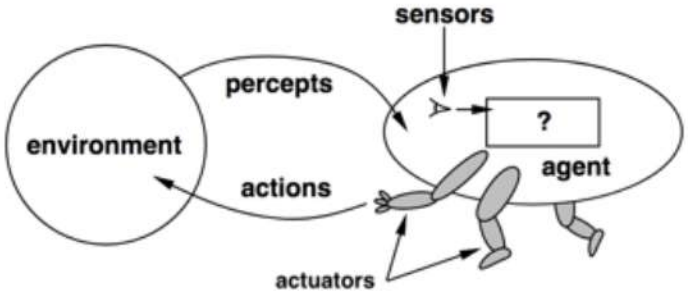
Unit-2
Agents

Introduction

An ***intelligent agent*** is an autonomous entity which acts upon an environment using sensors and actuators for achieving goals. It perceives its environment via sensors and acts rationally upon that environment with its effectors (actuators).



Properties of the agent:

- Autonomous
- Interacts with other agents plus the environment
- Reactive to the environment
- Pro-active (goal-directed)

- ***Percept:*** 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.
- ***Action:*** An operation involving an actuator is called an action. Action can be grouped into action sequences.
- ***Percept Sequence:*** It is the history of all that an agent has perceived till date.

Examples of agents

- ***Human agent:*** eyes, ears, nose, and other organs for sensors, and hands, legs, mouth, and other body parts for effectors.
- ***Robotic agent:*** cameras and infrared range finders for sensors, and various motors for effectors.
- ***Software agent:*** It receives keystrokes, file contents, and network packets as sensory inputs and acts on the environment by displaying on the screen, writing files, and sending network packets.

Structure of an Intelligent Agent

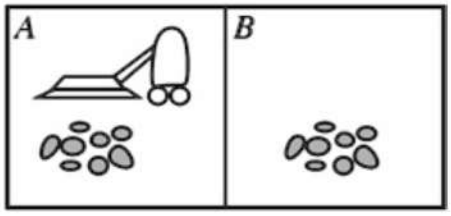
The structure of an intelligent agent is a combination of architecture and agent program. It can be viewed as:

$$\boldsymbol{Agent = Architecture + Agent\ Program}$$

Following are the main three terms involved in the structure of an AI agent:

- ***Architecture:*** Architecture is machinery that an AI agent executes on. It is a device with sensors and actuators.
- ***Agent Function:*** Agent function is used to map a percept sequence to an action. $\boldsymbol{f: P^* \rightarrow A}$
- ***Agent program:*** Agent program is an implementation of agent function. An *agent program* executes on the physical *architecture* to produce function $\boldsymbol{f}$.

The vacuum-cleaner world: Example of Agent



- Environment: square A and B
- Percepts: [location and content] e.g. [A, Dirty]
- Actions: left, right, suck, and no-op

Percept Sequence	***Action***
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
…….	…….

Agent's Performance

An agent's performance refers to how well an artificial intelligence (AI) agent accomplishes its assigned tasks within a specific environment.

To evaluate an agent's performance, the following key metrics are often considered:

- ***Accuracy:*** Measures how frequently the agent produces correct outputs.
- ***Speed/Efficiency:*** Evaluates how quickly the agent can complete tasks.
- ***Success Rate:*** The percentage of successfully completed tasks out of total attempts.
- ***Customer Satisfaction:*** In scenarios like customer service, this metric relies on feedback and interaction quality from users.
- ***Cost per Action:*** The cost of an agent performing a specific task.

Factors affecting agent performance

Several elements influence how well an agent performs:

- ***Agent Design:*** The architecture and algorithms used in the agent, including its decision-making and learning mechanisms, significantly impact its effectiveness.
- ***Training Data:*** The quality, quantity, and relevance of data provided during training play a critical role in enabling the agent to generalize and adapt.
- ***Environment Complexity:*** The complexity of the operating environment, such as unpredictability, dynamic changes, or partially observable conditions, challenges the agent's adaptability and performance.

