Artificial Intelligence Week 1 and Week 2

What is AI?

Systems that think like humans

Systems that think rationally

Systems that act like humans

Systems that act rationally

The Turing test

The Turing Test, introduced by Alan Turing in 1950, is a crucial milestone in the history of artificial intelligence (AI). It came to light in his paper titled 'Computing Machinery and Intelligence.' Turing aimed to address a profound question: Can machines mimic human-like intelligence?

Rational agents

An agent is an entity that perceives and acts

This course is about designing rational agents

Abstractly, an agent is a function from percept histories to actions:

$f : P* \rightarrow A$

For any given class of environments and tasks, we seek the

agent (or class of agents) with the best performance

A rational agent in AI is an agent that performs actions to achieve the best possible outcome based on its perceptions and knowledge. It operates under the premise of rationality, where it consistently makes decisions that maximize its expected utility or performance measure. Rational agents can be found in various AI applications, including robotics, automated trading systems, and decision support systems.

PEAS

to design a rational agent we must specify the task environment

P- perfonamce measure

E- Environment

A-Actuators

S-sensor

Components of a Rational Agent

A rational agent comprises several key components:

Perception: The ability to perceive the environment through sensors.

Knowledge Base: Information the agent has about the environment and itself.

Decision-Making Process: Algorithms and rules that guide the agent's actions.

Action: The ability to perform actions that affect the environment through actuators.

1. **Simple Reflex agent**:

The Simple reflex agents are the simplest agents. These agents take decisions on the basis of the current percepts and ignore the rest of the percept history.

These agents only succeed in the fully observable environment.

The Simple reflex agent does not consider any part of percepts history during their decision and action process.

The Simple reflex agent works on Condition-action rule, which means it maps the current state to action. Such as a Room Cleaner agent, it works only if there is dirt in the room.

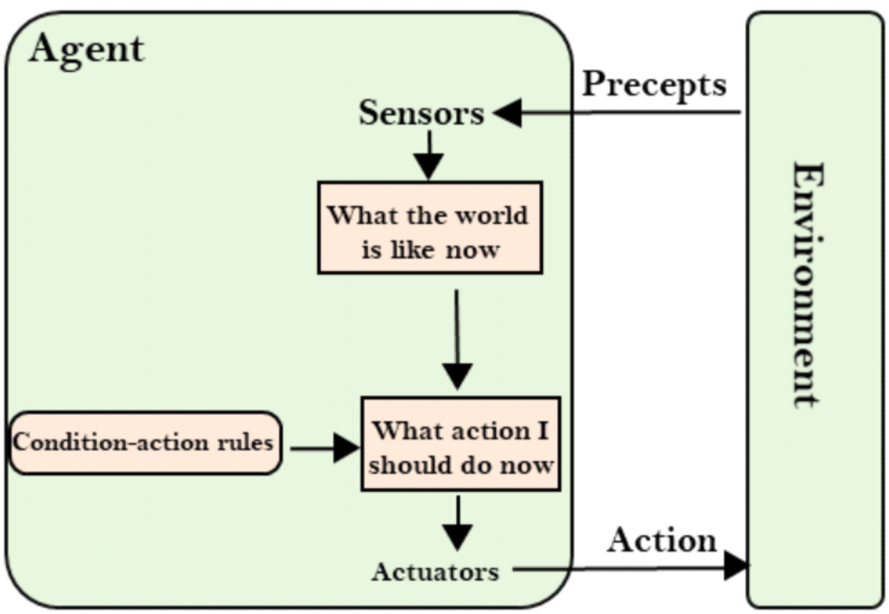
Problems for the simple reflex agent design approach:

They have very limited intelligence

They do not have knowledge of non-perceptual parts of the current state

Mostly too big to generate and to store.

Not adaptive to changes in the environment.



2. **Model-based reflex agent**

The Model-based agent can work in a partially observable environment, and track the situation.

A model-based agent has two important factors:

Model: It is knowledge about "how things happen in the world," so it is called a Model-based agent.

