AI-Module-1-Important-Topics

For more notes visit

https://rtpnotes.vercel.app

Playlist to refer

https://youtube.com/playlist?list=PLnzz0gSUYIN3mo-LB-l2Vfadz6HiPi8Po&feature=shared

1. History of AI

Here’s the timeline formatted as a table for a more structured presentation:

Year	Event
1943	**Early Beginnings**: McCulloch & Pitts: Boolean circuit model of the brain.
1950	**Turing's Vision**: Turing's "Computing Machinery and Intelligence."
1956	**Birth of AI**: Dartmouth meeting; The term "Artificial Intelligence" is adopted.
1950s	**Initial Promise**: Early AI programs, including:
	- Samuel's Checkers Program
	- Newell & Simon's Logic Theorist
1955-1965	**“Great Enthusiasm”**:
	- Newell & Simon: General Problem Solver (GPS)
	- Gelertner: Geometry Theorem Prover
	- McCarthy: Invention of LISP
1966-1973	**Reality Dawns**: Realization that many AI problems are intractable and limitations of existing neural network methods.

Year	Event
1969-1985	**Adding Domain Knowledge**: Development of knowledge-based systems and rule-based expert systems, such as:
	- DENDRAL
	- MYCIN
	- However, these systems were brittle and did not scale well in practice.
1986	**Rise of Machine Learning**: Neural networks return to popularity; major advances in machine learning algorithms and applications.
1990	**Role of Uncertainty**: Introduction of Bayesian networks as a knowledge representation framework.
1995	**AI as Science**: Integration of learning, reasoning, and knowledge representation.
1995-2011	**Intelligent Agents**: Development of intelligent agents as a major focus in AI.
2011-Present	**Deep Learning, Big Data, and Artificial General Intelligence**:
	- **2018 Achievements**:
	- Alibaba's language processing AI outperforms top humans in a Stanford University test (82.44 vs 82.304).
	- Google Duplex: AI assistant can book appointments over the phone with a "nearly flawless" imitation of human speech.

2. AI-Levels, Goals and Types

Artificial Intelligence (AI) is the field of computer science focused on creating machines that can think and act intelligently, similar to how humans do.

Levels of AI:

1. **Narrow AI**: **Machines perform specific tasks better** than humans, such as facial recognition or playing chess. Current AI research is focused here.
2. **General AI**: A machine that can perform **any intellectual task at the same level** as a human, though this has not been achieved yet.
3. **Strong AI**: Machines that **surpass human intelligence in many areas,** being able to perform tasks better than humans across a wide range of domains.

Goals of Artificial Intelligence (AI):

1. **Create Expert Systems**: Develop systems that can perform tasks intelligently, learning, explaining, and advising users, much like a human expert.
2. **Implement Human Intelligence in Machines**: Build machines that can think, learn, understand, and behave like humans.

Types of AI:

- **General-purpose AI**: Like the robots seen in science fiction, which can perform a wide range of tasks. However, replicating the complexity of the human brain is extremely difficult, and we still don't fully understand how it works.
- **Special-purpose AI**: Focuses on specific tasks, which is more achievable. Examples include AI used in chess or poker programs, logistics planning, voice recognition, web search, data mining, medical diagnosis, and self-driving cars.

3. Structure of an Agent

Structure of an AI Agent

- The structure of an AI agent explains how an agent works by combining two key parts: the architecture and the agent program.

1. Architecture:

- This is the **hardware** or **platform** on which the agent runs. It could be a physical machine (like a robot) or a computer system. Think of it as the "body" of the agent that allows it to function.

2. Agent Function:

- The **agent function** is like the brain behind the agent's actions. It takes the **percept** (what the agent senses from the environment) and decides what action to take in response. It's represented as:
  - f: P -> A* (where "P" is the percept and "A" is the action).

3. Agent Program:

- The **agent program** is the actual **code or software** that makes the agent function work. It runs on the architecture and translates the agent's observations into actions.

Formula:

- **Agent = Architecture + Agent Program**: An AI agent is a combination of the physical structure (architecture) and the intelligent software (agent program).

4. Intelligent and Rational Agent

Intelligent Agents

An **intelligent agent** is like a smart, independent system that senses its surroundings, makes decisions, and takes actions to achieve a goal. It uses **sensors** to gather information and **actuators** to interact with the environment.

