Sunday

# MID TERM

23/04/2023

## Expert System :-

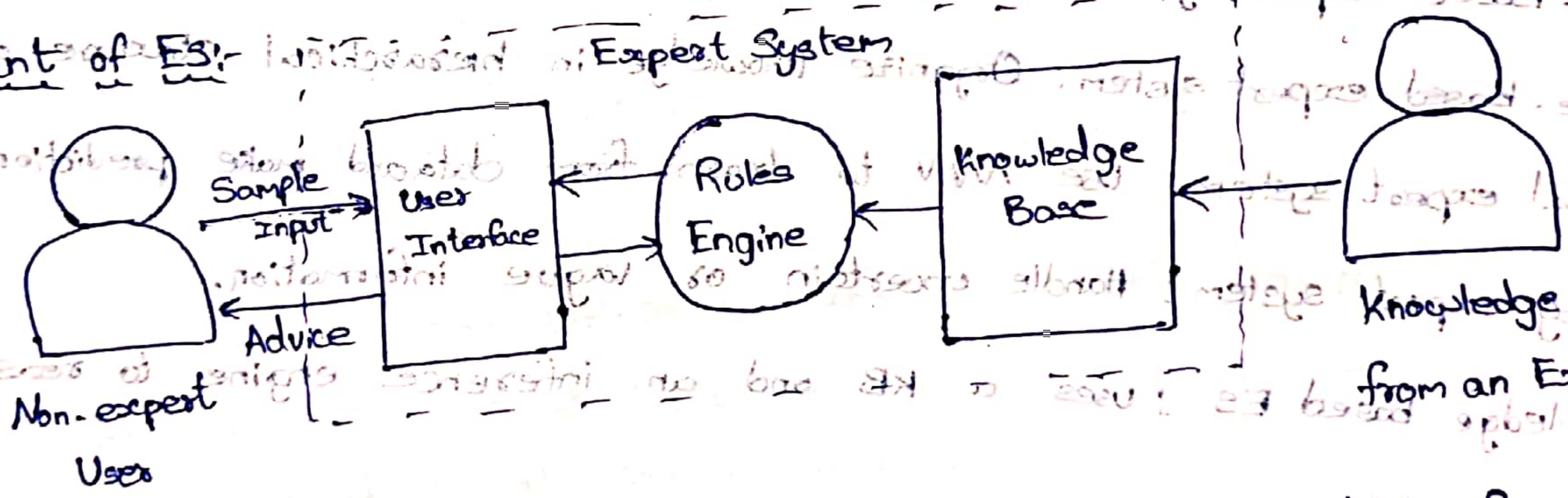
An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert. It perfoms this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.

[→ It solves the most complex issue as an expert by extracting the knowledge stored]

-> The performance of an expert system which solves complex problems, is based on the expert's knowledge stored in its knowledge base. The more knowledge stored in the knowledge base, the more that system improves its performance.

-> Expert systems use both factual knowledge and heuristics to solve complex problems within a specific domain, such as medicine (or) science.

## Component of ES:-




**User Interface :-** It allows the user to input queries in a readable format, which are then processed by the inference engine. The output from the engine is displayed back to the user as a solution.

**Inference engine:** It is the processing unit of the expert system that applies inference rules to the knowledge base to deduce new information and derive conclusions. It is responsible for providing the error-free solutions to the user's queries.

**Knowledge Base** :- It is a storage system for domain-specific knowledge that helps the expert system. The size and quality of the knowledge base directly impact the accuracy and precision of the expert system.

## Characteristics of Expert System :-

- Expert system provide high-performance solutions for complex problems in specific domains.
- They are easily understandable by users, taking input and providing output in human language.
- The expert system is reliable and generates accurate results.
- Expert system provide solution for any complex problem within a short period of time.

## Types of Expert System :-

- Rule-based expert system : Use logical rules to make decisions in a domain.
- Frame-based expert system : Organize knowledge in hierarchical structure of frames.
- Neural expert system : Use ANN to learn from data and make predictions.
- Fuzzy expert system : Handle uncertain or vague information.
- knowledge based ES : uses a KB and an inference engine to reason.

