ARTIFICIAL INTELLIGENCE AND SOFT COMPUTING (CS3101)

Sem: 5th (Section D)

Instructor: Bali Devi

Intelligent Agents in AI

- In the context of AI, intelligent agents are entities that are designed to perceive their environment, reason about it, and take actions to achieve specific goals.

- These agents use artificial intelligence techniques to mimic or exhibit intelligent behaviour.

Agent v/s Software Program

- Agents are autonomous, decision-making entities with perception capabilities, often using AI, while software programs follow predetermined instructions for specific tasks.

- All agents are software programs, but not all software programs are agents.

- An agent is a specialized type of software program that exhibits autonomous behaviour, typically involving perception, reasoning, and decision-making.

Agent v/s Software Program

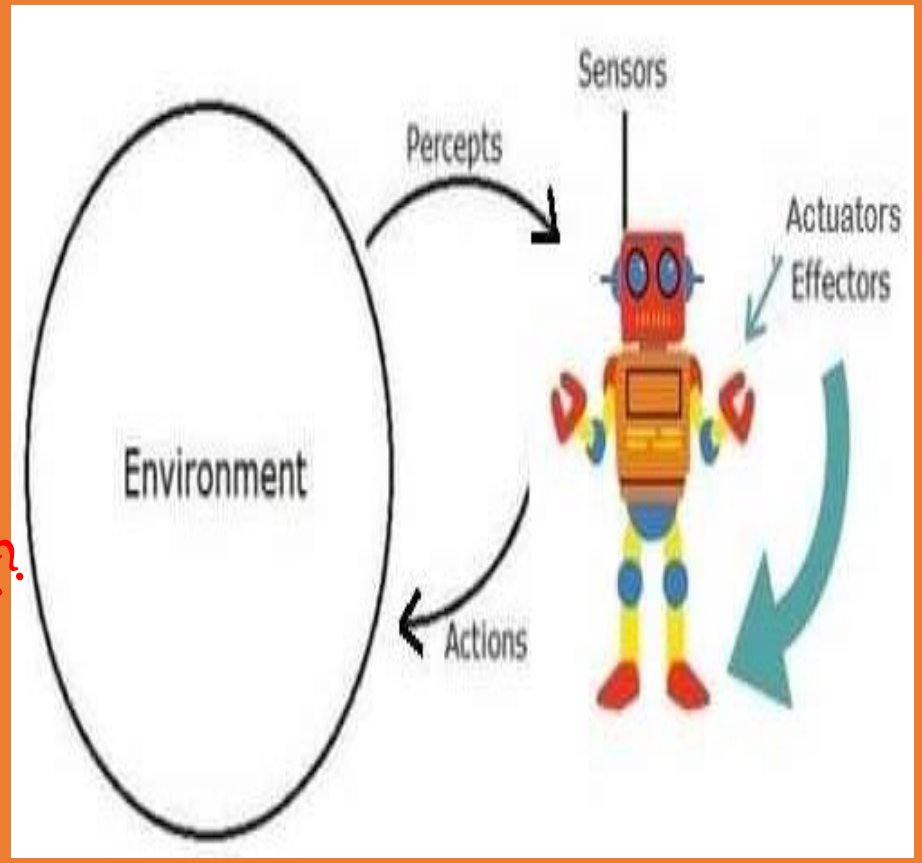
Characteristic	Agent	Software Program
Autonomy	Exhibits autonomy, acting independently	Typically follows a predetermined sequence of instructions
Perception	Capable of perceiving its environment	May not necessarily have sensing capabilities
Decision-Making	Makes decisions based on perception and goals	Follows predefined logic or algorithms
Learning	May incorporate learning mechanisms	May lack adaptive or learning capabilities
Complexity	Can be complex, incorporating AI techniques	Can vary in complexity, from simple to complex
Purpose	Designed to achieve specific goals or tasks	Developed for a particular function or task
Adaptability	Can adapt to changing conditions	May require manual updates for changes
Interaction	Can interact with other agents or systems	May or may not interact with external entities
Examples	Intelligent agents in robotics, virtual assistants	Text editors, calculators, basic programs

Agents

- An AI system is composed of an agent and its environment.
- The agents act in their environment. The environment may contain other agents.
- An agent is anything that can be viewed as:
  - Perceiving its environment through **sensors** and
  - Acting upon that environment through **actuators.**
- **We need to build intelligent agents that work in an environment.**
- We can define in simple terms, Agent as game and Environment as ground.

Agents	Environments
Robot	Room
Chatbot	Chatting
Vehicle	Road
Program	Data & Rules
Machine	Working Field

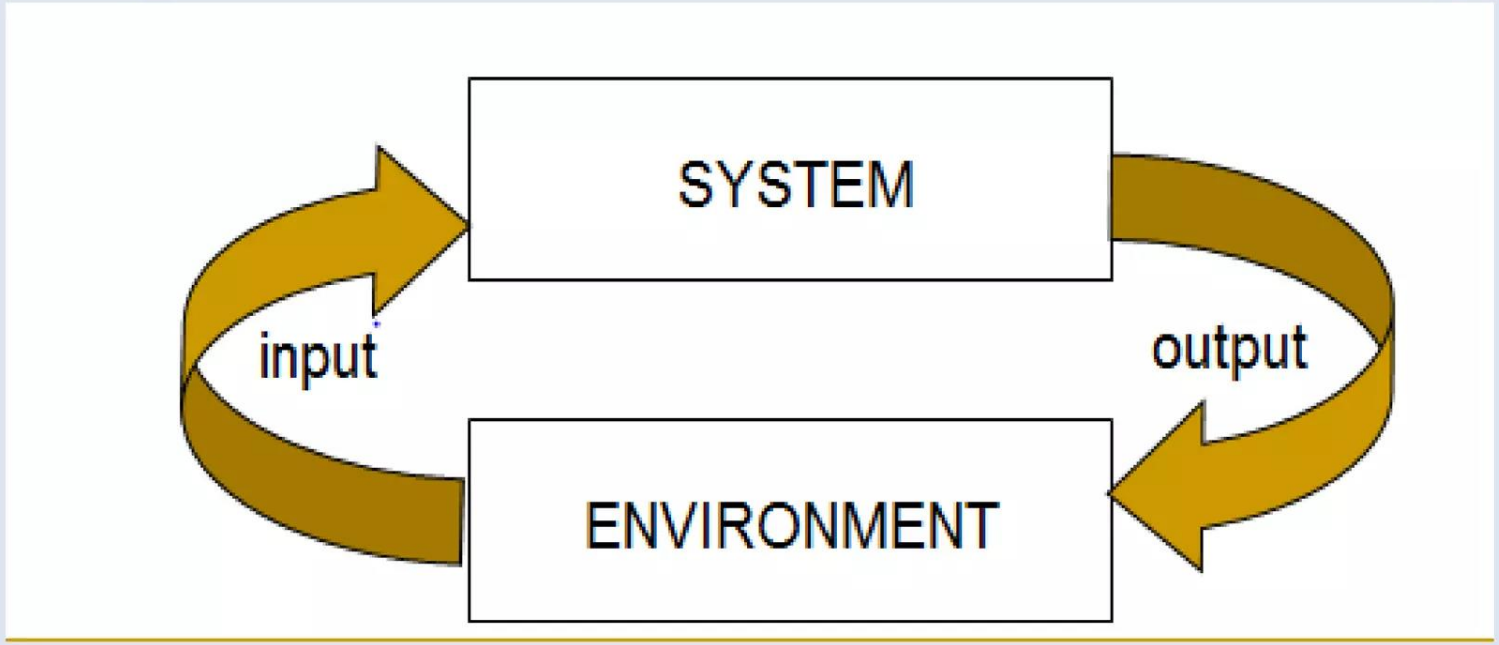
Human are Agents ???



