EXPERT SYSTEM

Introduction

Definition

Artificial Intelligence is a piece of software that simulates the behaviour and judgment of a human or an organization that has experts in a particular domain is known as an expert system

- **Acquiring knowledge**
- **Interpretation as per user problem**
- **Knowledge added by expert**
- **Useful for non-experts**

Application

- **Medical diagnosis, accounting, coding, games etc**

Introduction

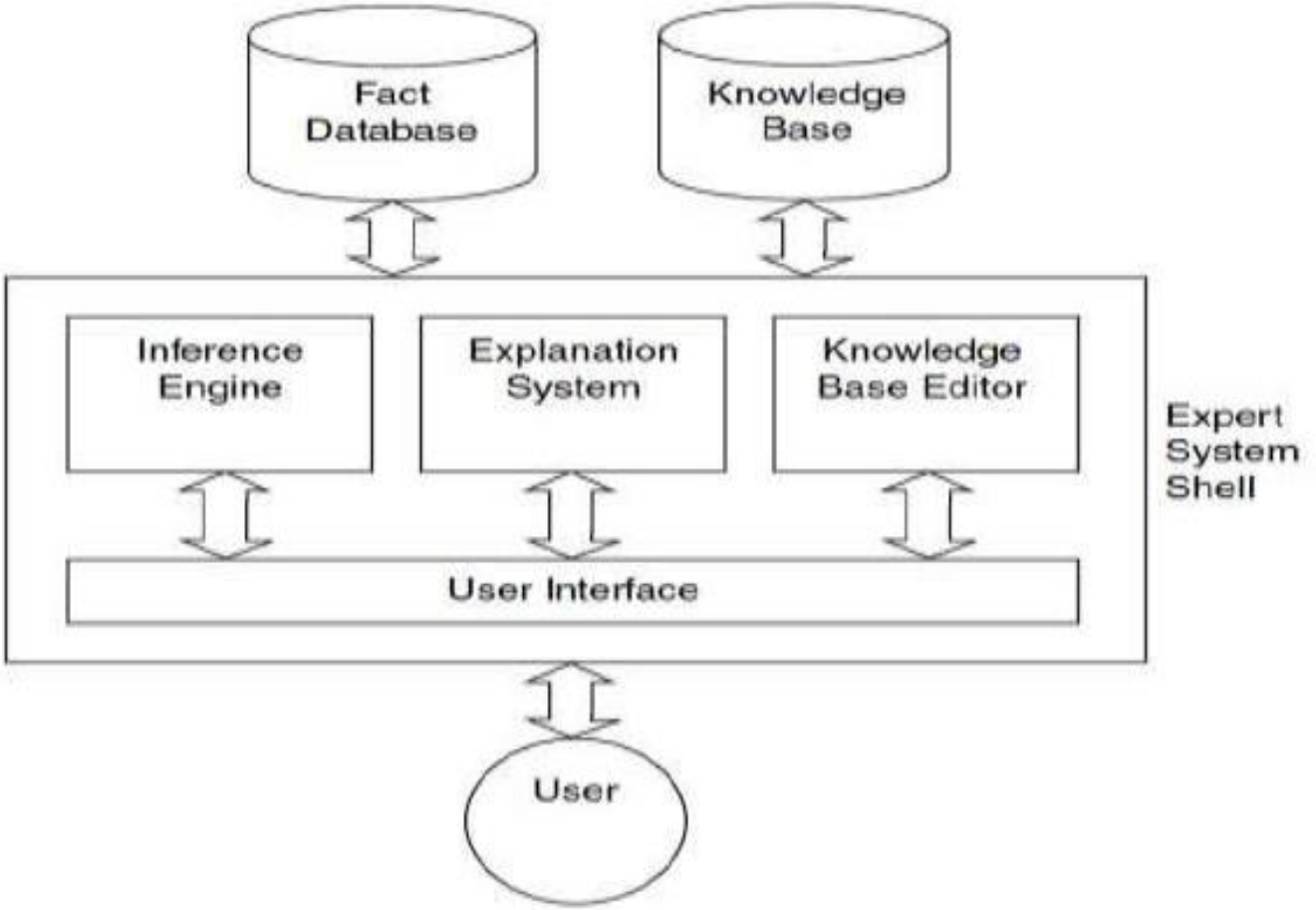
Examples

- **MYCIN: One of the earliest expert systems based on backward chaining. It can identify various bacteria that can cause severe infections and can also recommend drugs based on the person's weight**
- **DENDRAL: It was an artificial intelligence based expert system used for chemical analysis. It used a substance's spectrographic data to predict it's molecular structure**
- **CaDet: It is a clinical support system that could identify cancer in its early stages in patients**
- **DXplain: It was also a clinical support system that could suggest a variety of diseases based on the findings of the doctor**
- **PXDES: It could easily determine the type and the degree of lung cancer in a patient based on the data**

People Involved in an Expert System

- The end-user of the system is the person who has the need for the system
- The knowledge engineer is the person who designs the rules for the system, based on either observing the expert at work or by asking the expert questions about how he or she works
- The domain expert is very important to the design of an expert system

Architecture of an Expert System



Architecture of an Expert System

- The knowledge base contains the specific domain knowledge that is used by an expert to derive conclusions from facts
- The explanation system provides information to the user about how the inference engine arrived at its conclusions
- The fact database contains the case-specific data that are to be used in a particular case to derive a conclusion
- The user of the expert system interfaces with it through a user interface, which provides access to the inference engine, the explanation system, and the knowledge-base editor