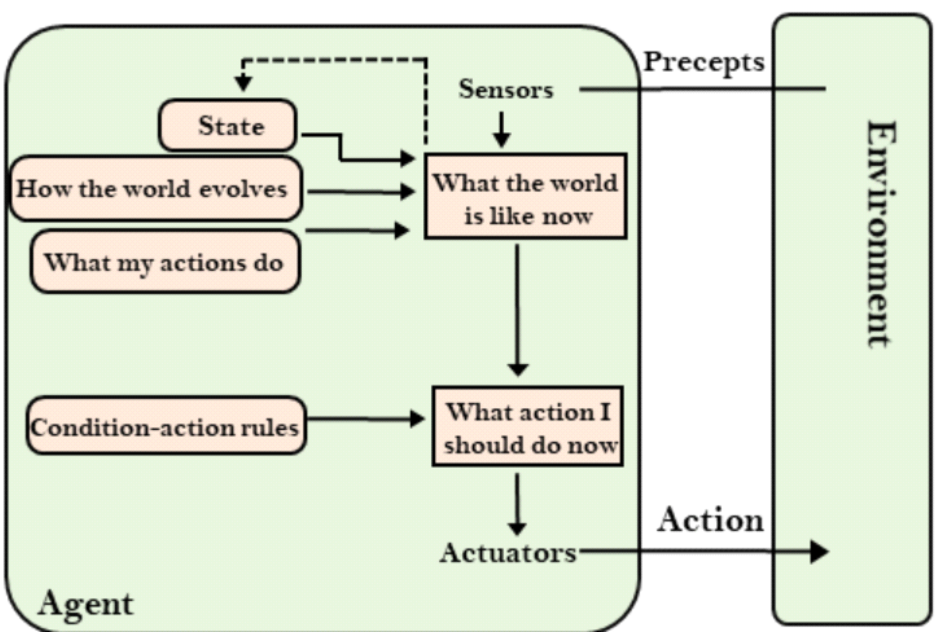
Internal State: It is a representation of the current state based on percept history.

These agents have the model, "which is knowledge of the world" and based on the model they perform actions.

Updating the agent state requires information about:

How the world evolves

How the agent's action affects the world.



3. **Goal-based agents**

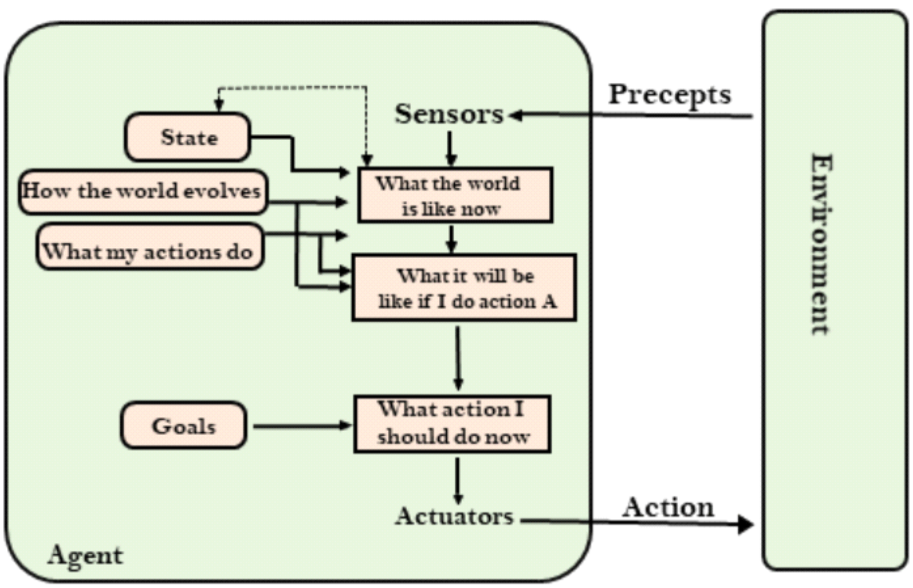
The knowledge of the current state environment is not always sufficient to decide for an agent to what to do.

The agent needs to know its goal which describes desirable situations.

Goal-based agents expand the capabilities of the model-based agent by having the "goal" information.

They choose an action, so that they can achieve the goal.