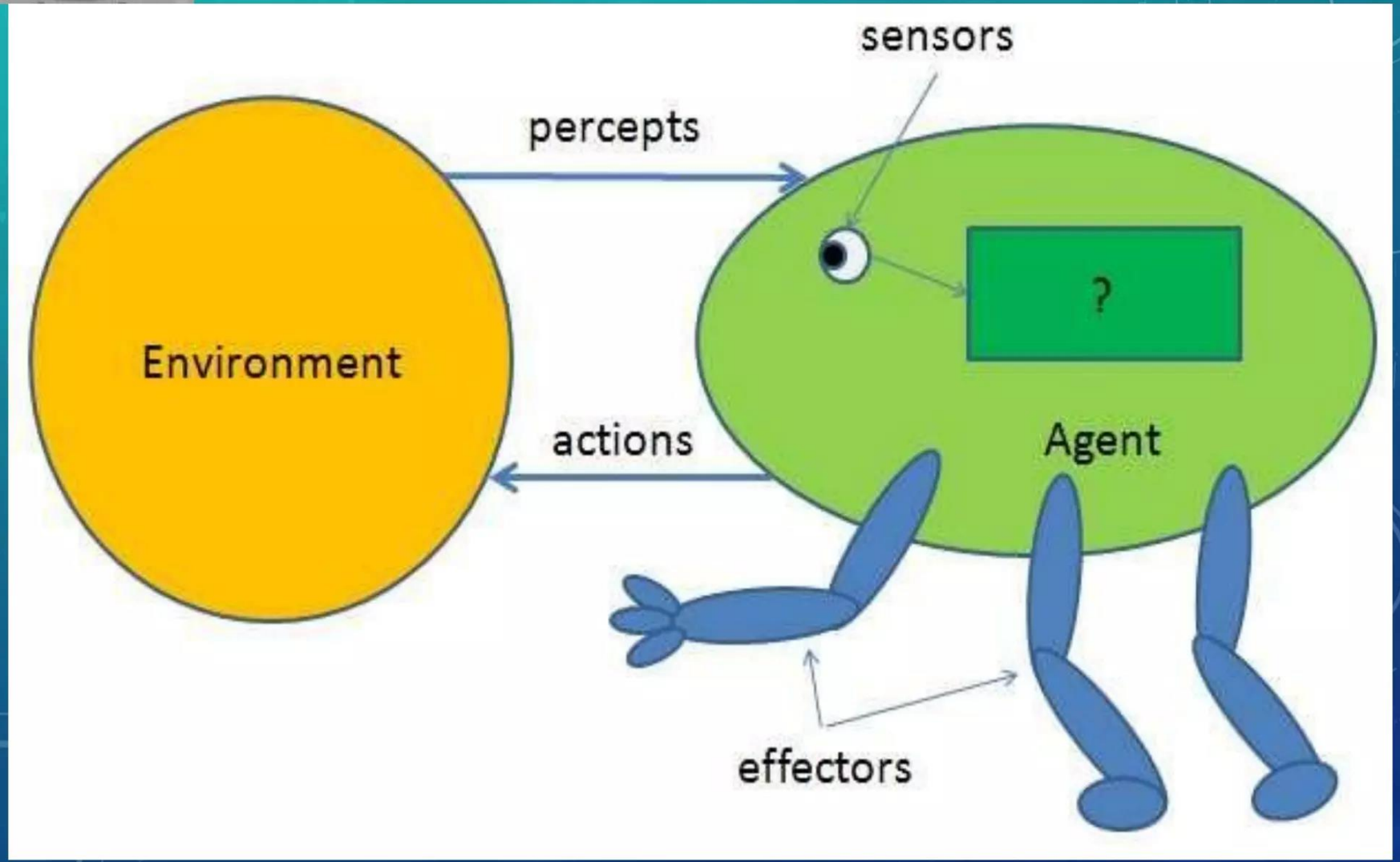
Agents

- Operate in an environment.
- Perceives and acts upon it's environment through actuators/sensors and have its goals.

.