Example

Performance measure of a ***vacuum-cleaner agent*** could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated etc.

Rationality and Omniscience

Rationality

A ***rational agent*** always performs right action, where the *right action* means the action that causes the agent to be most successful in the given percept sequence.

The ***rationality*** of an agent at a given time depends on four factors:

- *Performance Measure:* Defines the criterion of success.
- *Prior Environment Knowledge:* What the agent knows about the environment beforehand.
- *Actions:* The actions the agent can perform.
- *Percept Sequence to date:* The sequence of all percepts received up to the current point in time.

A ***rational agent*** chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date and prior environment knowledge.

Omniscience

Omniscience refers to having complete knowledge about the environment, including all outcomes of all possible actions.

An ***omniscient (perfect)*** agent knows the actual outcome of its actions and can act accordingly; but perfection is impossible in reality.

Rationality is NOT the same as ***perfection***. ***Rationality*** maximizes *expected performance*, while ***perfection*** maximizes *actual performance*.

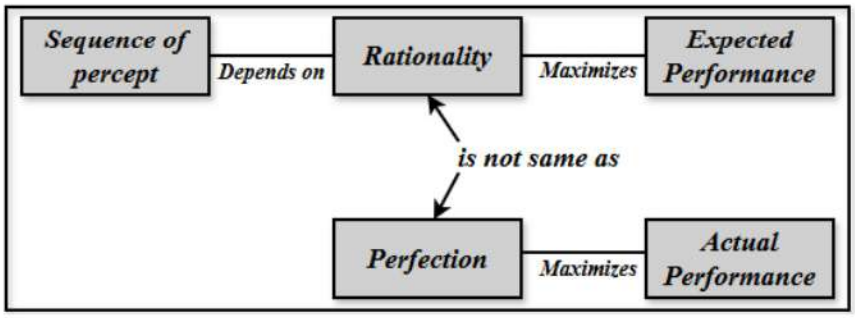


Fig: The relationship between rationality and omniscience

Rationality vs. Omniscience

Rationality	*Omniscience*
Acts to maximize expected performance based on available knowledge and resources.	Always knows and chooses the optimal action to maximize actual performance.
Operates with limited and incomplete knowledge of the environment.	Has perfect and complete knowledge of the environment and all outcomes.
Achievable in real-world scenarios.	Unrealistic and unattainable in practical applications.
Chooses the best action given constraints and uncertainty.	Always chooses the optimal action with certainty.
E.g. A chess-playing AI that evaluates the game state and selects the best possible move.	E.g. A perfect chess player that foresees all possible moves and outcomes and always makes the optimal move to win

Task Environment: PEAS

To design a *rational agent*, we must specify the ***task environment***. The performance measure, the environment, and the agent's actuators and sensors are grouped as the ***task environment***, and called as ***PEAS*** (Performance measure, Environment, Actuators, Sensors).

- ***Performance measure:*** It defines the success of an agent. It evaluates the criteria that determines whether the system performs well.
- ***Environment:*** All the surrounding things and conditions of an agent fall in this section. It basically consists of all the things under which the agents work.
- ***Actuators:*** The devices, hardware or software through which the agent performs any actions or processes any information to produce a result are the actuators of the agent.
- ***Sensors:*** The devices through which the agent observes and perceives its environment are the sensors of the agent.