These agents may have to consider a long sequence of possible actions before deciding whether the goal is achieved or not. Such considerations of different scenario are called searching and planning, which makes an agent proactive.



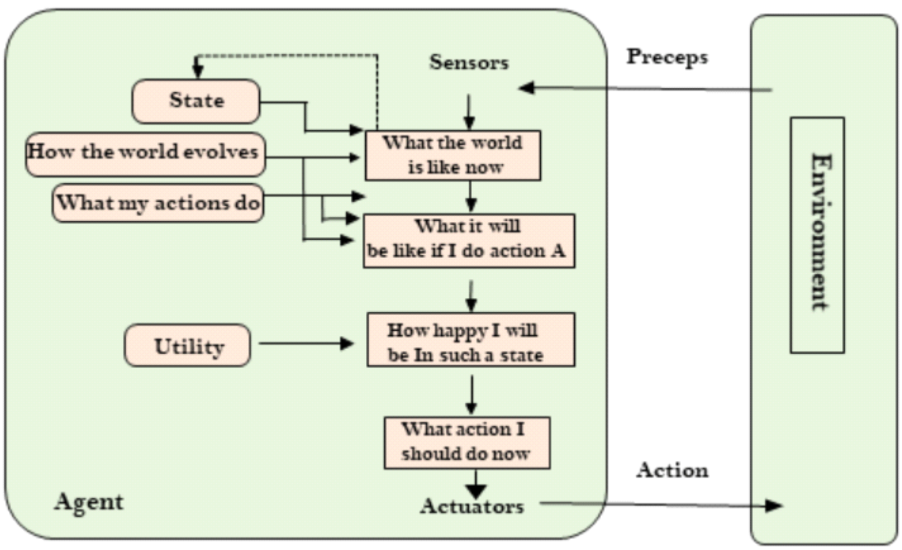
4. **Utility-based agents**

These agents are similar to the goal-based agent but provide an extra component of utility measurement which makes them different by providing a measure of success at a given state.

Utility-based agent act based not only goals but also the best way to achieve the goal.

The Utility-based agent is useful when there are multiple possible alternatives, and an agent has to choose in order to perform the best action.

The utility function maps each state to a real number to check how efficiently each action achieves the goals.



5. **Learning Agents**

A learning agent in AI is the type of agent which can learn from its past experiences, or it has learning capabilities.

It starts to act with basic knowledge and then able to act and adapt automatically through learning.

A learning agent has mainly four conceptual components, which are:

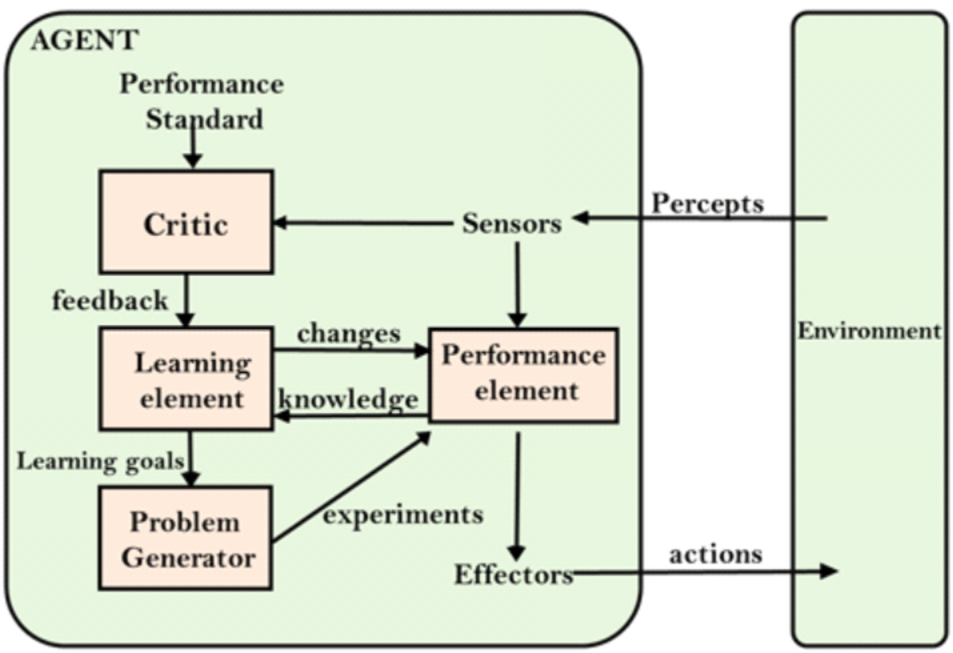
Learning element: It is responsible for making improvements by learning from environment

Critic: Learning element takes feedback from critic which describes that how well the agent is doing with respect to a fixed performance standard.