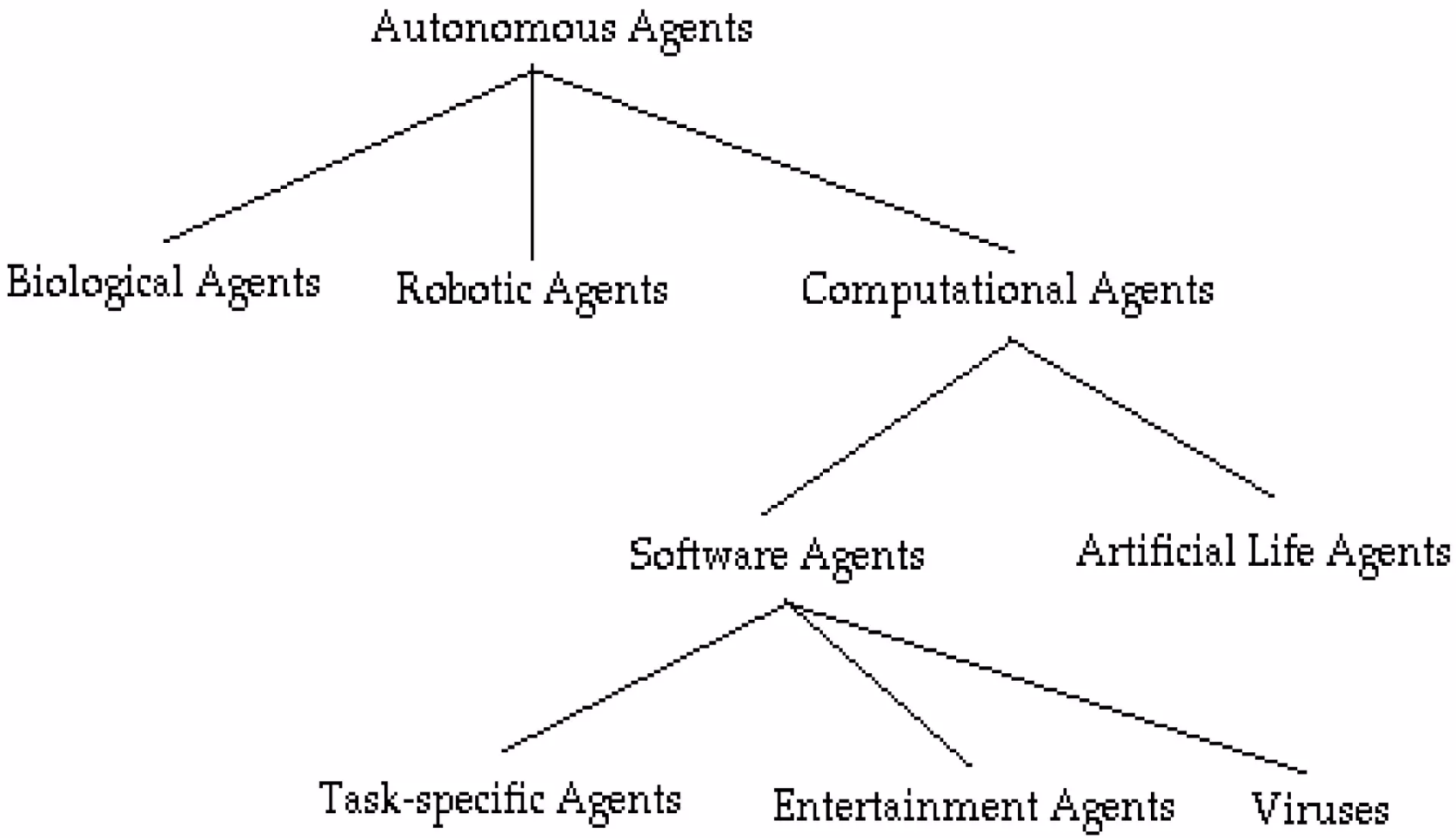
Agent and Environment



Sensors & Effectors

- An agent Perceives its environment through sensors.
- The complete set of inputs at a given time is called percept.
- The current percept, or a sequence of percepts can influence the actions of an agent.
- It can change the environment through effectors.
- An operation involving an actuator is called an action ,which can be grouped in to action sequences.

Agents Classification



Examples of agents

- **Humans**

  eyes, ears, skin, taste buds, etc. for Sensors.
  hands, fingers, legs, mouth for effectors.

- **Robots**

  camera, infrared, bumper, etc. for sensors.
  grippers, wheels, lights, speakers, effectors.

Components of Intelligent Agents

- **Perception (Sensors):** Intelligent agents are equipped with sensors or mechanisms to perceive information from their environment. These sensors could include cameras, microphones, temperature sensors, or other input devices depending on the nature of the agent and its application.

- **Knowledge Base:** Agents maintain a knowledge base that represents their understanding of the world. This knowledge can be pre-programmed or acquired through learning from data and experiences.

- **Reasoning:** Intelligent agents use reasoning mechanisms to interpret the information they receive from sensors and their knowledge base. This involves processing data, making inferences, and deriving conclusions.

- **Decision-Making:** Based on their reasoning, agents make decisions to achieve specific goals. Decision-making involves selecting appropriate actions from a range of possibilities, taking into account the current state of the environment and the agent's goals.

Components of Intelligent Agents

- **Actuators (Effectors):** Intelligent agents have actuators or effectors that allow them to interact with the environment and execute actions. These can include motors, speakers, displays, or other output devices.

- **Autonomy:** Intelligent agents operate autonomously, meaning they can function without constant human intervention. Autonomy enables them to adapt to changing conditions and make decisions independently.

- **Learning:** Many intelligent agents are designed to learn from experience or data. Machine learning algorithms allow agents to improve their performance over time by adjusting their behavior based on feedback and new information.

- **Communication:** Intelligent agents may communicate with other agents, humans, or external systems to exchange information, coordinate activities, or achieve common goals.

Examples of intelligent agents in AI include:

- **Virtual Assistants:** Agents like Siri, Google Assistant, or Amazon Alexa that understand natural language, respond to user queries, and perform tasks.

- **Autonomous Vehicles:** Agents that navigate and control vehicles without human intervention, such as self-driving cars or drones.

- **Game-Playing Agents:** Agents that play games like chess, Go, or video games, making decisions to optimize their chances of winning.