- **Autonomous**: It operates on its own without constant human guidance.
- **Goal-oriented**: It works toward specific goals, like keeping a room at the right temperature or navigating a car safely.
- **Learning**: An intelligent agent can learn from its environment and improve its decisions over time.

Example: A thermostat
A thermostat is an intelligent agent that senses the room temperature (using sensors) and turns the heater on or off (using actuators) to maintain the desired temperature.

Rational Agents

A **rational agent** is an agent that makes decisions with a clear goal in mind and tries to achieve the best possible outcome based on what it knows.

- **Clear Preferences**: A rational agent knows what it prefers or what it is trying to achieve (its goal).
- **Models Uncertainty**: It understands that it might not know everything, so it makes decisions based on the best possible information.
- **Maximizes Performance**: The agent aims to take actions that lead to the best results, according to a performance measure. For example, in a game, this could mean winning,

and in real life, it could mean solving a problem efficiently.

- **Acts Rationally**: It always tries to take the best possible action to reach its goal, given the situation it is in.

5. PEAS description

PEAS is a framework that helps describe how an AI agent works by breaking it down into four parts:

- **P (Performance Measure):** This is how we judge the success of the AI's actions. For example, in an automated taxi, performance could be measured by how safely and efficiently it gets passengers to their destination.
- **E (Environment):** This is the world the AI operates in. For a taxi, this would be the city streets, weather, traffic, and pedestrians.
- **A (Actuators):** These are the parts of the AI that allow it to take action. In a taxi, the actuators would be the steering wheel, brakes, and accelerator.
- **S (Sensors):** These help the AI gather information about the environment. In the taxi's case, sensors could include cameras, GPS, and radar.

Example 1:

- Task of designing a self-driving car
- Performance Measure
  - Safe, fast, legal, comfortable trip
- Environment
  - Roads, other traffic, pedestrials
- Actuators
  - Steering wheel, accelerator, brake, signal, horn
- Sensors
  - Cameras, Speedometers, GPS, Odometer, engine sensors etc

Example 2

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Figure 2.5 Examples of agent types and their PEAS descriptions.

6. Properties of task environment

Task Environment Types

- Fully observable vs partially observable
- Single agent vs Multi agent
- Deterministic vs stochastic
- Episodic vs sequential
- Static vs dynamic
- Discrete vs continous
- Known vs unknown

Here's a simplified version of the concepts:

Fully Observable vs Partially Observable

- **Fully Observable**: If an agent's sensors provide complete information about the environment at all times, it’s fully observable. The agent doesn't need to keep track of anything internally since all relevant details are available.
- **Partially Observable**: If the sensors are inaccurate, noisy, or miss some details, the environment is partially observable. This could be because parts of the environment aren't detected.
- **Unobservable**: If there are no sensors, the environment is unobservable.

Single Agent vs Multi Agent

- **Single Agent**: When there is only one agent in an environment, it's called a single-agent environment.
- **Multi-Agent**: When multiple agents are involved, it's a multi-agent environment. For example, chess is competitive, while a taxi-driving scenario can be both competitive (for parking) and cooperative (avoiding collisions).

Deterministic vs Stochastic

- **Deterministic**: If the agent’s actions completely determine the next state of the environment, it’s deterministic. There's no uncertainty.
- **Stochastic**: If the outcome is uncertain and has probabilities, it's stochastic. If the environment is partially observable, it can seem stochastic even if it’s not.

Episodic vs Sequential

- **Episodic**: The agent's experience is divided into episodes where each one is independent of the others. The agent makes decisions based on the current situation only.
- **Sequential**: Actions have consequences for future decisions. In games like chess or taxi driving, short-term choices affect long-term outcomes.

Static vs Dynamic

- **Static**: The environment doesn’t change while the agent is thinking. Time passing doesn’t affect the environment.

- **Dynamic**: The environment can change while the agent is thinking, so the agent has to make decisions in real-time.
- **Semi-Dynamic**: The environment doesn't change, but the agent's performance may vary over time (e.g., in chess with a clock).

Discrete vs Continuous

- **Discrete**: The environment has a limited number of distinct states, actions, and percepts. Chess is an example of a discrete environment.
- **Continuous**: The environment is smooth, with many possible states and actions, like in taxi driving where the position and speed can change continuously.

Known vs Unknown

- **Known**: The outcomes of actions are known, so the agent can make informed decisions.
- **Unknown**: The agent has to explore and learn how the environment works to make good decisions. Even in a fully observable environment, if the agent doesn't know the rules (like in a new game), it's unknown.