## Examples of Expert System :-

i) **DENDRAL** - A molecular structure prediction tool for chemical analysis

ii) **PXDES** - Predicts the kind and extent of lung cancer.

iii) **MYCIN** - used for recommendation of antibiotics and the diagnosis of blood clotting diseases.

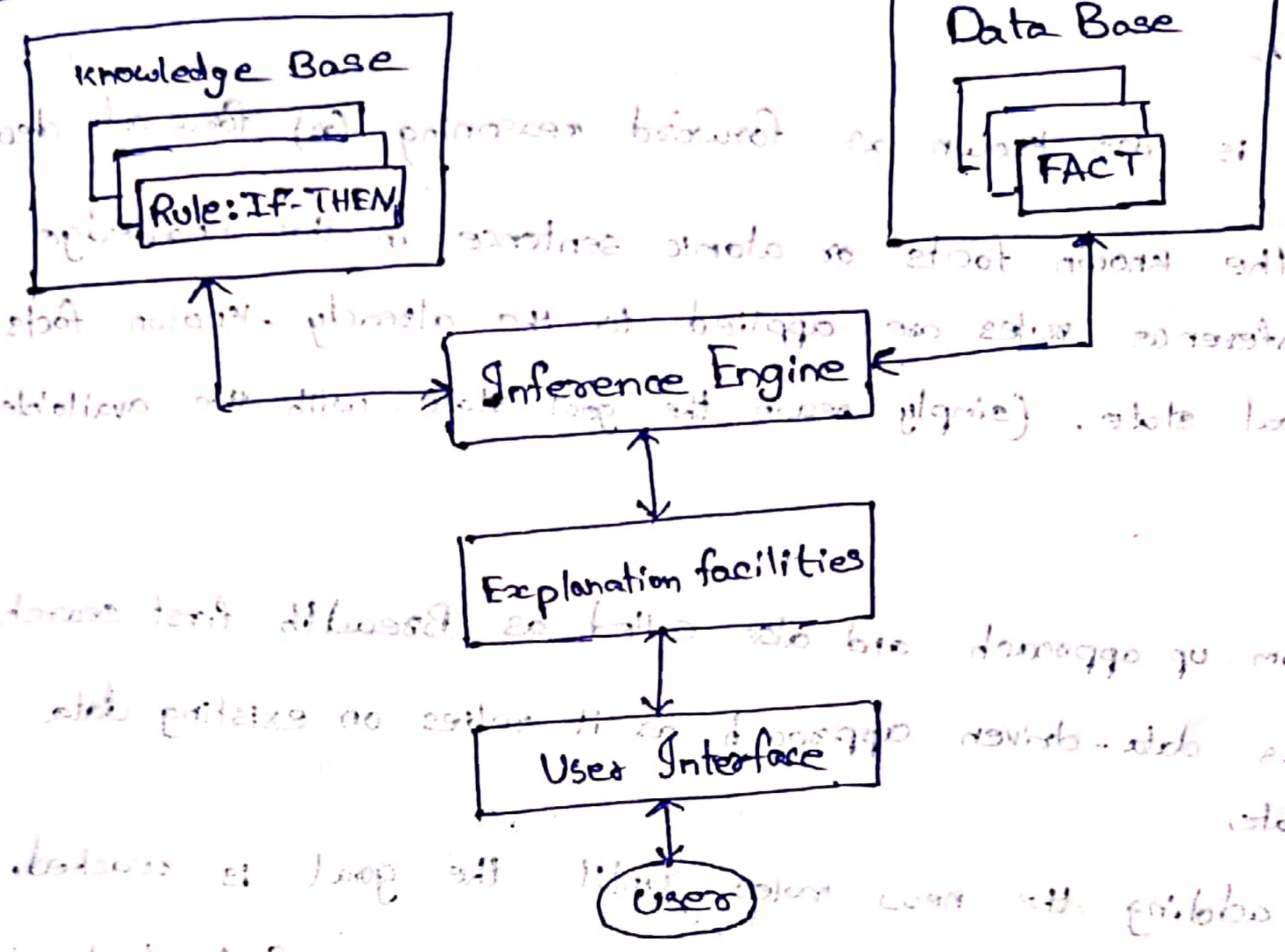
## Rule-based system:-

It is a type of AI system that uses a knowledge base of predefined rules to make decision or solve problem in a particular domain.