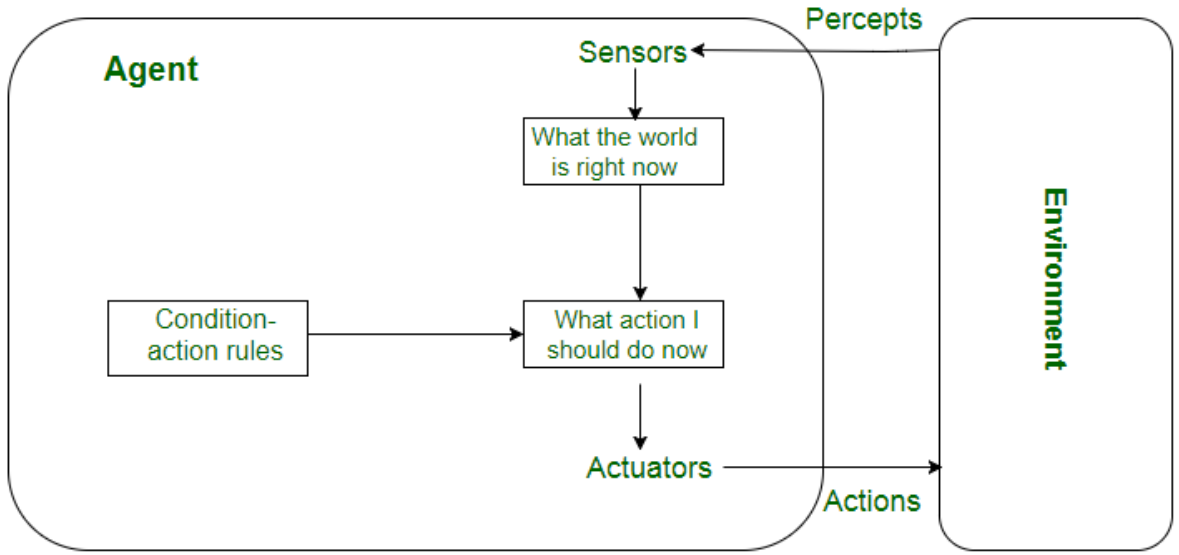
- **Robotics:** Physical robots that use sensors, actuators, and AI algorithms to perform tasks in the real world.

Classification of Agents

- **Simple Reflex Agents**
- **Model-Based Reflex Agents**
- **Goal-Based Agents:**
- **Utility-Based Agents:**

Simple Reflex Agents:

- These agents make decisions based solely on the current percept (environmental input) without considering the history of past percepts. They are reactive and lack memory.



Problems with Simple reflex agents are :

- Very limited intelligence.
- No knowledge of non-perceptual parts of the state.
- Usually too big to generate and store.
- If there occurs any change in the environment, then the collection of rules needs to be updated.

Case study

- **Simple Reflex Agent: Traffic Light Controller**

Case Study:

A simple reflex agent is used to control traffic lights at an intersection. The agent makes decisions based on the current state (percept) of the traffic, without considering the history.

Scenario:

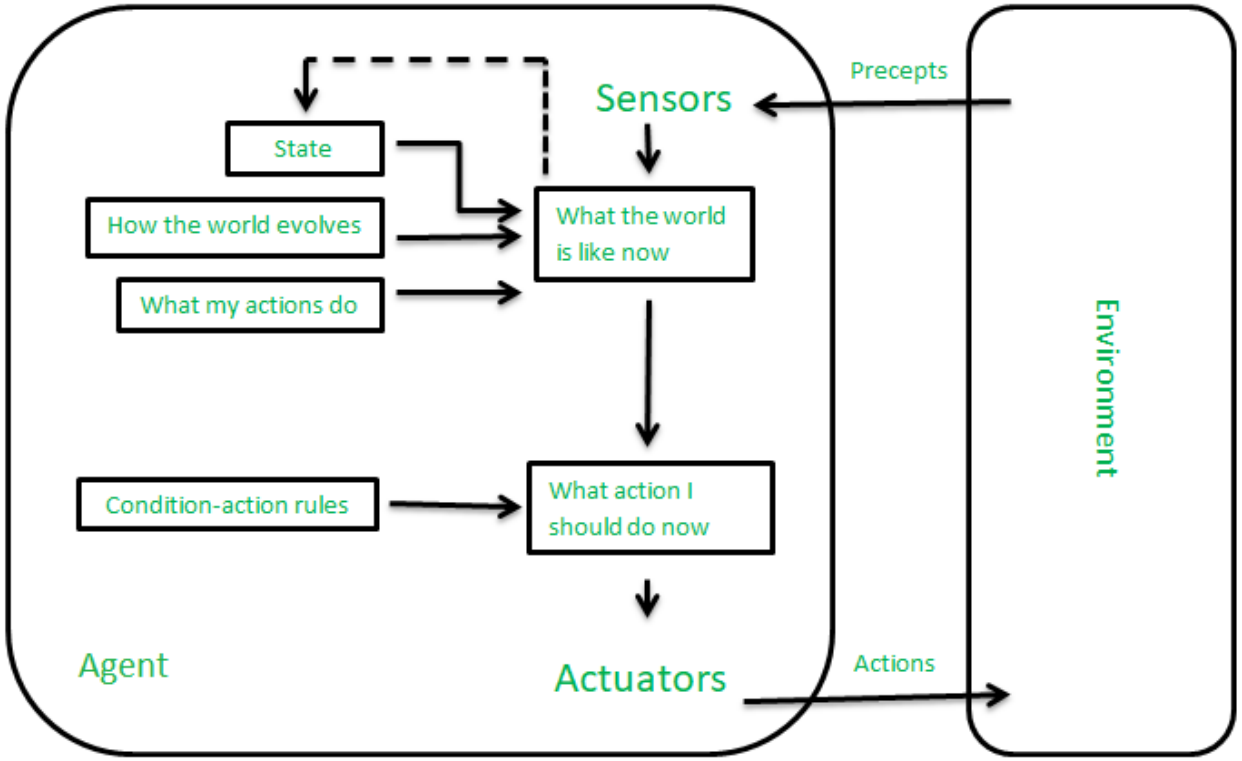
If the sensors detect heavy traffic on one road, the traffic light for that road stays green for a longer duration to allow vehicles to clear the congestion.

Model-Based Reflex Agents

- These agents maintain an internal model of the world, allowing them to consider the history of percepts. They use this model to make more informed decisions.

Updating the state requires information about:

- How the world evolves independently from the agent?
- How do the agent's actions affect the world?