Examples

Agent: Vacuum cleaner - ***Performance Measure:*** Cleanliness, security, battery - ***Environments:*** Room, table, carpet, floors - ***Actuators:*** Wheels, brushes, vacuum extractor. - ***Sensors:*** camera, dirt detection sensor, cliff sensor, bump sensors, infrared wall sensors	***Agent: Automated taxi driver*** - ***Performance Measure:*** Safe, fast, legal, comfortable trip, maximize profits. - ***Environment:*** Roads, other traffic, pedestrians, customers. - ***Actuators:*** Steering wheel, accelerator, brake, signal, horn. - ***Sensors:*** Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard.
Agent: Medical diagnosis system - ***Performance Measure:*** Healthy patient, Minimized cost - ***Environments:*** Patient, Hospital, Staff - ***Actuators:*** Display question, Tests, Diagnoses, Treatments - ***Sensors:*** Keyboard entry of symptoms, Findings, Patient's answers	***Agent: Part-picking robot*** - ***Performance Measure:*** Percentage of parts in correct bins - ***Environments:*** Conveyor belt with parts, Bins - ***Actuators:*** Jointed arm and hand - ***Sensors:*** Camera, Joint angle sensors
Agent: Internet shopping assistant - ***Performance Measure:*** price, quality, appropriateness, efficiency - ***Environments:*** current and future WWW sites, vendors, shippers - ***Actuators:*** display to user, follow URL, fill in form - ***Sensors:*** Web pages (text, graphics, scripts)	***Agent: Playing soccer*** - ***Performance Measure:*** Scoring goals, Defending, speed - ***Environments:*** Soccer, Playground, Teammates, Opponents, Referee, Audience - ***Actuators:*** Legs, Head, Upper body - ***Sensors:*** Camera, Ball sensor, Location sensor, Other players locator

Agent: Interactive English tutor	***Agent: Satellite image analysis system***
- ***Performance Measure:*** Maximize student's score on test. - ***Environments:*** Set of students, Testing agency - ***Actuators:*** Display exercises, Suggestions, Corrections - ***Sensors:*** Keyboard entry	- ***Performance Measure:*** Correct image categorization - ***Environments:*** Images from orbiting satellite - ***Actuators:*** Display categorization of scene - ***Sensors:*** Color pixel arrays
Agent: Human agent	***Agent: Robotic agent***
- ***Performance Measure:*** Achieve goals, follow societal norms, maximize efficiency, maintain well-being, happiness. - ***Environment:*** Physical world, other humans, societal structures, natural elements. - ***Actuators:*** Limbs, vocal cords, facial expressions. - ***Sensors:*** Eyes, ears, skin, tongue, nose, internal sensory systems.	- ***Performance Measure:*** Task efficiency, accuracy, reliability, resource optimization. - ***Environment:*** Task-specific surroundings (e.g., factories, homes, offices), objects, humans. - ***Actuators:*** Robotic arms, wheels, grippers, speakers, tools. - ***Sensors:*** Cameras, microphones, touch sensors, infrared, ultrasonic, gyroscope, accelerometer.
Agent: Solver for the 8-queen problem	***Agent: Refinery Controller***
- ***Performance Measure:*** Correct placement of 8 queens such that no two queens threaten each other. - ***Environment:*** 8x8 chessboard. - ***Actuators:*** Ability to place or move queens on the board. - ***Sensors:*** Board state (positions of queens).	- ***Performance Measure:*** maximize purity, yield, safety - ***Environment:*** refinery, operators - ***Actuators:*** valves, pumps, heaters, displays - ***Sensors:*** temperature, pressure, chemical sensors

Types or Properties of Agent Environment

The ***environment type*** largely determines the agent design.

1. ***Fully observable vs. Partially observable:*** If an agent's sensors give it access to the complete state of the environment at each point in time, then task environment is called as *fully observable*; otherwise it is only *partially observable*.
   - Fully observable environments are convenient because the agent need not maintain any internal state to keep track of the world.

   Examples:

   - Fully observable: Chess (A player gets to see the whole board).
   - Partially observable: Poker (A player gets to see only his own cards, not the cards of everyone in the game)

2. ***Deterministic vs. Stochastic:*** If the next state of the environment is completely determined by the current state and the action executed by the agent, then the environment is *deterministic*; otherwise it is *stochastic*. A stochastic environment is random in nature and cannot be determined completely by an agent.

   Examples:

   - Deterministic environment: Tic Tac Toe game (The result of placing a mark (X or O) in a specific cell is predictable)
   - Stochastic environment: Self-driving vehicles (because one can never predict the behavior of traffic exactly)

3. ***Static vs. Dynamic:*** If the environment does not change while an agent is acting, then it is *static*; otherwise it is *dynamic*.