Performance element: It is responsible for selecting external action

Problem generator: This component is responsible for suggesting actions that will lead to new and informative experiences.

Hence, learning agents are able to learn, analyze performance, and look for new ways to improve the performance.



Environment:

An environment is everything in the world which surrounds the agent, but it is not a part of an agent itself. An environment can be described as a situation in which an agent is present. The environment is where agent lives, operate and provide the agent with something to sense and act upon it. An environment is mostly said to be non-feministic.

Features of Environment

As per Russell and Norvig, an environment can have various features from the point of view of an agent:

Fully observable vs Partially Observable

Static vs Dynamic

Discrete vs Continuous

Deterministic vs Stochastic

Single-agent vs Multi-agent

Episodic vs sequential

Known vs Unknown

Accessible vs Inaccessible

1. **Fully observable vs Partially Observable**:

If an agent sensor can sense or access the complete state of an environment at each point in time then it is a fully observable environment, it is partially observable. For reference, Imagine a chess-playing agent. In this case, the agent can fully observe the state of the chessboard at all times. Its sensors (in this case, vision or the ability to access the board's state) provide complete information about the current position of all pieces. This is a fully observable environment because the agent has perfect information about the state of the world.

A fully observable environment is easy as there is no need to maintain the internal state to keep track of the history of the world. For reference, Consider a self-driving car navigating a busy city. While the car has sensors like cameras, lidar, and radar, it can't see everything at all times. Buildings, other vehicles, and pedestrians can obstruct its sensors. In this scenario, the car's environment is partially observable because it doesn't have complete and constant access to all relevant information. It needs to maintain an internal state and history to make informed decisions even when some information is temporarily unavailable.

An agent with no sensors in all environments then such an environment is called unobservable. For reference, think about an agent designed to predict earthquakes but placed in a sealed, windowless room with no sensors or access to external data. In this situation, the environment is unobservable because the agent has no way to gather information about the outside world. It can't sense any aspect of its environment, making it completely unobservable.

2. **Deterministic vs Stochastic:**

If an agent's current state and selected action can completely determine the next state of the environment, then such an environment is called a deterministic environment. For reference, Chess is a classic example of a deterministic environment. In chess, the rules are well-defined, and each move made by a player has a clear and predictable outcome based on those rules. If you move a pawn from one square to another, the resulting state of the chessboard is entirely determined by that action, as is your opponent's response. There's no randomness or uncertainty in the outcomes of chess moves because they follow strict rules. In a deterministic environment like chess, knowing the current state and the actions taken allows you to completely determine the next state.

A stochastic environment is random and cannot be determined completely by an agent. For reference, The stock market is an example of a stochastic environment. It's highly influenced by a multitude of unpredictable factors, including economic events, investor sentiment, and news. While there are patterns and trends, the exact behavior of stock prices is inherently random and cannot be completely determined by any individual or agent. Even with access to extensive data and analysis tools, stock market movements can exhibit a high degree of unpredictability. Random events and market sentiment play significant roles, introducing uncertainty.

In a deterministic, fully observable environment, an agent does not need to worry about uncertainty.

3**. Episodic vs Sequential:**

In an episodic environment, there is a series of one-shot actions, and only the current percept is required for the action. For example, Tic-Tac-Toe is a classic example of an episodic environment. In this game, two players take turns placing their symbols (X or O) on a 3x3 grid. Each move by a player is independent of previous moves, and the goal is to form a line of three symbols horizontally, vertically, or diagonally. The game consists of a series of one-shot actions where the current state of the board is the only thing that matters for the next move. There's no need for the players to remember past moves because they don't affect the current move. The game is self-contained and episodic.

