EXPERT SYSTEMS

. Expert system is a computer system that emulates the decision making of a human expert. The expert knowledge is stored in the computer in an organized manner. This so called knowledge base is used to provide advice in same reasoning process that a human decision maker would go through to arrive at a decision. Expert system differs from others in that:

- It deals with subject matter of realistic complexity
- It must exhibit high performance
- It must be plausible
- Expert Systems, Knowledge Based Systems and Knowledge Based Expert Systems are often used synonymously

CHARACTERISTICS OF AN EXPERT SYSTEM

An expert system is usually designed to have the following general characteristics:

1. **High level Performance:** The system must be capable of responding at high level of competency.
2. **Domain Specificity:** Domain Specificity: Expert systems are typically very domain specific, it is needed to solve the target problem..
3. **Good Reliability:** The expert system must be as reliable as a human expert.
4. **Understandable:** The system should be understandable i.e. be able to explain the steps of reasoning while executing. The expert system should have an explanation capability similar to the reasoning ability of human experts.
5. **Adequate Response time:** The system should be designed in such a way that it is able to perform within a small amount of time, comparable to or better than the time taken by a human expert to reach at a decision point.
6. **Use symbolic representations:** Expert system use symbolic representations for knowledge (rules, networks or frames) and perform their inference through symbolic computations that closely resemble manipulations of natural language.
7. **Linked with Metaknowledge:** Expert systems often reason with metaknowledge i.e. they reason with knowledge about themselves and their own knowledge limits and capabilities.
8. **Expertise knowledge:** expert system must be skillful in applying its knowledge to produce solutions both efficiently and effectively by using the intelligence human experts.

9. **Justified Reasoning:** Justified Reasoning: This allows the users to ask the expert system to justify the solution or advice provided by explaining their reasoning.
10. **Explaining capability:** Expert systems are capable of explaining how a particular conclusion was reached and why requested information is needed during a consultation.
11. **Programming Languages:** Expert systems are typically written in special programming languages. The use of languages like LISP and PROLOG in the development of an expert system simplifies the coding process.

KNOWLEDGE ENGINEERING

The process of building an expert system is called knowledge engineering. Knowledge Engineers acquire the knowledge from a human expert or other source and code in the expert system. The problem of transferring human knowledge into an expert system is so major that it is called the knowledge acquisition bottleneck Major bottlenecks are due to: Cognitive barrier, linguistic barrier, representation barrier and the problem of creating model.

<u>CONVENTIONAL PROGRAMS vs. E</u>

ESs differ from the conventional computer programs in the following aspects:

(i) ESs are knowledge intensive programs

(ii) ESs are highly interactive

(iii) ESs mimic human experts in decision making and reasoning process

(iv) ESs divides expert knowledge into number of separate rules

(v) ESs are user friendly and intelligent.

ES DEVELOPMENT

There are five stages in the development of ES

(i)Identification –determining characteristics of the problem

(ii)Conceptualization –finding concepts to represent the knowledge (iii)Formalization –designing structures to organize knowledge

(iv)Implementation –formulating rules embodying the knowledge

(v)Testing –validating the rules

ES TOOLS

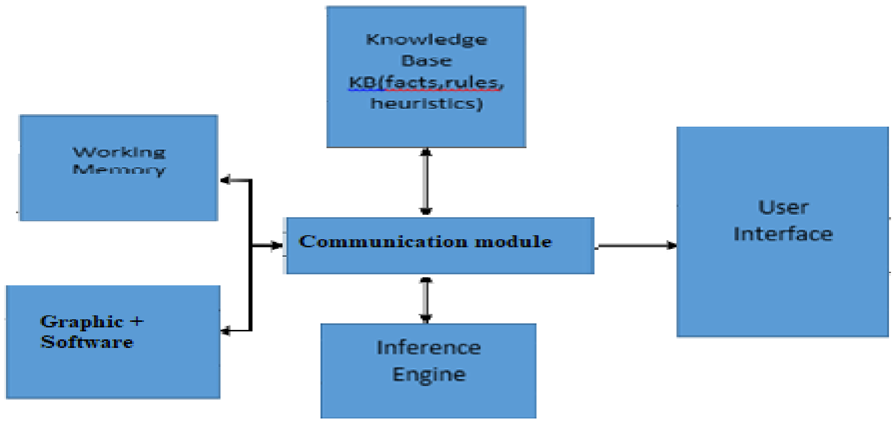
Language: A translator of commands written in a specific syntax. An expert system language will also provide an inference engine to execute the statement of the language.