   Examples:

   - Taxi driving is dynamic.
   - Crossword puzzles are static.

4. ***Discrete vs. Continuous:*** If there are a limited number of distinct, clearly defined states, percepts and actions, the environment is *discrete*; otherwise it is *continuous*.

   Examples:

   - Chess has finite number of discrete states, and has discrete set of percepts and actions.
   - Taxi driving has continuous states, and actions.

5. ***Single agent vs. Multi agent:*** If only one agent is involved in an environment, and operating by itself then such an environment is called *single agent environment*. However, if multiple agents are operating in an environment, then such an environment is called a *multi-agent environment*.

   Examples:

   - An agent solving a ***crossword puzzle*** by itself is clearly in a single-agent environment.
   - An agent playing ***chess*** is in a two- agent environment

6. ***Episodic vs. Sequential:*** In an *episodic task environment*, the agent's experience is divided into atomic episodes. In each episode the agent receives a percept and then performs a single action. The next episode does not depend on the actions taken in previous episodes.

   In *Sequential environment*, current actions may affect all future decisions. In sequential environment, an agent requires memory of past action to determine the next best action.

   Examples:

   - Part-Picking Robot is episodic: each picking task is independent, and the robot's current action does not affect future tasks.
   - Chess and taxi driving are sequential: in both cases, short-term actions can have long term consequences.

Examples of task environments

Task Environment	*observable*	*deterministic / stochastic*	*episodic / sequential*	*Static / Dynamic*	*Discrete / continuous*	*agents*
Taxi driving	partially	stochastic	sequential	dynamic	continuous	multi
Crossword puzzle	fully	deterministic	sequential	static	discrete	single
Chess with a clock	fully	deterministic	sequential	semi-dynamic	discrete	multi
Chess without clock	fully	deterministic	sequential	static	discrete	multi
Poker	partially	stochastic	sequential	static	discrete	multi
Medical diagnosis	partially	stochastic	sequential	dynamic	continuous	single
Image analysis	fully	deterministic	episodic	Semi-dynamic	continuous	single
Part-picking robot	partially	stochastic	episodic	dynamic	continuous	single
Refinery controller	partially	stochastic	sequential	dynamic	continuous	single
Interactive English tutor	partially	stochastic	sequential	dynamic	discrete	multi

Agent Architecture

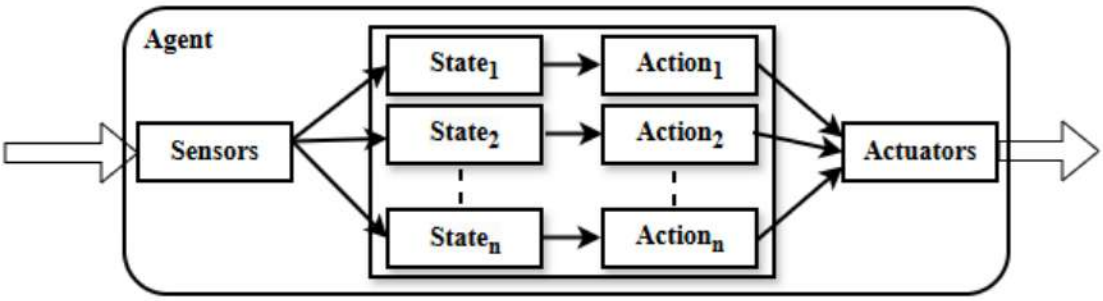
Agent architecture refers to the framework or design that defines how intelligent agents perceive their environment, make decisions, and take actions to achieve specific goals. It describes the essential components and how they interact, providing a clear structure for building and understanding intelligent systems. The architecture ensures that the agent can function autonomously, responding to its environment and working towards its objectives in a systematic way.