Case study

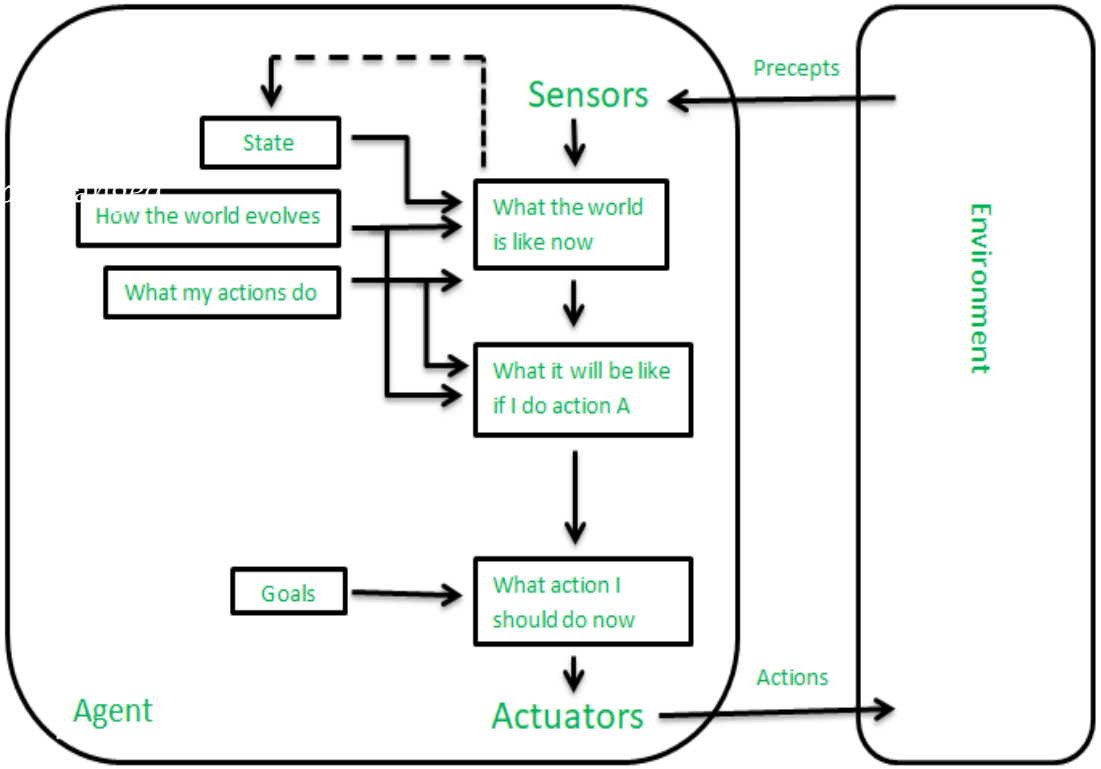
- **Model-Based Reflex Agent: Chess Playing Agent**
- **Case Study:**
- A model-based reflex agent plays chess. It maintains an internal model of the chessboard to make decisions based on the current state and history of moves.

- **Scenario:**
- The agent considers the positions of all pieces on the board and uses a model to determine the best move based on learned strategies and past game data.

Goal-Based Agents

- Goal-based agents have specific goals they aim to achieve. They take actions that move them closer to these goals, considering the current state and available actions.

- They usually require search and planning.
- The goal-based agent's behavior can easily be changec



Case study

- **Goal-Based Agent: Personal Assistant App**
- **Case Study:**
- A goal-based agent in the form of a personal assistant app helps users manage tasks and appointments.

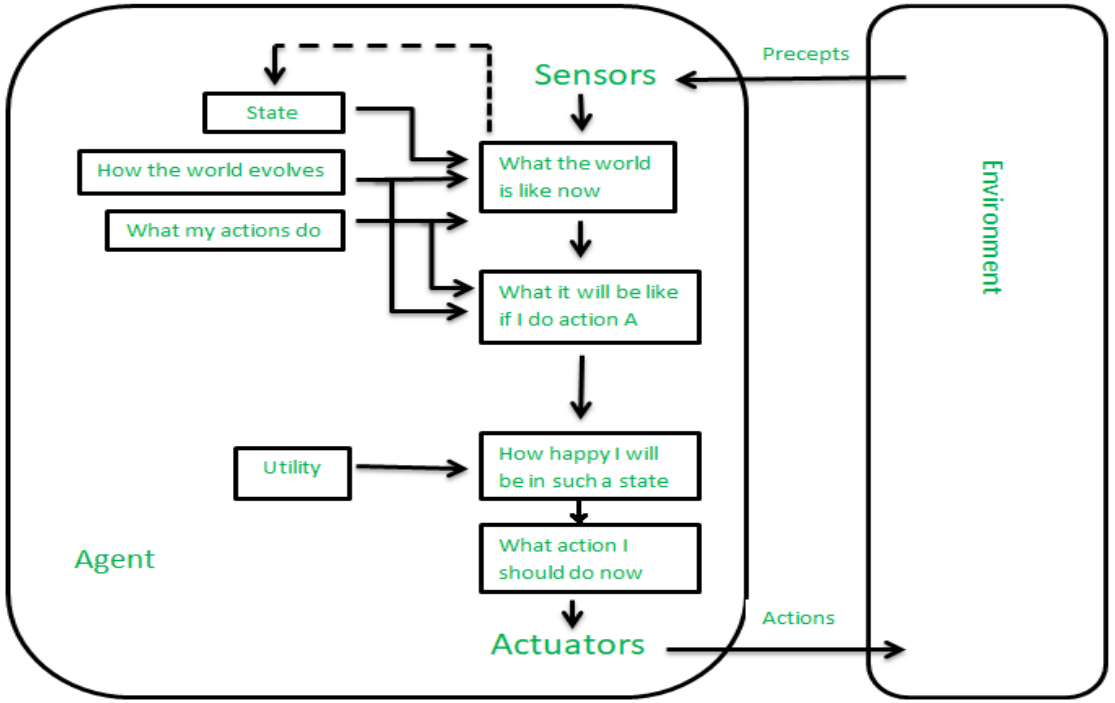
- **Scenario:**
- The user sets a goal of scheduling a meeting. The agent analyzes the user's calendar, considers the goal, and suggests available time slots for the meeting.

Utility-Based Agents:

- Utility-based agents evaluate different actions based on a utility or preference function. They choose actions that maximize expected utility, considering both goals and the desirability of outcomes.

Utility describes how "happy" the agent is:

- Because of the uncertainty in the world, a utility agent chooses the action that maximizes the expected utility.
- A utility function maps a state onto a real number which describes the associated degree of happiness.



Case study

- **Utility-Based Agent: Online Retail Recommendation System**
- **Case Study:**
- A utility-based agent is employed in an online retail recommendation system, such as Amazon's.

- **Scenario:**
- The agent evaluates the utility or preference of different products for a user based on their browsing and purchase history, recommending items with the highest predicted utility.

Classification of Agents

- **Learning Agents**
- **Rational Agents**
- **Belief-Desire-Intention (BDI) Agents**
- **Perception-Based Agents**
- **Collaborative Agents**
- **Reactive vs. Deliberative Agents**
- **Mobile Agents**
- **Intelligent Software Agents**

Classification of Agents

- **Learning Agents:**
- Learning agents have the ability to improve their performance over time by learning from experience. They adapt their behavior based on feedback from the environment.