Shells: A special purpose tool designed for certain types of applications in which the user must only supply the knowledge base. (Eg. EMYCIN)

Tools: A language + utility programs to facilitate the development debugging, and delivery of application programs.

ES ARCHITECTURE

A ES is specific to one problem domain. However, it is not for domain modeling but for problem solving. The expert system consists of (i) a knowledgebase, (ii) a working memory, (iii) an inference engine, (iv) system analysis, graphic and other softwares and (v) user interface.



- Knowledge base consists of declarative knowledge that are facts about the domain and procedural knowledge that are heuristic rules from the domain.
- The working memory is the active set of knowledge base.

- Inference engine is the problem solving module. It also gives justification (explanation) for the advice from the ES.
- Communication module helps in interaction between other modules and also provide user –developer interfaces.

KNOWLEDGE BASE

Knowledge base module contains domain specific knowledge. Knowledge can be either

(i) Prior Knowledge which comes before and is independent of knowledge from the senses. It is considered to be universally true and cannot be denied without contradiction. Ex: All triangles in the plane have 180 degrees

(ii) Posteriori Knowledge that is derived from the senses. It can be denied on the basis of new knowledge without the necessity of contradictions. Ex: The light is green.

Knowledge can be represented in various forms as: (Rules, Semantic Nets, Frames, Scripts , Object Oriented)

Rules: The most popular format of rules are the IF –condition –THEN –action statements.

This is useful when the knowledge is in the form of condition action. P1,....,Pm==> Q1, ..., Q n means if premises P1and ..., and Pm are true then perform actions Q1 and..., and Qn . An example of a rule is

IF Inflow < 0.7 * Average

AND Storage < Capacity /2

THEN irrigation release = 0.6* Demand.

INFERENCE ENGINE

This module examines the knowledge base and answers the questions (how and why) from the user. It is the most crucial component of ES. It derives the knowledge i.e, guides the selection of a proper response to a specific situation which is called pruning. Three formal approaches used in this case are: production rules, structured objects and predicate logic. Production rules consist of a rule set, a rule interpreter which specifies when and how to apply the rules and a working memory which holds the data, goals and intermediate results. Structured objects use vector representation of essential and accidental properties. Predicate logic uses propositional and predicate calculi. The inference engine can work in the following ways:

1. Forward Chaining

2. Backward Chaining

3. Abduction

4. Reasoning under Uncertainty

1. Forward Chaining (Bottom –up reasoning)

The inference engine searches the knowledge base with the given information for rules whose precedence matches the given current state.

The basic steps are;

(i)The system is given one or more conditions

(ii)The system searches the rules in the knowledge base for each condition correspond to IF part are selected.

(iii)Each rule can generate new conditions from the conclusions of the invoked THEN part, which in turn are again added to the existing ones

(iv) The added conditions, if any will be processed again (step ii)

2. Backward Chaining (Top-down reasoning)

The system selects a goal state and reasons in the backward direction. The initial state condition is established for the goal to be true. If the given initial state conditions matches with the established ones, then the goal is the solution. Otherwise, the system selects another goal and the process is repeated. The basic steps are:

(i) Select a goal state and rules whose THEN portion has the goal state as conclusion

(ii) Establish sub goals to be satisfied for the goal state to be true, from the IF portion of the selected rules.

(iii) Establish initial conditions necessary to satisfy all the sub goals.

(iv) Check whether the given initial state matches with the established ones. If so, then the goal is one solution. If not, select another goal state.

3. Abduction

Reasoning from observed facts to the best explanation

$$p \rightarrow q, \; q \text{ proves } p$$

Abduction is related to the analysis of backward chaining and implication. Abduction is a mathematically justifiable, practical, and reasonable way to generate hypotheses. Abduction is another name for a fallacious argument. It is not guaranteed to work.

4. Reasoning under Uncertainty

When knowledge is certain, the conclusions are also certain. We can use the normal rules of logic to deduce conclusions.