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Figure 2.6 Examples of task environments and their characteristics.

7. Types of Agents

What is an Agent?

- Think of an "agent" as a robot or a computer program that can "see" what's happening around it and "do" things to change or respond to its surroundings.
    - **Operates in an environment**:.
    - **Perceives its environment through sensors**
    - **Acts upon its environment through actuators/effectors**
    - **Has goals**
- In simpler terms, an agent is like a smart helper that watches, thinks, and acts to accomplish something in the world it operates in.

Actuators and Effectors

Actuators and **effectors** are devices or components that allow an agent (like a robot or a system) to take action and interact with its environment.

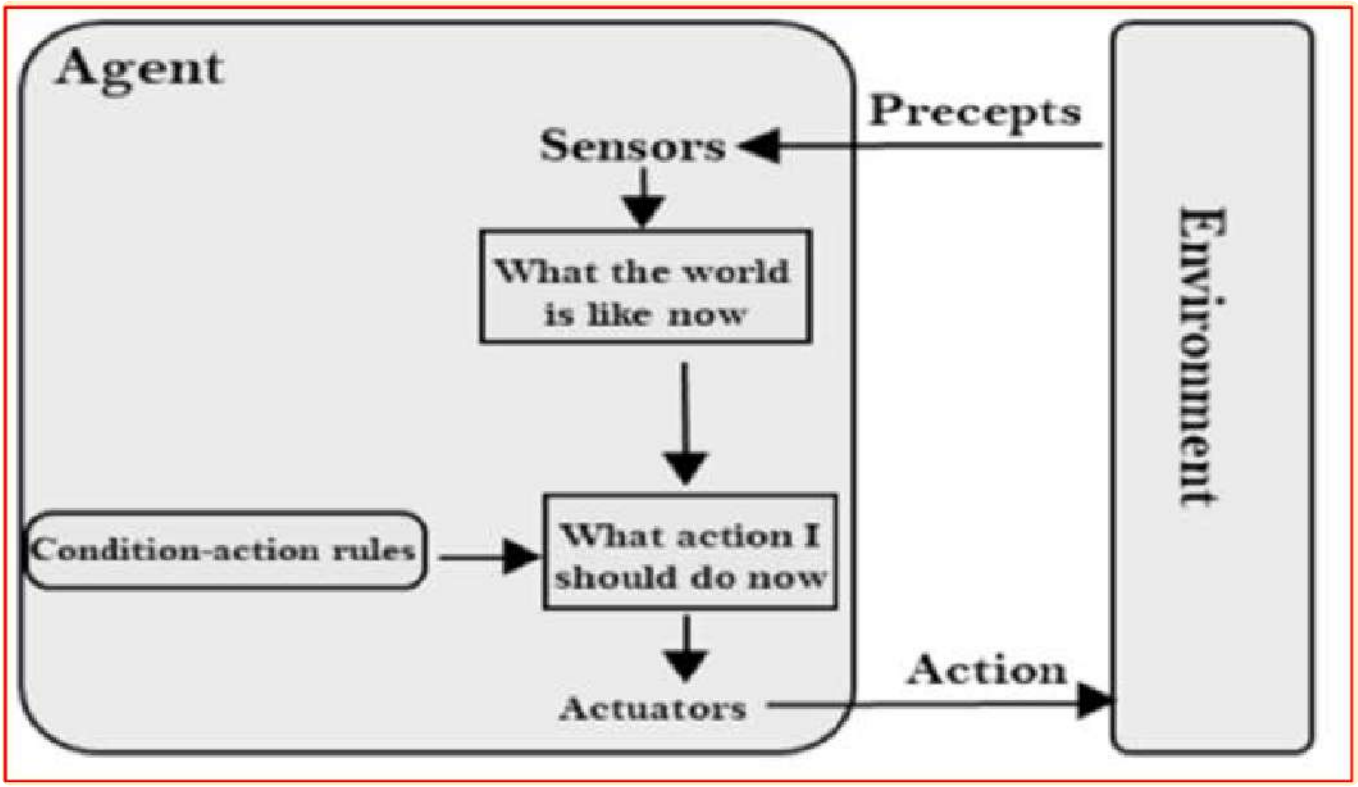
- **Actuators**: These are the mechanisms that physically make something happen. In robots, for example, actuators are like muscles—they control movements such as turning wheels, moving arms, or opening grippers.
- **Effectors**: These are the parts of the agent that cause the actual change in the environment. In a robot, effectors might be things like wheels, robotic arms, or grippers. **The actuators control the effectors**. For example, an actuator moves a robot's arm, and the **arm** (the effector) is what interacts with the object or the environment.