Case study: Email Spam Filter

- A learning agent is used as an email spam filter.

- **Scenario:**

- The agent learns from the user's actions (marking emails as spam or not) and continuously adapts its filtering criteria to improve accuracy over time.

Classification of Agents

- **Rational Agents:**
- Rational agents make decisions that maximize expected utility, given their current knowledge and beliefs about the world. They are designed to act logically and effectively.

Case Study: Stock Trading Algorithm

A rational agent operates as a stock trading algorithm.

- **Scenario:**

The agent analyzes market data, news, and historical trends to make buy or sell decisions, aiming to maximize returns while considering risk.

Example of Rational Agents

Agent Type	Performance Measure	Environment	Actuators	Sensors
Robot soccer player	Winning game, goals for/against	Field, ball, own team, other team, own body	Devices (e.g., legs) for locomotion and kicking	Camera, touch sensors, accelerometers, orientation sensors, wheel/joint encoders
Internet book-shopping agent	Obtain re-quested/interesting books, minimize expenditure	Internet	Follow link, enter/submit data in fields, display to user	Web pages, user requests
Autonomous Mars rover	Terrain explored and reported, samples gathered and analyzed	Launch vehicle, lander, Mars	Wheels/legs, sample collection device, analysis devices, radio transmitter	Camera, touch sensors, accelerometers, orientation sensors, , wheel/joint encoders, radio receiver

Classification of Agents

- **Belief-Desire-Intention (BDI) Agents:**
- BDI agents model human-like reasoning by maintaining beliefs about the world, desires (goals) to be achieved, and intentions (plans) to achieve those goals.

Classification of Agents: Case study

- **Belief-Desire-Intention (BDI) Agent: Social Robot in Elderly Care**
- **Case Study:**
- A BDI agent is implemented in a social robot designed to assist elderly individuals.

- **Scenario:**

- The agent maintains beliefs about the environment, desires to provide companionship and assistance, and forms intentions to perform specific tasks, such as reminding the user to take medication.

Classification of Agents

Perception-Based Agents:

- These agents focus on perception and interpretation of their environment. They use sensory information to make decisions and take actions.

Classification of Agents: Case study

- **Perception-Based Agent: Image Recognition System**
- **Case Study:**
- A perception-based agent is employed in an image recognition system.

- **Scenario:**

- The agent analyzes images using computer vision techniques, perceiving and identifying objects or patterns within the images.

Classification of Agents

Collaborative Agents:

- Collaborative agents work together with other agents or entities to achieve common goals. This may involve communication, coordination, and cooperation.

Classification of Agents: Case study

- **Collaborative Agent: Swarm Robotics**
- **Case Study:**
- Collaborative agents in the form of swarm robotics work together to accomplish tasks, such as exploring an unknown environment.

- **Scenario:**

- Each robot in the swarm communicates with others to share information about the terrain, optimizing exploration efficiency.

Classification of Agents

Reactive vs. Deliberative Agents:

- Reactive agents respond quickly to immediate stimuli without extensive planning. Deliberative agents, on the other hand, engage in more thoughtful planning and decision-making.

Classification of Agents: Case study

- **Reactive vs. Deliberative Agent: Robotic Vacuum Cleaner**
- **Case Study:**
- A robotic vacuum cleaner can be modeled as a reactive agent.

- **Scenario:**

- The vacuum cleaner reacts to immediate dirt detection by changing its cleaning pattern or direction.

Classification of Agents

Mobile Agents:

- These agents have the ability to move between different computational environments, executing tasks while in transit.

Classification of Agents: Case study

- **Mobile Agent: Search and Rescue Drone Fleet**
- **Case Study:**
- Mobile agents in the form of search and rescue drones are deployed to locate and assist disaster victims.

- **Scenario:**

- Drones move between different areas affected by a disaster, searching for survivors and delivering essential supplies.

Classification of Agents

Intelligent Software Agents:

- General term encompassing various types of software entities that exhibit intelligent behavior, including virtual assistants, recommendation systems, and autonomous agents in robotics.

Classification of Agents: Case study

- **Intelligent Software Agent: Siri (Virtual Assistant)**
- **Case Study:**
- Siri, Apple's virtual assistant, serves as an intelligent software agent.

- **Scenario:**

- Siri responds to user voice commands, providing information, setting reminders, and initiating various tasks on the user's device.

Types Of Environment In AI

- Fully Observable / Partially Observable
- Deterministic / Non-deterministic
- Episodic / Non-episodic(Sequential)
- Static / Dynamic
- Discrete / Continuous
- Single agent / Multiple agents

Fully Observable/Partially Observable

An agent's sensors give it access to complete state of the environment at each point in time, then we say that the task environment is fully observable; otherwise it is only partially observable.

Chess is fully observed: A player gets to see the whole board.

Poker is partially observable: A player gets to see only his own cards, not the cards of everyone in the game.

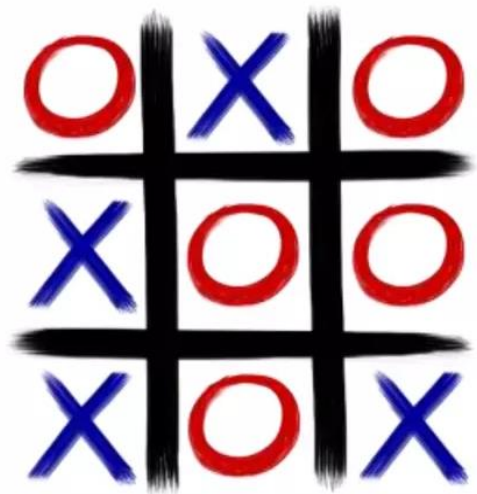
Deterministic/Non-Deterministic

If the next state of the environment is completely determined by the current state and the actions of the agent, then the environment is deterministic; otherwise it is non-deterministic.

Examples:

Deterministic environment: Tic Tac Toe game

Self-driving vehicles are a classic example of Non-Deterministic AI processes.



Episodic / Non-episodic(Sequential)

The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action) and the choice of action in each episode depends only on the episode itself. Episodic environments are much simpler because the agent does not need to think ahead.

Sequential if current decisions affect future decisions, or rely on previous ones.