Components of Agent Architecture

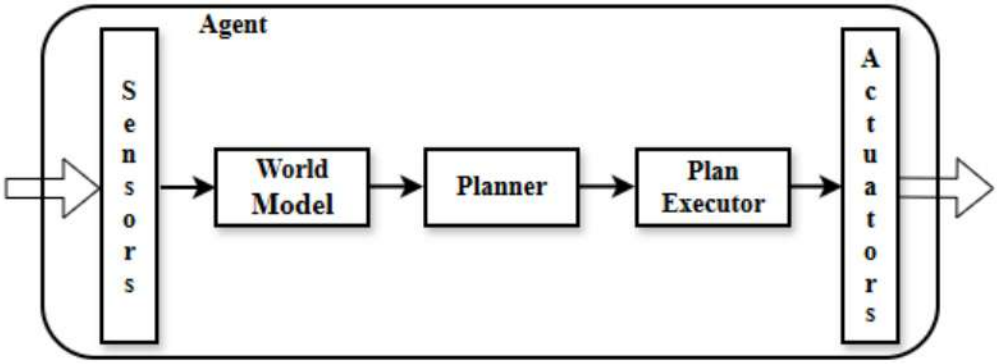
- ***Perception Module:*** With the help of this module, the agent may collect and analyze information from its environment.
- ***Memory Module:*** The memory module is essential for organizing and storing data as it serves as the ***agent's knowledge base***. It allows the agent to remember information, rules, patterns, and events from the past.
- ***Reasoning/Decision Making Module:*** This module analyzes the current situation and decides the best action to achieve the agent's goals using information from memory and perception.
- ***Action Module:*** The action module executes the agent's decisions through actuators to interact with the environment.
- ***Learning Module:*** The learning module helps agents change, improve, and learn new things by adapting based on past experiences.

Types of Agent Architecture

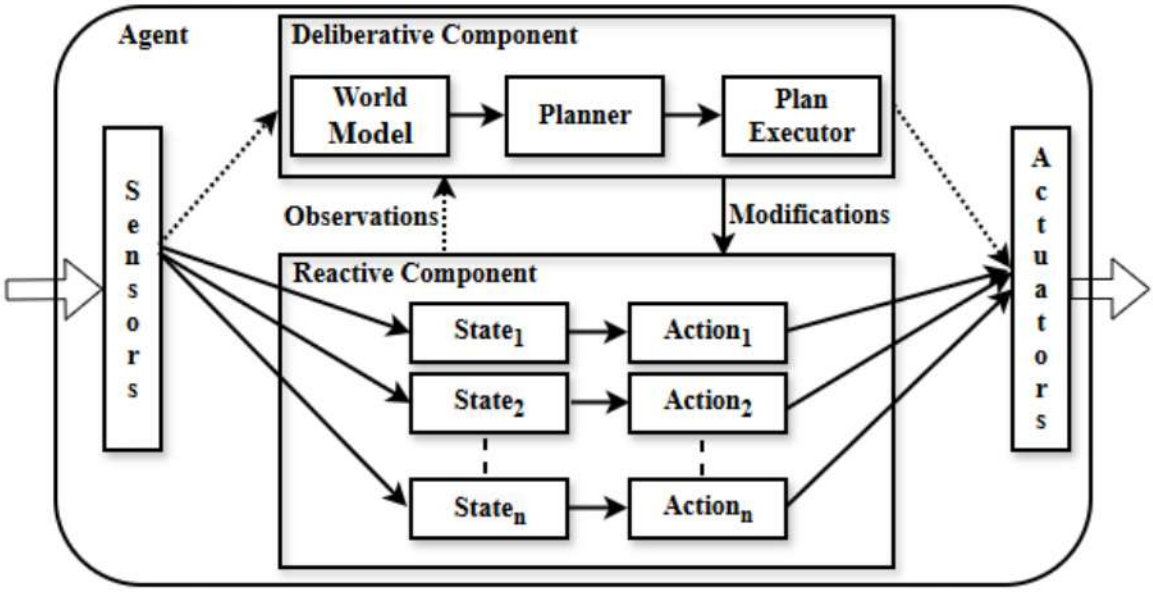
- ***Reactive Architectures:*** These are simple and do not involve any internal symbolic model of the world. They operate based on a set of predefined rules and are often used in scenarios where quick responses are essential.