- The knowledge-base editor allows the user to edit the information that is contained in the knowledge base
- The inference engine is the part of the system that uses the rules and facts to derive conclusions. The inference engine will use forward chaining, backward chaining, or a combination of the two to make inferences from the data that are available to it

Forward Chaining

- **The system starts from a set of facts, and a set of rules, and tries to find a way of using those rules and facts to deduce a conclusion or come up with a suitable course of action**
- **Data-driven reasoning**
- **When applying forward chaining, the first step is to take the facts in the fact database and see if any combination of these matches all the antecedents of one of the rules in the rule database. When all the antecedents of a rule are matched by facts in the database, then this rule is triggered. Usually, when a rule is triggered, it is then fired, which means its conclusion is added to the facts database. If the conclusion of the rule that has fired is an action or a recommendation, then the system may cause that action to take place or the recommendation to be made**

Example

- consider the following set of rules that is used to control an elevator in a three-story building:

Rule 1

IF on first floor and button is pressed on first floor

THEN open door

Rule 2

IF on first floor

AND button is pressed on second floor

THEN go to second floor

Rule 3

IF on first floor

AND button is pressed on third floor

THEN go to third floor

Example Cont.….

Rule 4

IF on second floor AND button is pressed on first floor

AND already going to third floor

THEN remember to go to first floor later

Let us imagine that we start with the following facts in our database:

Fact 1

At first floor

Fact 2

Button pressed on third floor

Fact 3

Today is Tuesday

The system examines the rules and finds that Facts 1 and 2 match the antecedents of Rule 3. Hence, Rule 3 fires, and its conclusion Go to third floor is added to the database of facts

Example cont..