Examples:

Episodic environment: mail sorting system

Non-episodic environment: chess game

Static / Dynamic

If the environment does not change while an agent is acting, then it is static; otherwise it is dynamic.

OR

An environment is static if only the actions of an agent modify it. It is dynamic on the other hand if other processes are operating on it.

- Dynamic if the environment may change over time.
- Static if nothing (other than the agent) in the environment changes.

Examples:

(Static): If we add 2+2=4 this will remain same they will never be change…

(Dynamic**):** Playing football game, other players make it dynamic.Every action there will be new reaction..

Discrete / Continuous

If there are a limited number of distinct, clearly defined, states of the environment, the environment is discrete.

E.g. A game of chess or checkers where there are a set number of moves

Continuous = Signals constantly coming into sensors, actions continually changing.

E.g. Taxi driving. There could be a route from to anywhere to anywhere else. OR Simple Driving a car

Single agent / Multiple agents

- An agent operating by itself in an environment is single agent!
- Multi agent is when other agents are present!

Examples:

(Static): Automated Taxi Car, Crossword Puzzle

(Multi agent): Other players in a football team (or opposing team)

PEAS

- Use PEAS to describe task environment
  - Performance measure
  - Environment
  - Actuators
  - Sensors
- Example: Taxi driver
  - Performance measure: safe, fast, comfortable (maximize profits)
  - Environment: roads, other traffic, pedestrians, customers
  - Actuators: steering, accelerator, brake, signal, horn
  - Sensors: cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors

PEAS

- Agent: Medical diagnosis system
- Performance measure: Healthy patient, minimize costs, lawsuits
- Environment: Patient, hospital, staff
- Actuators: Screen display (questions, tests, diagnoses, treatments, referrals)
-
- Sensors: Keyboard (entry of symptoms, findings, patient's answers)

PEAS

- Agent: Interactive English tutor
- Performance measure: Maximize student's score on test
- Environment: Set of students
- Actuators: Screen display (exercises, suggestions, corrections)
- Sensors: Keyboard

PEAS Description

- **Performance** Measure defines the goal and the metrics used to evaluate the success of the agent.
- **Environment** outlines the context and entities with which the agent interacts.
- **Actuators** describe how the agent takes actions to influence the environment.
- **Sensors** detail how the agent perceives and gathers information about its surroundings.

PEAS Description

1.Performance Measure:

1. **Goal:** Specifies what the agent is intended to achieve or optimize.
2. **Measure:** Defines how the success or failure of the agent is quantitatively evaluated. It could be a numerical value or a set of criteria.

2.Environment:

1. **Components:** Describes the entities or objects the agent interacts with in its surroundings.
2. **Dynamics:** Explains how the environment changes over time and the factors that influence these changes.

3.Actuators:

1. **Actions:** Identifies the mechanisms or processes through which the agent affects its environment.
2. **Strategy:** Describes the decision-making or control mechanisms the agent employs to determine its actions.

4.Sensors:

1. **Inputs:** Specifies the sources of information or data the agent uses to perceive and understand its environment.
2. **Feedback:** Describes how the agent receives information about the consequences of its actions and the changes in the environment.

PEAS Description: Case Study

- **1. PEAS Description for a Chess-Playing Agent:**
- **Performance Measure:**
  - **Goal:** Win the game.
  - **Measure:** Number of games won, time taken for decision-making, quality of moves.
- **Environment:**
  - **Components:** Chessboard, chess pieces, opponent player.
  - **Dynamics:** Turn-based moves, legal and illegal moves, game rules.
- **Actuators:**
  - **Actions:** Moving chess pieces on the board, capturing opponent pieces.
  - **Strategy:** Decision-making based on the current board state.
- **Sensors:**
  - **Inputs:** Current state of the chessboard, positions of all pieces.
  - **Feedback:** Opponent moves, overall game state.

PEAS Description: Case Study

- **PEAS Description for a Recommender System:**
- **Performance Measure:**
  - **Goal:** Maximize user satisfaction and engagement.
  - **Measure:** Accuracy of recommendations, user click-through rates, user feedback.
- **Environment:**
  - **Components:** User profiles, item database (movies, products), historical user interactions.
  - **Dynamics:** Changing user preferences, new items, user feedback.
- **Actuators:**
  - **Actions:** Recommending items based on user preferences.
  - **Strategy:** Machine learning algorithms, collaborative filtering.
- **Sensors:**
  - **Inputs:** User preferences, browsing history, item features.
  - **Feedback:** User interactions, ratings, feedback on recommendations.

PEAS Description: Case Study

- **PEAS Description for an Autonomous Drone:**
- **Performance Measure:**
  - **Goal:** Efficient exploration and data collection in a designated area.
  - **Measure:** Coverage of the area, time taken to complete the mission, quality of collected data.
- **Environment:**
  - **Components:** Physical space, obstacles, waypoints, other drones (if collaborative).
  - **Dynamics:** Changing weather conditions, obstacles, real-time sensor data.
- **Actuators:**
  - **Actions:** Movement in three-dimensional space, capturing images or data.
  - **Strategy:** Path planning, obstacle avoidance, collaborative coordination.
- **Sensors:**
  - **Inputs:** GPS data, altitude, obstacle detection, camera images.
  - **Feedback:** Environmental changes, proximity to obstacles, status of data collection.

Single and Multi-Agent System

- In a single-agent system, there is only one autonomous entity or intelligent agent that operates in its environment to achieve specific goals.

- This agent is responsible for perceiving the environment, making decisions, and taking actions without interacting with other agents.

- Single-agent systems are common in many AI applications, such as game-playing agents, expert systems, and autonomous robots.

- In a multi-agent system, multiple intelligent agents exist and operate within the same environment.
- These agents may have their own goals, capabilities, and decision-making processes. The agents can interact with each other, sharing information and influencing each other's behavior.
- Multi-agent systems are prevalent in applications where collaboration, coordination, or competition between agents is essential.

Single and Multi-Agent System

- ***Example: Chess-Playing Agent***
- **Agent:** The chess-playing program that makes decisions based on the current state of the chessboard and its goal to win the game.
- **Environment:** The chessboard, opponent, and game rules.
- **Performance Measure:** Winning the game or making optimal moves.
- ***Example: Multi-Agent Robotic Swarm***
- **Agents:** A group of small robotic drones working together to explore an unknown environment.
- **Environment:** The physical space with obstacles, potential targets, and other drones.
- **Performance Measure:** Efficient exploration, coverage of the entire area.

