



Agents in Artificial Intelligence

Last Updated : 05 Jun, 2023

In artificial intelligence, an agent is a computer program or system that is designed to perceive its environment, make decisions and take actions to achieve a specific goal or set of goals. The agent operates autonomously, meaning it is not directly controlled by a human operator.

Agents can be classified into different types based on their characteristics, such as whether they are reactive or proactive, whether they have a fixed or dynamic environment, and whether they are single or multi-agent systems.

- Reactive agents are those that respond to immediate stimuli from their environment and take actions based on those stimuli. Proactive agents, on the other hand, take initiative and plan ahead to achieve their goals. The environment in which an agent operates can also be fixed or dynamic. Fixed environments have a static set of rules that do not change, while dynamic environments are constantly changing and require agents to adapt to new situations.
- Multi-agent systems involve multiple agents working together to achieve a common goal. These agents may have to coordinate their actions and communicate with each other to achieve their objectives. Agents are used in a variety of applications, including robotics, gaming, and intelligent systems. They can be implemented using different programming languages and techniques, including machine learning and natural language processing.

Artificial intelligence is defined as the study of rational agents. A rational agent could be anything that makes decisions, such as a person, firm, machine, or software. It carries out an action with the best outcome after considering past and current percepts (agent's perceptual inputs at a given

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

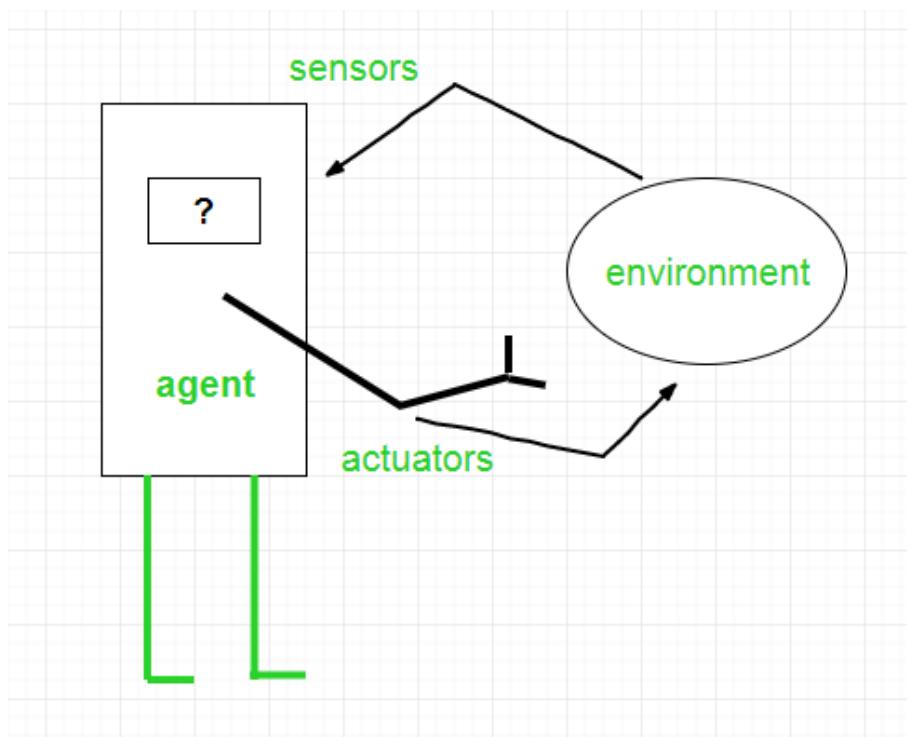
[Got It !](#)

[Privacy Policy](#)

An agent is anything that can be viewed as:

- Perceiving its environment through **sensors** and
- Acting upon that environment through **actuators**

Note: Every agent can perceive its own actions (but not always the effects).