1. Simple Reflex Agent

A **Simple Reflex Agent** is a type of artificial agent that works by reacting to its current situation (called the "percept") without considering any history or future consequences.

- **Choose actions only based on the current percept**: The agent decides what to do just by looking at the current situation it's in, without worrying about past events or future outcomes.
- **Rational only if a correct decision is made based on the current percept**: The agent can be called "smart" or "rational" only if it can make the right decision based solely on what it currently observes.
- **Their environment is completely observable**: The agent can see or sense everything it needs to know about its environment right now.
- **Condition-Action Rule**: The agent follows simple rules that connect what it sees (condition) to what it should do (action). For example, "If the light is red (condition), then

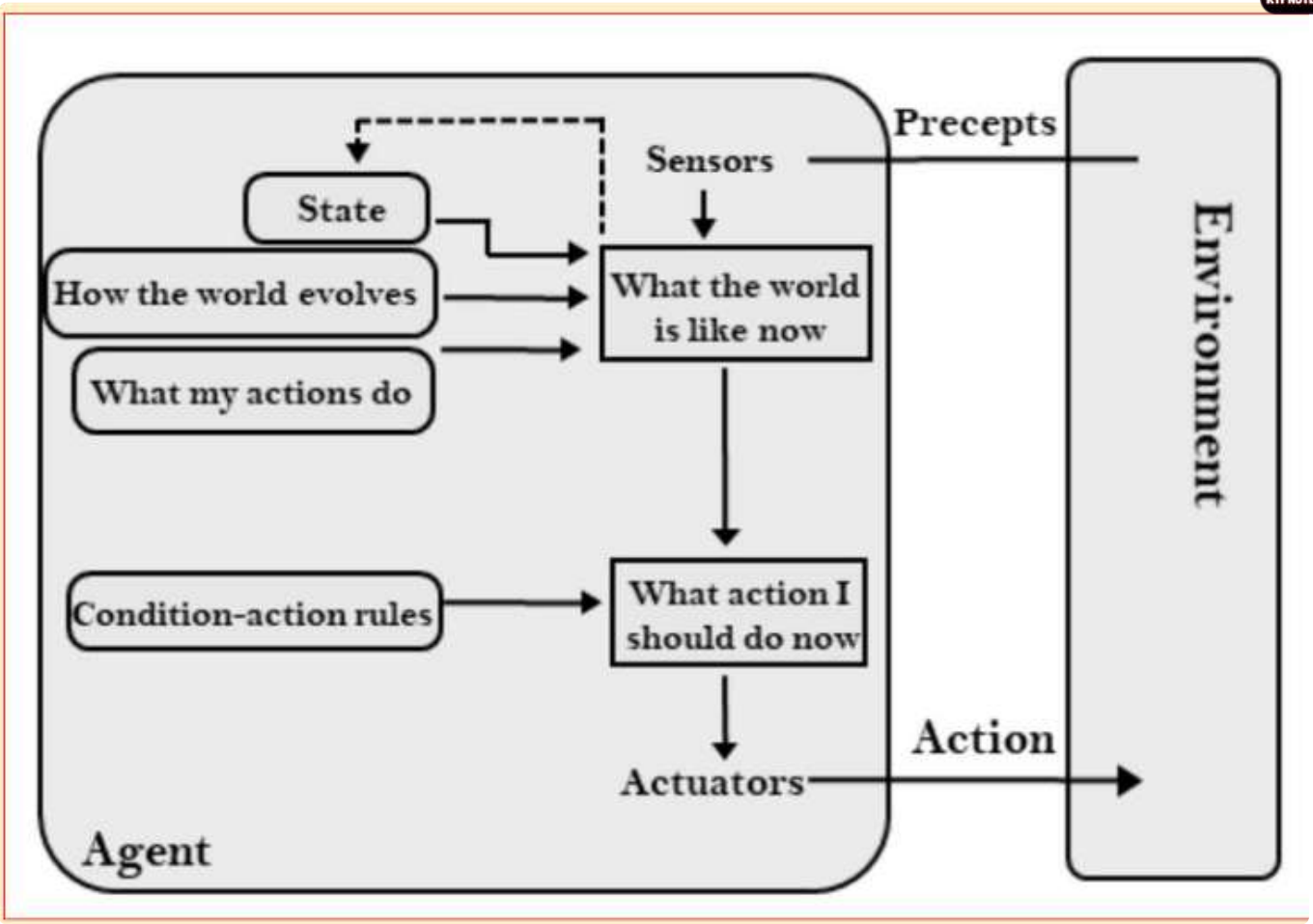
stop (action)."



