of an expert system to answer the question, Here, the interface engine follows the chain of conditions and derivations and finally deduces the outcome. It considers all the facts and rules, and sorts them before concluding to a solution. This strategy is followed for working on conclusion, result, or effect. For example, prediction of share market status as an effect of changes in interest rates.



Backward Chaining: With this strategy, an expert system finds out the answer to the question, "Why this happened?" On the basis of what has already happened, the interface engine tries to find out which conditions could have happened in the past for this result. This strategy is followed for finding out cause or reason. For example, diagnosis of blood cancer in humans.

User Interface: User interface provides interaction between user of the ES and the ES itself. It is generally Natural Language Processing so as to be used by the user who is well-versed in the task domain. The user of the ES need not be necessarily an expert in Artificial Intelligence. It explains how the ES has arrived at a particular recommendation. The explanation may appear in the following forms − Natural language displayed on screen. Verbal narrations in natural language. Listing of rule numbers displayed on the screen. The user interface makes it easy to trace the credibility of the deductions.

Requirements of Efficient ES User Interface It should help users to accomplish their goals in shortest possible way. It should be designed to work for user's existing or desired work practices. Its technology should be adaptable to user's requirements; not the other way round. It should make efficient use of user input.

Uncertainity: Uncertainty is essentially lack of information to formulate a decision. Uncertainty may result in making poor or bad decisions. As living creatures, we are accustomed to dealing with uncertainty – that's how we survive. Dealing with uncertainty requires reasoning under uncertainty along with possessing a lot of common sense.

Theories to Deal with Uncertainty:

Expert systems provide an advantage when dealing with uncertainty as compared to decision trees. With decision trees, all the facts must be known to arrive at an outcome. Probability theory is devoted to dealing with theories of uncertainty. There are many theories of probability – each with advantages and disadvantages.

- Bayesian Probability
- Hartley Theory
- Shannon Theory
- Dempster-Shafer Theory
- Markov Models
- Zadeh's Fuzzy Theory

Expert System Technology:

There are several levels of ES technologies available. Expert systems technologies include −

- **Expert System Development Environment** − The ES development environment includes hardware and tools. They are −
    - Workstations, minicomputers, mainframes.
    - High level Symbolic Programming Languages such as **LIS**t**P**rogramming (LISP) and **PRO**grammation en **LOG**ique (PROLOG).

- Large databases.

- **Tools** – They reduce the effort and cost involved in developing an expert system to large extent.
    - Powerful editors and debugging tools with multi-windows.
    - They provide rapid prototyping
    - Have Inbuilt definitions of model, knowledge representation, and inference design.
- **Shells** – A shell is nothing but an expert system without knowledge base. A shell provides the developers with knowledge acquisition, inference engine, user interface, and explanation facility. For example, few shells are given below −
    - Java Expert System Shell (JESS) that provides fully developed Java API for creating an expert system.
    - *Vidwan*, a shell developed at the National Centre for Software Technology, Mumbai in 1993. It enables knowledge encoding in the form of IF-THEN rules.

VP-Expert: Rule-Based Expert System Development Tool

VP-Expert is the expert system shell we will be using for this course. Like any shell, it contains everything needed for running the expert system (except for the knowledge base of rules for the particular domain). This includes: An inference engine for consulting the KB in order to answer queries. An editor for creatingwriting the rules of the knowledge base. A user interface capable of handling queries, asking the user questions, and presenting traces and explanations where needed. It also has limited graphical capabilities.

Development of Expert Systems: General Steps

The process of ES development is iterative. Steps in developing the ES include −

Identify Problem Domain

- The problem must be suitable for an expert system to solve it.
- Find the experts in task domain for the ES project.
- Establish cost-effectiveness of the system.

Design the System

- Identify the ES Technology
- Know and establish the degree of integration with the other systems and databases.
- Realize how the concepts can represent the domain knowledge best.

Develop the Prototype

From Knowledge Base: The knowledge engineer works to −

- Acquire domain knowledge from the expert.
- Represent it in the form of If-THEN-ELSE rules.

Test and Refine the Prototype

- The knowledge engineer uses sample cases to test the prototype for any deficiencies in performance.
- End users test the prototypes of the ES.

Develop and Complete the ES

- Test and ensure the interaction of the ES with all elements of its environment, including end users, databases, and other information systems.
- Document the ES project well.
- Train the user to use ES.

Maintain the ES

- Keep the knowledge base up-to-date by regular review and update.
- Cater for new interfaces with other information systems, as those systems evolve.

Benefits of Expert Systems

- **Availability** − They are easily available due to mass production of software.
- **Less Production Cost** − Production cost is reasonable. This makes them affordable.
- **Speed** − They offer great speed. They reduce the amount of work an individual puts in.
- **Less Error Rate** − Error rate is low as compared to human errors.

- **Reducing Risk** − They can work in the environment dangerous to humans.
- **Steady response** − They work steadily without getting motional, tensed or fatigued.

Expert Systems Limitations:

No technology can offer easy and complete solution. Large systems are costly, require significant development time, and computer resources. ESs have their limitations which include − Limitations of the technology Difficult knowledge acquisition ES are difficult to maintain High development costs