However, in a Sequential environment, an agent requires memory of past actions to determine the next best actions. For example, Chess is an example of a sequential environment. Unlike Tic-Tac-Toe, chess is a complex game where the outcome of each move depends on a sequence of previous moves. In chess, players must consider the history of the game, as the current position of pieces, previous moves, and potential future moves all influence the best course of action. To play chess effectively, players need to maintain a memory of past actions, anticipate future moves, and plan their strategies accordingly. It's a sequential environment because the sequence of actions and the history of the game significantly impact decision-making.

4. **Single-agent vs Multi-agent**

If only one agent is involved in an environment, and operating by itself then such an environment is called a single-agent environment. For example, Solitaire is a classic example of a single-agent environment. When you play Solitaire, you're the only agent involved. You make all the decisions and actions to achieve a goal, which is to arrange a deck of cards in a specific way. There are no other agents or players interacting with you. It's a solitary game where the outcome depends solely on your decisions and moves. In this single-agent environment, the agent doesn't need to consider the actions or decisions of other entities.

However, if multiple agents are operating in an environment, then such an environment is called a multi-agent environment. For reference, A soccer match is an example of a multi-agent environment. In a soccer game, there are two teams, each consisting of multiple players (agents). These players work together to achieve common goals (scoring goals and preventing the opposing team from scoring). Each player has their own set of actions and decisions, and they interact with both their teammates and the opposing team. The outcome of the game depends on the coordinated actions and strategies of all the agents on the field. It's a multi-agent environment because there are multiple autonomous entities (players) interacting in a shared environment.

The agent design problems in the multi-agent environment are different from single-agent environments.

5. **Static vs Dynamic**:

If the environment can change itself while an agent is deliberating then such an environment is called a dynamic environment it is called a static environment.

Static environments are easy to deal with because an agent does not need to continue looking at the world while deciding on an action. For reference, A crossword puzzle is an example of a static environment. When you work on a crossword puzzle, the puzzle itself doesn't change while you're thinking about your next move. The arrangement of clues and empty squares remains constant throughout your problem-solving process. You can take your time to deliberate and find the best word to fill in each blank, and the puzzle's state remains unaltered during this process. It's a static environment because there are no changes in the puzzle based on your deliberations.

However, for a dynamic environment, agents need to keep looking at the world at each action. For reference, Taxi driving is an example of a dynamic environment. When you're driving a taxi, the environment is constantly changing. The road conditions, traffic, pedestrians, and other vehicles all contribute to the dynamic nature of this environment. As a taxi driver, you need to keep a constant watch on the road and adapt your actions in real time based on the changing circumstances. The environment can change rapidly, requiring your continuous attention and decision-making. It's a dynamic environment because it evolves while you're deliberating and taking action.

6. **Discrete vs Continuous**:

If in an environment, there are a finite number of percepts and actions that can be performed within it, then such an environment is called a discrete environment it is called a continuous environment.

Chess is an example of a discrete environment. In chess, there are a finite number of distinct chess pieces (e.g., pawns, rooks, knights) and a finite number of squares on the chessboard. The rules of chess define clear, discrete moves that a player can make. Each piece can be in a specific location on the board, and players take turns making individual, well-defined moves. The state of the chessboard is discrete and can be described by the positions of the pieces on the board.

Controlling a robotic arm to perform precise movements in a factory setting is an example of a continuous environment. In this context, the robot arm's position and orientation can exist along a continuous spectrum. There are virtually infinite possible positions and orientations for the robotic arm within its workspace. The control inputs to move the arm, such as adjusting joint angles or applying forces, can also vary continuously. Agents in this environment must operate within a continuous state and action space, and they need to make precise, continuous adjustments to achieve their goals.

7. **Known vs Unknown**

Known and unknown are not actually a feature of an environment, but it is an agent's state of knowledge to perform an action.

In a known environment, the results of all actions are known to the agent. While in an unknown environment, an agent needs to learn how it works in order to perform an action.

It is quite possible for a known environment to be partially observable and an Unknown environment to be fully observable.