2. Model-based reflex agent

A Model-based Reflex Agent is a more advanced type of agent that can make better decisions by understanding the world and keeping track of changes

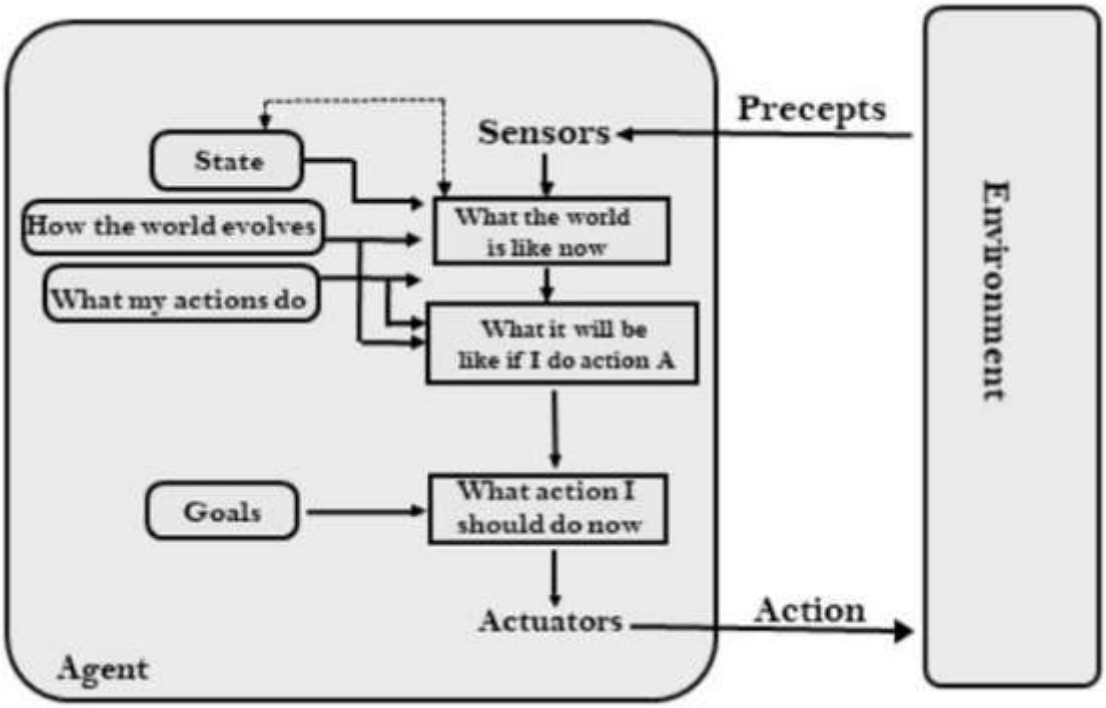
- **A model of the world to choose their actions**: Instead of just reacting to what it sees in the moment, this agent builds a mental picture (or "model") of how the world works to guide its actions. It also keeps track of what's happening, even if it can't see everything at once.
- **Model**: This is the agent's understanding of how things usually happen in the world. For example, it knows that if it drops something, it will fall.
- **Internal State**: The agent remembers things that it can’t currently see. This memory is built up from everything it has observed so far (called "percept history"). For example, if the agent has walked into another room, it remembers where it came from, even if it can’t see it now.
- **Updating the state**: To keep this memory accurate, the agent needs to:
  - Understand how the world changes over time.
  - Know how its own actions will change the world around it.