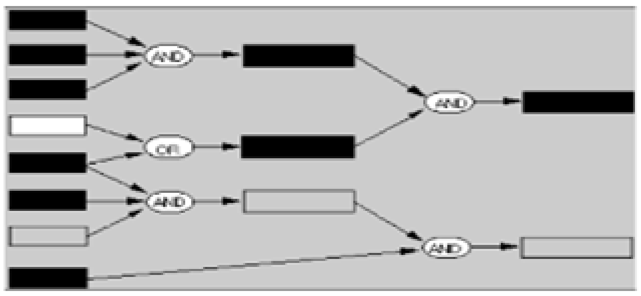


Fig.4. Reasoning under Certainty

Often, experts can't give definite answers. It may require an inference mechanism that derives conclusions by combining

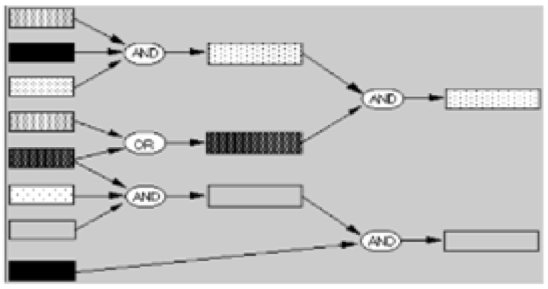


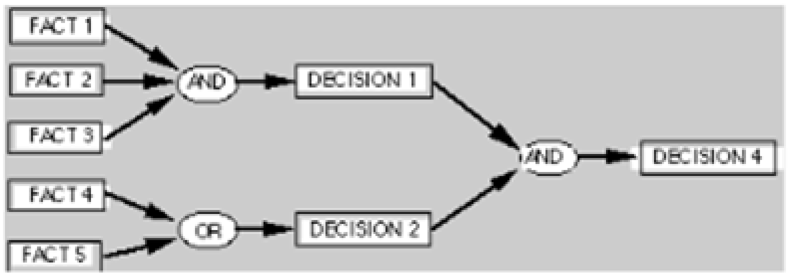
Fig.5. Reasoning under Uncertainty

Inference engine –explanations

An expert system seeks to make problem solving knowledge explicit. The knowledge applied to a problem must be available to the user. The system must be able to explain how it arrived at a conclusion and why it is performing some computation. It may also be required to answer what if questions.

Answering HOW?

To answer how a conclusion was reached, work back through the inference chain. Decision 4 was made as a result of making decision 1 and decision 2 .Decision 1 was made because Facts 1, 2 & 3 are true, etc.



Answering WHY?

To answer why a computation is being performed, the system must state its current goal. The system may ask the user if fact 3 is true because it is trying to determine if decision 1 should be made

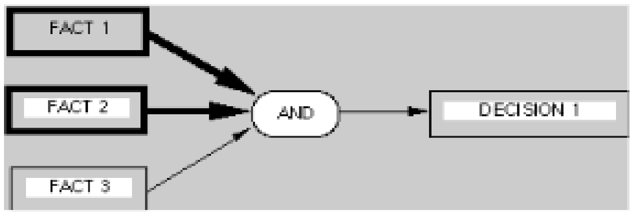


Fig. 7

LEARNING BY INDUCTION

Inductive learning is the process of acquiring generalized knowledge from examples or instances of some class. This form of learning is accomplished through inductive inference, the process of learning from a part to a whole, from particular instances to generalizations or from the individual to the universal. It is a powerful form of learning which we humans do almost effortlessly. Even though it is not a valid form of inference, it appears to work well much of the time.

Examples

- We conclude that “weather in South India is always pleasant in winter”, by observing a few seasons.

- "All swans are white": After seeing only a small number of white swans.
- "All North Indians speak Hindi": After talking to a few people in North India.
- The inductive process can be described symbolically through the use of predicates P and Q. If we observe repeated occurrence of events P(a1), P(a2), ..., P(ak), we generalize by inductively concluding that for all x, P(x), Q(y) will happen (ex. Paddy – Green).

Different Approaches

- Learning by Observation
- Learning by Discovery
- Supervised learning
- Learning from examples
- Unsupervised learning

CLIPS(C Language Integrated Production System)

CLIPS is a tool for Building Expert Systems http://www.ghg.net/clips/CLIPS.html. CLIPS is a multi-paradigm programming language that provides support for rule-based object-oriented, and procedural programming. It was designed with the specific purpose of providing high portability, low cost, and easy integration with external systems.

The main characteristics are:

- Expert system shell
- It has an excellent external language integration
- Uses forward chaining based on Rete's algorithm
- Allows both rule based and procedural programming paradigms