The opening theory in chess can be considered as a known environment for experienced chess players. Chess has a vast body of knowledge regarding opening moves, strategies, and responses. Experienced players are familiar with established openings, and they have studied various sequences of moves and their outcomes. When they make their initial moves in a game, they have a good understanding of the potential consequences based on their knowledge of known openings.

Imagine a scenario where a rover or drone is sent to explore an alien planet with no prior knowledge or maps of the terrain. In this unknown environment, the agent (rover or drone) has to explore and learn about the terrain as it goes along. It doesn't have prior knowledge of the landscape, potential hazards, or valuable resources. The agent needs to use sensors and data it collects during exploration to build a map and understand how the terrain works. It operates in an unknown environment because the results and consequences of its actions are not initially known, and it must learn from its experiences.

8. **Accessible vs Inaccessible**

If an agent can obtain complete and accurate information about the state's environment, then such an environment is called an Accessible environment else it is called inaccessible.

For example, Imagine an empty room equipped with highly accurate temperature sensors. These sensors can provide real-time temperature measurements at any point within the room. An agent placed in this room can obtain complete and accurate information about the temperature at different locations. It can access this information at any time, allowing it to make decisions based on the precise temperature data. This environment is accessible because the agent can acquire complete and accurate information about the state of the room, specifically its temperature.

For example, Consider a scenario where a satellite in space is tasked with monitoring a specific event taking place on Earth, such as a natural disaster or a remote area's condition. While the satellite can capture images and data from space, it cannot access fine-grained information about the event's details. For example, it may see a forest fire occurring but cannot determine the exact temperature at specific locations within the fire or identify individual objects on the ground. The satellite's observations provide valuable data, but the environment it is monitoring (Earth) is vast and complex, making it impossible to access complete and detailed information about all aspects of the event. In this case, the Earth's surface is an inaccessible environment for obtaining fine-grained information about specific events.

What is an 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 performs this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.

The expert system is a part of AI, and the first ES was developed in the year 1970, which was the first successful approach of artificial intelligence. It solves the most complex issue as an expert by extracting the knowledge stored in its knowledge base. The system helps in decision making for compsex problems using both facts and heuristics like a human expert. It is called so because it contains the expert knowledge of a specific domain and can solve any complex problem of that particular domain. These systems are designed for a specific domain, such as medicine, science, etc.

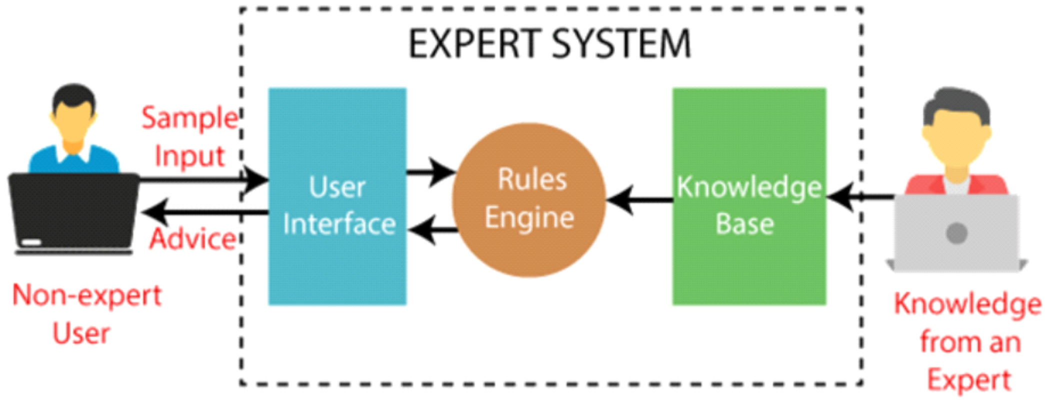
The performance of an expert system is based on the expert's knowledge stored in its knowledge base. The more knowledge stored in the KB, the more that system improves its performance. One of the common examples of an ES is a suggestion of spelling errors while typing in the Google search box.

Below is the block diagram that represents the working of an expert system:

Expert Systems in AI

Note: It is important to remember that an expert system is not used to replace the human experts; instead, it is used to assist the human in making a complex decision. These systems do not have human capabilities of thinking and work on the basis of the knowledge base of the particular domain.