Single and Multi-Agent System

- **Key Differences:**

1.Interaction:

1. **Single-Agent:** Operates independently without interaction with other agents.
2. **Multi-Agent:** Involves interactions, collaborations, or competitions between multiple agents.

2.Goals:

1. **Single-Agent:** Has its own set of goals and objectives.
2. **Multi-Agent:** Each agent may have individual goals, and there may be collective goals for the group.

3.Decision-Making:

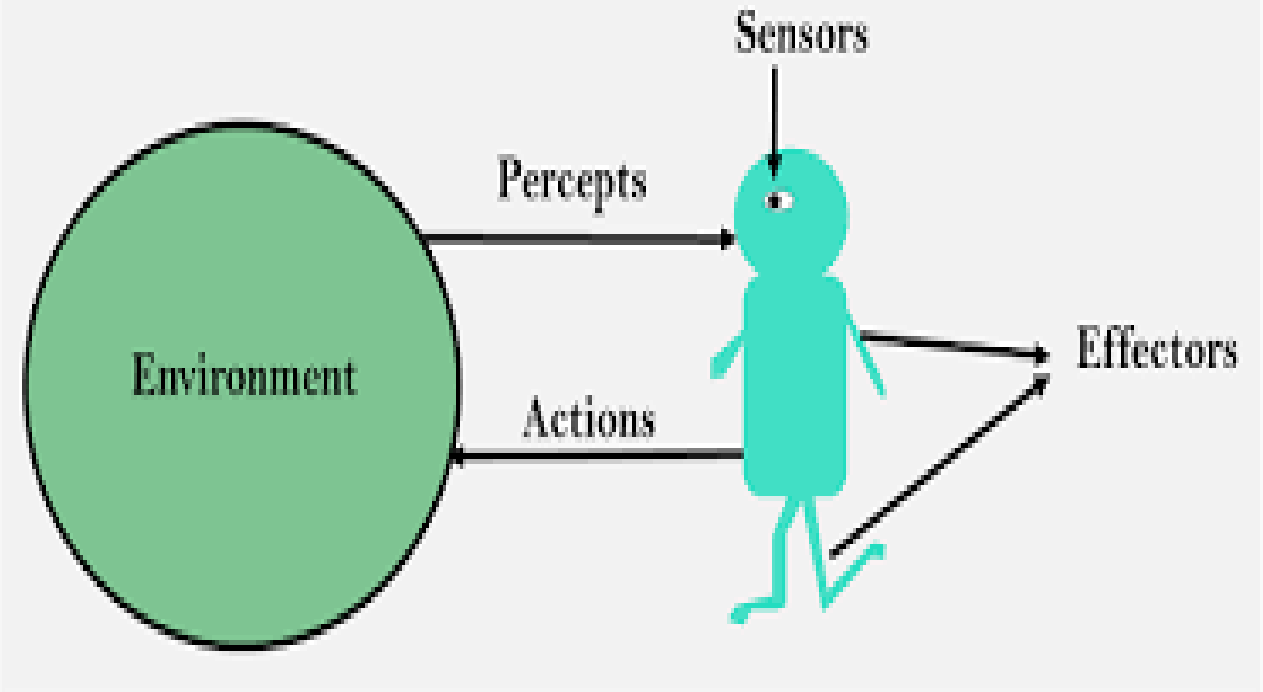
1. **Single-Agent:** Makes decisions based on its own perception and goals.
2. **Multi-Agent:** Decision-making may involve considering the actions and influence of other agents.

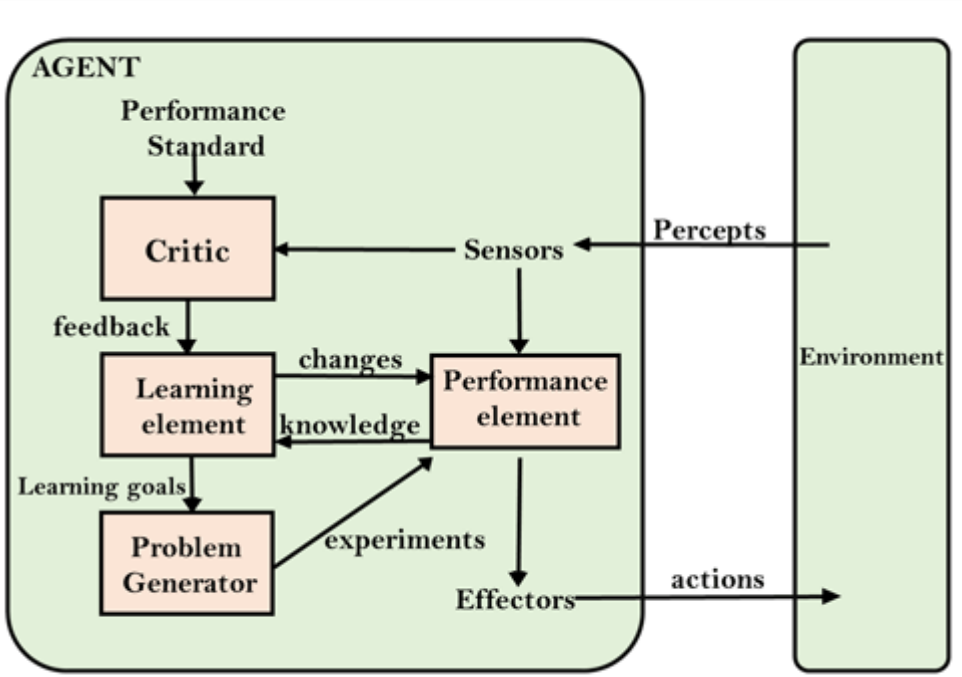
4.Complexity:

1. **Single-Agent:** Generally simpler, as there's only one agent to consider.
2. **Multi-Agent:** Can be more complex due to interactions and coordination between multiple agents.

5.Applications:

1. **Single-Agent:** Found in applications where individual entities operate independently.
2. **Multi-Agent:** Used in scenarios requiring collaboration, distributed problem-solving, or competition.


Sensors
Percepts
Environment
Actions
Effectors


AGENT
Performance Standard
Critic
Sensors
Percepts
feedback
changes
Learning element
Performance element
knowledge
Learning goals
Problem Generator
experiments
Effectors
actions
Environment

Agents in AI