- ***Deliberative Architectures:*** These involve an internal symbolic model of the world, and agents can use this model to deliberate on what action to take. They are capable of more complex behaviors because they can plan and reason about future actions.



- ***Hybrid Architectures:*** These combine elements of both reactive and deliberative architectures. They leverage the strengths of each type to create more robust agents.
  - A *deliberative* one, containing a symbolic world model, which develops plans and makes decisions.
  - A *reactive* one, which is capable of reacting quickly to events without complex reasoning.



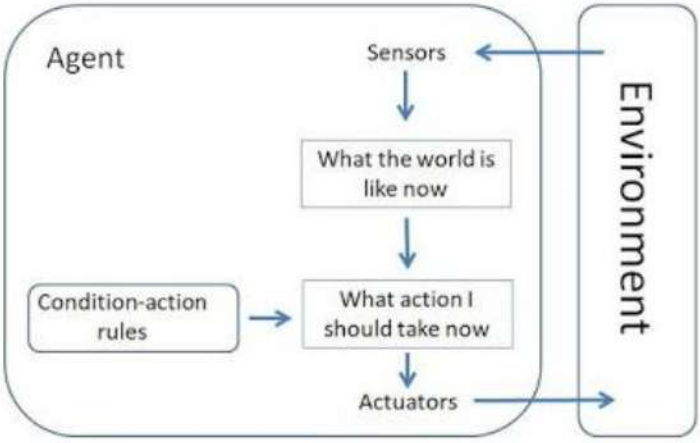
Types of Agent

There are 5 main categories of intelligent agents.

1. *Simple Reflex Agent*

Simple reflex agents take decisions on the basis of the current percepts and ignore the rest of the percept history. The agent function is based on the *condition-action rule*. A condition-action rule is a rule that maps the current state i.e. condition to action. It acts according to a rule whose condition matches the current state, as defined by the percept. For example, Room cleaner agent – it works only if there is dirt in the room.

This agent function only succeeds when the environment is fully observable.



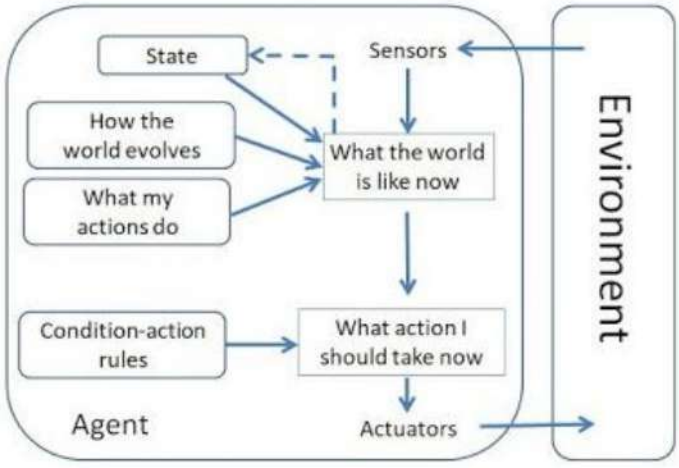
2. *Model-based Reflex Agent*

A model-based agent can handle partially observable environments by the use of a model about the world. Its current state is stored inside the agent which maintains some kind of structure describing the part of the world which cannot be seen.

A model-based reflex agent should maintain some sort of internal state that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state.

Updating the internal state information as time goes by requires two kinds of knowledge to be encoded in the agent program:

- Information about how the world evolves independently of the agent
- Information about how the agent's own actions affects the world