Below are some popular examples of the Expert System:



DENDRAL: It was an artificial intelligence project that was made as a chemical analysis expert system. It was used in organic chemistry to detect unknown organic molecules with the help of their mass spectra and knowledge base of chemistry.

MYCIN: It was one of the earliest backward chaining expert systems that was designed to find the bacteria causing infections like bacteraemia and meningitis. It was also used for the recommendation of antibiotics and the diagnosis of blood clotting diseases.

PXDES: It is an expert system that is used to determine the type and level of lung cancer. To determine the disease, it takes a picture from the upper body, which looks like the shadow. This shadow identifies the type and degree of harm.

CaDeT: The CaDet expert system is a diagnostic support system that can detect cancer at early stages.

Characteristics of Expert System

High Performance: The expert system provides high performance for solving any type of complex problem of a specific domain with high efficiency and accuracy.

Understandable: It responds in a way that can be easily understandable by the user. It can take input in human language and provides the output in the same way.

Reliable: It is much reliable for generating an efficient and accurate output.

Highly responsive: ES provides the result for any complex query within a very short period of time.

Components of Expert System

An expert system mainly consists of three components:

User Interface

Inference Engine

Knowledge Base

Expert Systems in AI

1. User Interface

With the help of a user interface, the expert system interacts with the user, takes queries as an input in a readable format, and passes it to the inference engine. After getting the response from the inference engine, it displays the output to the user. In other words, it is an interface that helps a non-expert user to communicate with the expert system to find a solution.

2. Inference Engine(Rules of Engine)

The inference engine is known as the brain of the expert system as it is the main processing unit of the system. It applies inference rules to the knowledge base to derive a conclusion or deduce new information. It helps in deriving an error-free solution of queries asked by the user.

With the help of an inference engine, the system extracts the knowledge from the knowledge base.

There are two types of inference engine:

Deterministic Inference engine: The conclusions drawn from this type of inference engine are assumed to be true. It is based on facts and rules.

Probabilistic Inference engine: This type of inference engine contains uncertainty in conclusions, and based on the probability.

Inference engine uses the below modes to derive the solutions:

Forward Chaining:It starts from the known facts and rules, and applies the inference rules to add their conclusion to the known facts.

Backward Chaining: It is a backward reasoning method that starts from the goal and works backward to prove the known facts.

3. **Knowledge Base**

The knowledgebase is a type of storage that stores knowledge acquired from the different experts of the particular domain. It is considered as big storage of knowledge. The more the knowledge base, the more precise will be the Expert System.

It is similar to a database that contains information and rules of a particular domain or subject.

One can also view the knowledge base as collections of objects and their attributes. Such as a Lion is an object and its attributes are it is a mammal, it is not a domestic animal, etc.

Components of Knowledge Base

Factual Knowledge: The knowledge which is based on facts and accepted by knowledge engineers comes under factual knowledge.

Heuristic Knowledge: This knowledge is based on practice, the ability to guess, evaluation, and experiences.

Knowledge Representation: It is used to formalize the knowledge stored in the knowledge base using the If-else rules.

Knowledge Acquisitions: It is the process of extracting, organizing, and structuring the domain knowledge, specifying the rules to acquire the knowledge from various experts, and store that knowledge into the knowledge base.

Development of Expert System

Here, we will explain the working of an expert system by taking an example of MYCIN ES. Below are some steps to build an MYCIN:

Firstly, ES should be fed with expert knowledge. In the case of MYCIN, human experts specialized in the medical field of bacterial infection, provide information about the causes, symptoms, and other knowledge in that domain.

The KB of the MYCIN is updated successfully. In order to test it, the doctor provides a new problem to it. The problem is to identify the presence of the bacteria by inputting the details of a patient, including the symptoms, current condition, and medical history.

The ES will need a questionnaire to be filled by the patient to know the general information about the patient, such as gender, age, etc.

Now the system has collected all the information, so it will find the solution for the problem by applying if-then rules using the inference engine and using the facts stored within the KB.