Any rule consists of two parts:-

- The "If" Part is called as the antecedent (condition)
- The "Then" Part is called as the consequent (action)

## Structure of Rule-based:-




- The knowledge Base contains the domain knowledge useful for problem solving.
- The Data Base contains a set of facts used to match against the IF (condition) Parts of rules stored in the knowledge base.
- The inference engineer carry out the reasoning where by the expert system reaches a solution. It links the rules given by the KB with the facts provided in the DB.
- The explanation facilities enables the user to ask the expert system how a particular conclusion is reached and why a specific fact is need.
- The user interface is the means of communication between a user seeking a solution to the problem and an expert system.

## Forward & backward chaining :-

→ Inference Engine is one of the major components of the expert system that applies a set of logical rules to the existing information (knowledge Base) to deduce new information from the already known facts.

→ Forward and Backward Chaining are the two modes by which the Inference engine deduces new information.

## Forward Chaining :-

→ Forward chaining is also known as forward reasoning (or) forward deduction

→ It starts with the known facts or atomic sentence in the knowledge base and gradually inference rules are applied to the already known facts till we reach the goal state. (simply reach the goal state with the available data)

## Properties :

→ It follows a bottom-up approach and also called as Breadth first search.

→ It is also called as data-driven approach, as it relies on existing data to reach the goal state.

→ The system keeps adding the new rules until the goal is reached.

→ It is more efficent when the system has large set of facts to begin with

→ Used in situations where the system needs to predict or diagnose psoblems based on avaiable data.

Example: Rule 1: If A is human Then A is mammal
Rule 2: If A is mammal, Then A is living
Rule 3: If A is living Then A is mortal

### ADV

→ Can handle large amount of data
→ Can be used to find multiple solutions
→ useful in expert system that requires Control, planning, monitoring.

### DISADV

→ Can produce large no/of false positives
→ prone to errors, if the rules are poorly defined
→ Can be slow or inefficent when there are many rules.

## Backward Chaining :-

-> Backward chaining is also known as backward Propagation (or) backward deduction

-> It starts with the goal state and looks backward to determine the possible facts that could lead to that conclusion by breaking down the problem into a series of sub problems, and then using a set of rules (or) if-then statements to deduce the cause of each sub-problem.

## Properties :-

-> It follows a Top-bottom approach which also called as goal driven approach.

-> Backward chaining uses depth first search & searchs only requried facts.

-> The system keeps asking questions until it has enough information to reach conclusion

-> used in situations where the system needs to find out why something has happened or what caused it.

## Example :-

- Start with the goal : getting the car started
- work backward through rules to find a solution
- Check for conditions in each rule and take corresponding actions
- Achieve the goal by jump starting the car.

| ADV | Disadv |
|---|---|
| -> performs only requried inferences | -> Can't handel large amount of data. |
| -> Can solve more complex problems | -> Can't find all possible solutions for problem |
| -> efficient when only a small noof rules apply. | -> difficult to implement backward chaining. |

## Types of knowledge in Expert system :-

1. **Declarative knowledge :-**

factual statements about the domain. This is used by system to draw conclusions.

2. **Procedural knowledge :-**

used to know how to perform a task or procedure. This is used by system to perform tasks (or) make decisions.

3. **Heuristic knowledge :-**

This is based on the experience. This is used by system to make guesses or judgments in uncertain situations (or) incomplete information.

4. **Structural knowledge :-**

This is concerned with the organization and structure of the domain. This is used by the system to organize and retrieve info from the KB.

5. **Meta- knowledge :-**

knowledge about knowledge itself. This helps in improvement of the system by optimizing the use of the diff types of knowledge.

## Conflict Resolution:

- Conflict Resolution resolves conflicting rules or information to arrive at a single conclusion.
- Expert system use various methods to resolve conflicts, by assigning the Priorities to rules or using weights to evaluate the importance of information
- The goal is to arrive at a decision that follows the rules and heuristics in their knowledge base.
- By resolving conflict efficiently, expert system can provide accurate advice in complex decision-making scenarios.

eg:- Let us consider two simple rules for crossing a road. And let us add a third rule:-

Rule1: If the "Traffic light" is green, then the action is to go.