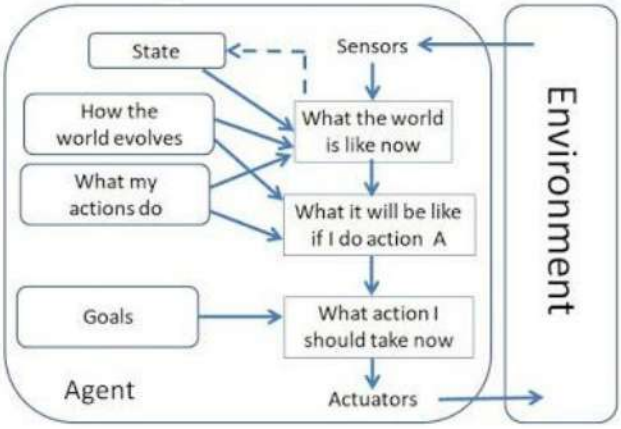


3. ***Goal Based Agent***

These agents have higher capabilities than model-based reflex agents. Goal-based agents use goal information to describe situations that are desirable. This provides the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state.

It keeps track of the world as well as a set of goals it is trying to achieve, and choose an action that will (eventually) lead to the achievement of its goal.

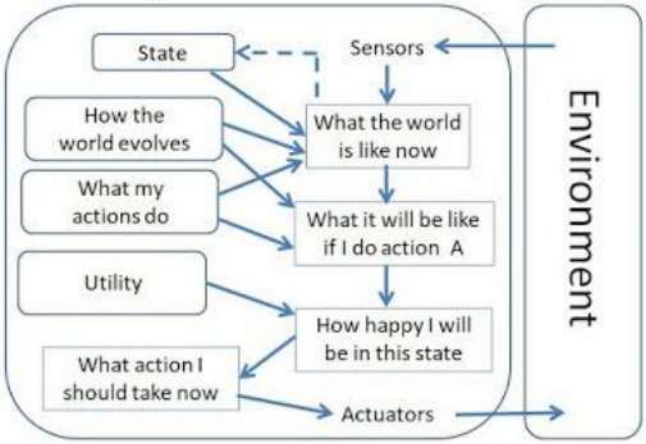
E.g. Searching robot that has an initial location and wants to reach a destination.



4. ***Utility-based Agent***

Utility-based agents help to choose the best alternatives, when there are multiple alternatives available. They choose actions based on a preference (utility) for each state. A utility agent chooses the action that maximizes the expected utility. A utility function maps a state to measure of the utility of that state which describes the agent's degree of happiness.

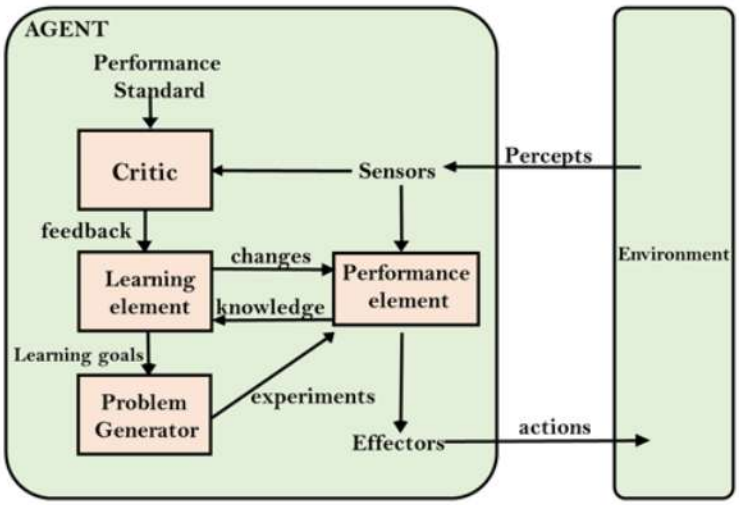
E.g. Route recommendation system which solves the 'best' route to reach a destination.



5. ***Learning Agent***

A learning agent 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.

Learning agents are able to learn, analyze performance, and look for new ways to improve the performance. For example, a recommendation system learns user preferences to provide better recommedations over time.



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

- ***Learning element:*** It is responsible for making improvements by learning from environment. It uses knowledge about the agent and feedback on its actions to improve performance.
- ***Critic (feedback element):*** 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 actions according to the percepts it takes.
- ***Problem generator:*** This component is responsible for suggesting actions that will lead to new and informative experiences.