Interaction of Agents with the Environment

Structure of an AI Agent

To understand the structure of Intelligent Agents, we should be familiar with *Architecture* and *Agent* programs. **Architecture** is the machinery that the agent executes on. It is a device with sensors and actuators, for example, a robotic car, a camera, and a PC. **An agent program** is an implementation of an agent function. **An agent function** is a map from the percept sequence(history of all that an agent has perceived to date) to an action.

We use cookies to ensure you have the best browsing experience on our website. By

using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

Got It !

[Privacy Policy](#)

There are many examples of agents in artificial intelligence. Here are a few:

- **Intelligent personal assistants:** These are agents that are designed to help users with various tasks, such as scheduling appointments, sending messages, and setting reminders. Examples of intelligent personal assistants include Siri, Alexa, and Google Assistant.
- **Autonomous robots:** These are agents that are designed to operate autonomously in the physical world. They can perform tasks such as cleaning, sorting, and delivering goods. Examples of autonomous robots include the Roomba vacuum cleaner and the Amazon delivery robot.
- **Gaming agents:** These are agents that are designed to play games, either against human opponents or other agents. Examples of gaming agents include chess-playing agents and poker-playing agents.
- **Fraud detection agents:** These are agents that are designed to detect fraudulent behavior in financial transactions. They can analyze patterns of behavior to identify suspicious activity and alert authorities. Examples of fraud detection agents include those used by banks and credit card companies.
- **Traffic management agents:** These are agents that are designed to manage traffic flow in cities. They can monitor traffic patterns, adjust traffic lights, and reroute vehicles to minimize congestion. Examples of traffic management agents include those used in smart cities around the world.
- A **software agent** has Keystrokes, file contents, received network packages that act as sensors and displays on the screen, files, and sent network packets acting as actuators.
- A Human-agent has eyes, ears, and other organs which act as sensors, and hands, legs, mouth, and other body parts act as actuators.
- A **Robotic agent** has Cameras and infrared range finders which act as sensors and various motors act as actuators.

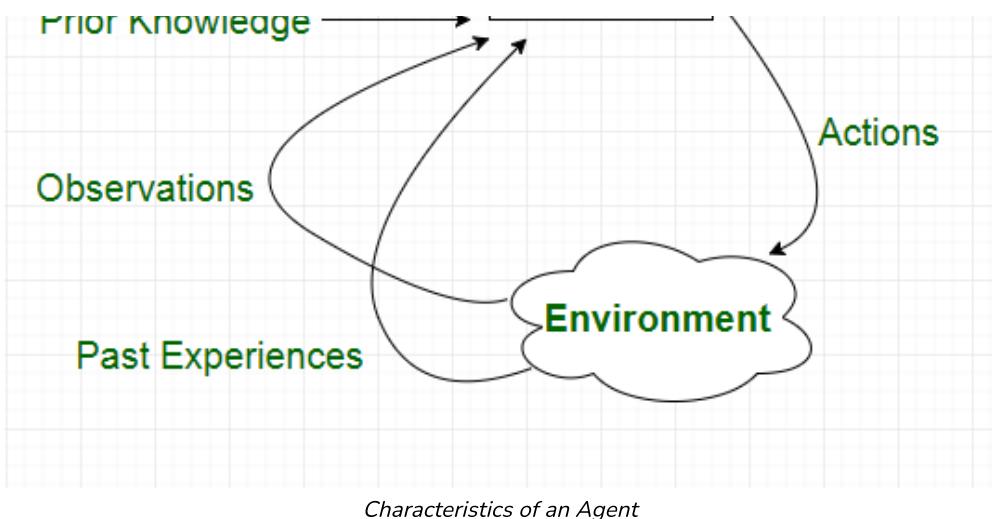


We use cookies to ensure you have the best browsing experience on our website. By

using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

Got It !