In the end, it will provide a response to the patient by using the user interface.

Participants in the development of Expert System

There are three primary participants in the building of Expert System:

Expert: The success of an ES much depends on the knowledge provided by human experts. These experts are those persons who are specialized in that specific domain.

Knowledge Engineer: Knowledge engineer is the person who gathers the knowledge from the domain experts and then codifies that knowledge to the system according to the formalism.

End-User: This is a particular person or a group of people who may not be experts, and working on the expert system needs the solution or advice for his queries, which are complex.

Why Expert System? Expert Systems in AI

Before using any technology, we must have an idea about why to use that technology and hence the same for the ES. Although we have human experts in every field, then what is the need to develop a computer-based system. **So below are the points that are describing the need of the ES:**

No memory Limitations: It can store as much data as required and can memorize it at the time of its application. But for human experts, there are some limitations to memorize all things at every time.

High Efficiency: If the knowledge base is updated with the correct knowledge, then it provides a highly efficient output, which may not be possible for a human.

Expertise in a domain : There are lots of human experts in each domain, and they all have different skills, different experiences, and different skills, so it is not easy to get a final output for the query. But if we put the knowledge gained from human experts into the expert system, then it provides an efficient output by mixing all the facts and knowledge

Not affected by emotions: These systems are not affected by human emotions such as fatigue, anger, depression, anxiety, etc.. Hence the performance remains constant.

High security: These systems provide high security to resolve any query.

Considers all the facts: To respond to any query, it checks and considers all the available facts and provides the result accordingly. But it is possible that a human expert may not consider some facts due to any reason.

Regular updates improve the performance: If there is an issue in the result provided by the expert systems, we can improve the performance of the system by updating the knowledge base.

Capabilities of the Expert System

Below are some capabilities of an Expert System:

Advising: It is capable of advising the human being for the query of any domain from the particular ES.

Provide decision-making capabilities: It provides the capability of decision making in any domain, such as for making any financial decision, decisions in medical science, etc.

Demonstrate a device: It is capable of demonstrating any new products such as its features, specifications, how to use that product, etc.

Problem-solving: It has problem-solving capabilities.

Explaining a problem: It is also capable of providing a detailed description of an input problem.

Interpreting the input: It is capable of interpreting the input given by the user.

Predicting results: It can be used for the prediction of a result.

Diagnosis: An ES designed for the medical field is capable of diagnosing a disease without using multiple components as it already contains various inbuilt medical tools.

Advantages of Expert System

- These systems are highly reproducible.
- They can be used for risky places where the human presence is not safe.
- Error possibilities are less if the KB contains correct knowledge.
- The performance of these systems remains steady as it is not affected by emotions, tension, or fatigue.
- They provide a very high speed to respond to a particular query.

Limitations of Expert System

- The response of the expert system may get wrong if the knowledge base contains the wrong information.
- Like a human being, it cannot produce a creative output for different scenarios.
- Its maintenance and development costs are very high.
- Knowledge acquisition for designing is much difficult.
- For each domain, we require a specific ES, which is one of the big limitations.

It cannot learn from itself and hence requires manual updates.

Applications of Expert System

- In designing and manufacturing domain
- It can be broadly used for designing and manufacturing physical devices such as camera lenses and automobiles.
- In the knowledge domain
- These systems are primarily used for publishing the relevant knowledge to the

users. The two popular ES used for this domain is an advisor and a tax advisor.

- In the finance domain
- In the finance industries, it is used to detect any type of possible fraud, suspicious activity, and advise bankers that if they should provide loans for business or not.
- In the diagnosis and troubleshooting of devices
- In medical diagnosis, the ES system is used, and it was the first area where these systems were used.
- Planning and Scheduling
- The expert systems can also be used for planning and scheduling some particular tasks for achieving the goal of that task.

The stages of expert system development in artificial intelligence (AI) are:

Identification: Define the problem and goals of the system

Conceptualization: Analyze and diagram the key concepts and relationships

Formalization: Connect the problem to a technical solution and select techniques

Implementation: Program the formalized concepts into a prototype system

Testing: Test the prototype to identify weaknesses and ensure it performs like a human expert