Lets imagine another facts

Fact 1

At first floor

Fact 2

Button pressed on second floor

Fact 3

Button pressed on third floor

- **In this case, two rules are triggered—Rules 2 and 3. In such cases where there is more than one possible conclusion, conflict resolution needs to be applied to decide which rule to fire**

Conflict Resolution

- **In a situation where more than one conclusion can be deduced from a set of facts, there are a number of possible ways to decide which rule to fire (i.e., which conclusion to use or which course of action to take)**

For example, consider the following set of rules:

IF it is cold THEN wear a coat

IF it is cold THEN stay at home

IF it is cold THEN turn on the heat

If there is a single fact in the fact database, which is "it is cold," then clearly there are three conclusions that can be derived

Conflict Resolution

- **Method-1**

  Rules are given priority levels, and when a conflict occurs, the rule that has the highest priority is fired

Example:

IF patient has pain
THEN prescribe painkillers priority 10

IF patient has chest pain
THEN treat for heart disease priority 100

- **Method-2**

  Longest-matching strategy. This method involves firing the conclusion that was derived from the longest rule

Conflict Resolution

Example:

IF patient has pain
THEN prescribe painkiller

IF patient has chest pain
AND patient is over 60
AND patient has history of heart conditions
THEN take to emergency room

- **Method-3**

  Fire the rule that has matched the facts most recently added to the database

Backward Chaining

- **In backward chaining, we start from a conclusion, which is the hypothesis we wish to prove, and we aim to show how that conclusion can be reached from the rules and facts in the database**
- **Goal – driven reasoning**

Comparing Forward and Backward Chaining

Let us use an example to compare forward and backward chaining

Rules:

Rule 1 : $A \wedge B \rightarrow C$
Rule 2 : $A \rightarrow D$
Rule 3 : $C \wedge D \rightarrow E$
Rule 4 : $B \wedge E \wedge F \rightarrow G$
Rule 5 : $A \wedge E \rightarrow H$
Rule 6 : $D \wedge E \wedge H \rightarrow I$

Facts:

Fact 1 : A
Fact 2 : B
Fact 3 : F

Goal:

Our goal is to prove H

Forward chaining

Facts	Rules triggered	Rule fired
A, B, F	1, 2	1
A, B, C, F	2	2
A, B, C, D, F	3	3
A, B, C, D, E, F	4, 5	4
A, B, C, D, E, F, G	5	5
A, B, C, D, E, F, G, H	6	STOP

Backward chaining

Facts	Goals	Matching rules
A, B, F	H	5
A, B, F	E	3
A, B, F	C, D	1
A, B, C, F	D	2
A, B, C, D, F		STOP

Expert System Shell

- The expert system that do not contain domain-specific or case-specific information are contained within the expert system shell
- This shell is a general toolkit that can be used to build a number of different expert systems, depending on which knowledge base is added to the shell.
- An example of such a shell is CLIPS (C Language Integrated Production System), which is a public domain software tool for building expert systems
- Other examples in common use include OPS5, ART, JESS

Rete Algorithm

- One potential problem with expert systems is the number of comparisons that need to be made between rules and facts in the database. In some cases, where there are hundreds or even thousands of rules, running comparisons against each rule can be impractical
- The Rete Algorithm is an efficient method for solving this problem and is used by a number of expert system tools, including OPS5 and Eclipse
- The Rete is a directed, acyclic, rooted graph Each path from the root node to a leaf in the tree represents the left-hand side of a rule. Each node stores details of which facts have been matched by the rules at that point in the path
- As facts are changed, the new facts are propagated through the Rete from the root node to the leaves, changing the information stored at nodes appropriately. This could mean adding a new fact, or changing information about an old fact, or deleting an old fact
- In this way, the system only needs to test each new fact against the rules, and only against those rules to which the new fact is relevant, instead of checking each fact against each rule

Limitations

- **Don’t have human-like decision making power**
- **Can’t possess human capabilities**
- **Can’t produce correct result from less amount of knowledge**
- **Requires excessive training**

Advantages

- **Low accessibility cost**
- **Fast response**
- **Not affected by emotions unlike humans**
- **Low error rate**
- **Capable of explaining how they reached a solution**

Disadvantages

- **Expert system have no emotions**
- **Common sense is the main issue of the expert system**
- **It is developed for a specific domain**
- **It needs to be updated manually. It does not learn itself**
- **Not capable to explain the logic behind the decision**