3. Goal-based agents

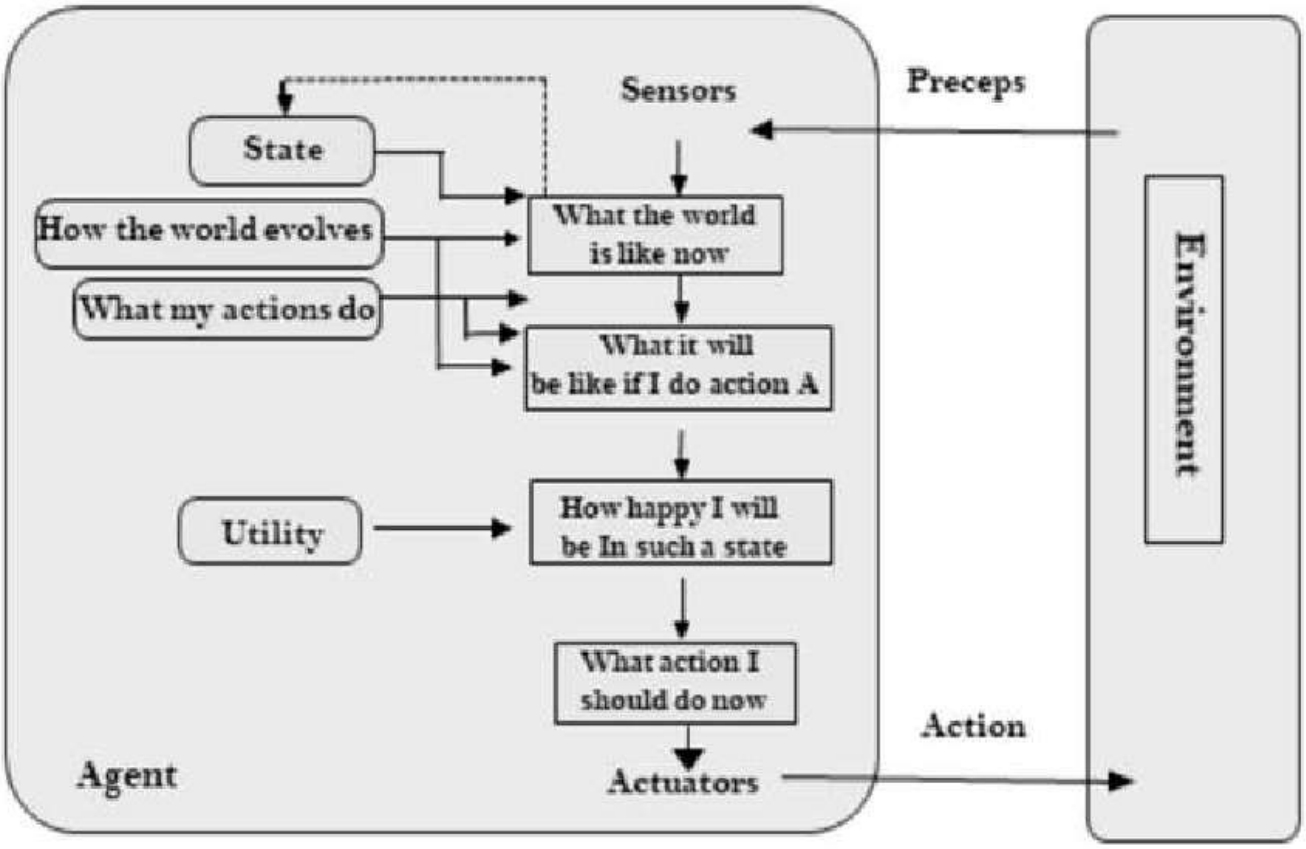
- Goal-based agents are a type of AI that make decisions to achieve specific goals. Unlike reflex agents, they don't just react to situations; they use knowledge to plan their actions.
- This makes them more flexible, as you can change their behavior by adjusting their goals.

- A "goal" simply describes the outcome the agent is trying to achieve.



4. Utility Based Agent

- **Utility-based agents** decide what to do by ranking different outcomes, or states, based on how useful or beneficial (utility) they are.
- Unlike goal-based agents, they don’t just aim for one goal but **balance multiple goals that might conflict**.
- When some goals can't be fully achieved, utility-based agents choose the action that gives the **best possible result**.
- They also factor in uncertainty, considering both the chances of success and how important each goal is.



5. Learning agent

A learning agent is an AI that improves its performance by learning from past experiences. It starts with basic knowledge and then adapts automatically as it learns. It has four main parts:

1. **Learning element**: This part learns from the environment and makes improvements.
2. **Critic**: It provides feedback to the learning element, showing how well the agent is doing compared to a set standard.
3. **Performance element**: This part chooses the actions the agent will take.
4. **Problem generator**: It suggests new actions that help the agent explore and gain more useful experiences.