Rule2: If the "Traffic light" is Red, then the action is to stop.

Rule3: If the "Traffic light" is Red, then the action is to go.

-> Here, both the Rules 2 & 3 are having same IF Part. Thus both of them are set to fire when the condition part is satisfied. These rules represent a conflict set.

-> The inference engin must determine which rule to fire from such a set.

-> A method to choosing a rule to fire, when more than one rule can be fired in a given cycle is called conflict resolution.

### Steps to solve:-

i) First in First Serve :- The first rule that is matched will be the one that is applied.

ii) Last in First serve :- The last rule that is matched will be the one that is applied.

iii) Prioritisation :- The selection of the rules to apply is determined by the prioritization of rules, which is typically established by an expert.

iv) Specificity :- The rule applied is usually the most specific rule or the rule that matches the most facts.

- Recency :- The rule applied is the rule that matches the most recently derived facts.
- Fired rules involves not using the rule that are already fired.
- Line of Reasoning :- Involves firing all rules with a separate line of reasoning

## Combining Neural Network with expert system :-

-> A Neural network may be defined as information processing model that is inspired by the human brain.

-> It consists of inter connected nodes (or) neurons that transmit and process information to each other.

-> Neural networks are used in various applications such as image recognition, NLP, speech recognition, and predictive analytics.

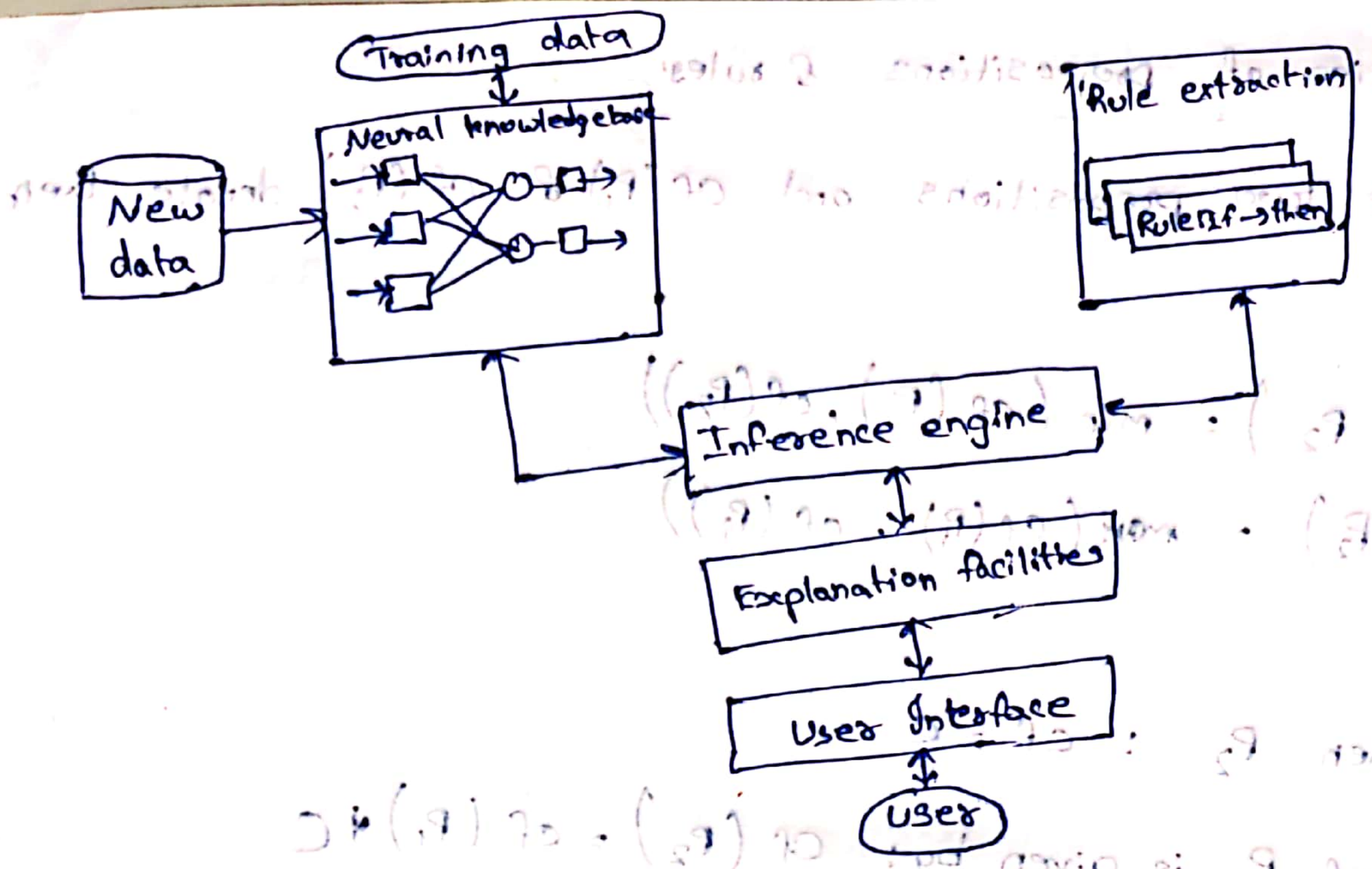
- Expert system :
  - Adpats to customizations and theories.
  - Generates multiple solutions and explain them.

- ANN :
  - Not adpatble for customizations and Based on past Data.
  - Provides exact solution but can't explain it.

-> Combining ANN with expert system is benifical for solving complex problems that requires both statistical learning and domain specific knowledge.

-> Expert systems are AI application that uses knowledge base of human expertise for Problem solving.

-> IN neural network expert system, the knowledge is encoded in the weights and the ANN generates the inference rules.




Rule-based NN :- NN used as a classifier to predict the output of a rule-based ES.

NN-based expert system :- NN used to replace the Rule-based ES, learns rules directly from data

Hybrid system :- The ES provides rules and constrains to NN to learn

Fuzzy ES :- Fuzzy logic to combine the output of ES & NN to handel uncertainties

## Frame-based expert system:

It is a type of KB system that uses frames, also known as schema (or) templets, to represent and organize the knowledge about a particular domain. A frame is a data structure that consists of set of attributes, or slots that describe the properties of an object or concept in the domain.

-> each frame has its own name and set of attributes associated with it

eg: Name, weight, height and age are slots in the frame of a person.

model, Processor, memory and price are slots in the frame of a "computer".

-> Frams are an application of object-oriented programming for expert system.

-> Frame-based expert systems also provide an extension to the slot-value structure through the application of facets.

-> A facets is a means of providing extered knowledge about an attribute of a frame.

## Combining certainties of propositions & rules:-

→ let $P_1$ and $P_2$ be two propositions and $cf(P_1)$ & $cf(P_2)$ denote their certainities.

Then
$$cf(P_1 \text{ and } P_2) = \min(cf(P_1), cf(P_2))$$
$$cf(P_1 \text{ or } P_2) = \max(cf(P_1), cf(P_2))$$

Given the rule :-

if $P_1$ then $P_2$ : cf = C

and certainty of $P_2$ is given by: $Cf(P_2) = cf(P_1) * C$

-> $cf(H_1 E) = cf(E) * cf$

For example, IF sky is clear, Then the forecast is sunny {cf 0.8}
and the current certainty factor of sky is clear is 0.5.

Then $cf(H_1 E) = (0.5)(0.8) = 0.4$

This result can be interpreted as "It may be sunny".

## Predicate conversion:-

original rule: If student has completed all the requried coursework then they will receive a pass grade.

Predicate logic format : $\forall x$ : CompletedAllCoursework(x) $\longrightarrow$ ReceivedPassingGrade(x)

using the logical equivalence : $p \longrightarrow Q \cong \neg Q \longrightarrow \neg P$

Converted rule: If a student did not receive a passing grade, then they must not have completed all the requried coursework.

Predicate logic format: $\forall x$ : $\neg$ ReceivedPassingGrade(x) $\longrightarrow$ $\neg$ CompletedAllCoursework(x)