[Privacy Policy](#)



Types of Agents

Agents can be grouped into five classes based on their degree of perceived intelligence and capability :

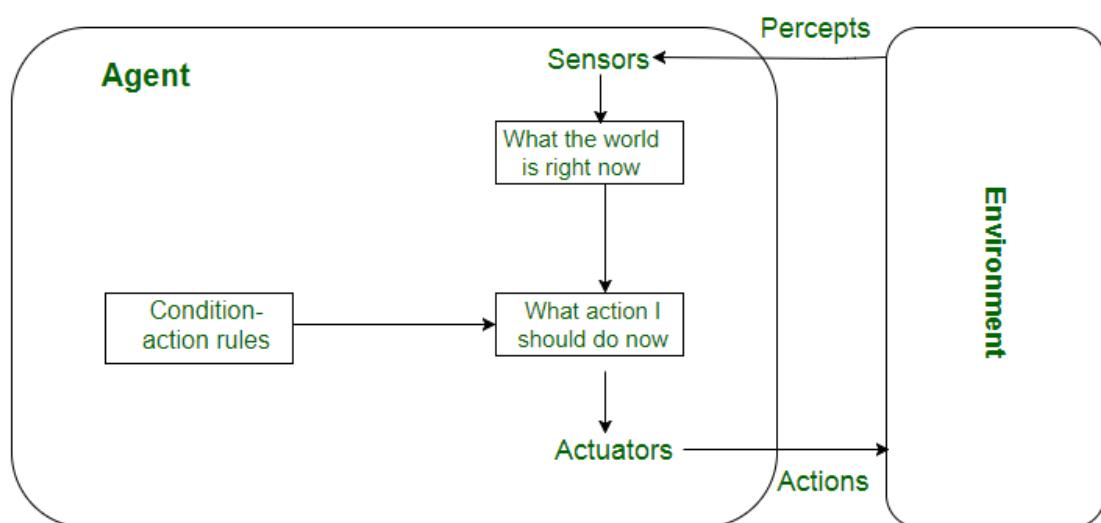
- Simple Reflex Agents
- Model-Based Reflex Agents
- Goal-Based Agents
- Utility-Based Agents
- Learning Agent
- Multi-agent systems
- Hierarchical agents

Simple Reflex Agents

Simple reflex agents ignore the rest of the percept history and act only on the basis of the **current percept**. Percept history is the history of all that an agent has perceived to date. The agent function is based on the **condition-action rule**. A condition-action rule is a rule that maps a state i.e., a condition to an action. If the condition is true, then the action is taken, else not. This agent function only succeeds when the environment is fully observable. For simple reflex agents operating in partially observable environments, infinite loops are often unavoidable. It may be possible to escape from infinite loops if the agent can randomize its actions.

Problems with Simple reflex agents are :

- If there occurs any change in the environment, then the collection of rules needs to be updated.



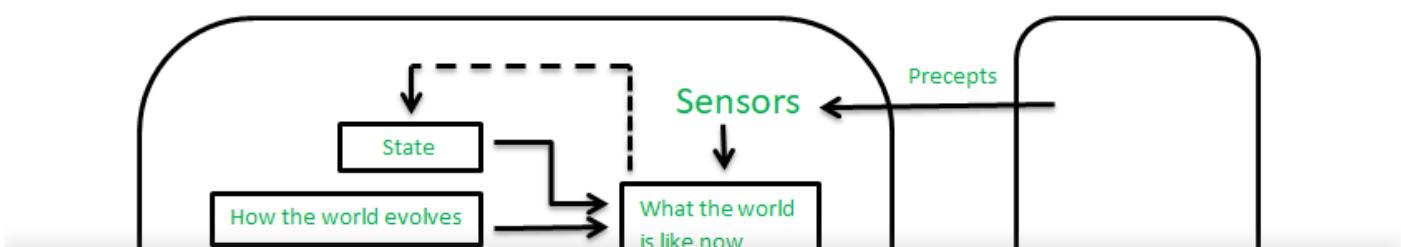
Simple Reflex Agents

Model-Based Reflex Agents

It works by finding a rule whose condition matches the current situation. A model-based agent can handle **partially observable environments** by the use of a model about the world. The agent has to keep track of the **internal state** which is adjusted by each percept and that depends on the percept history. The current state is stored inside the agent which maintains some kind of structure describing the part of the world which cannot be seen.

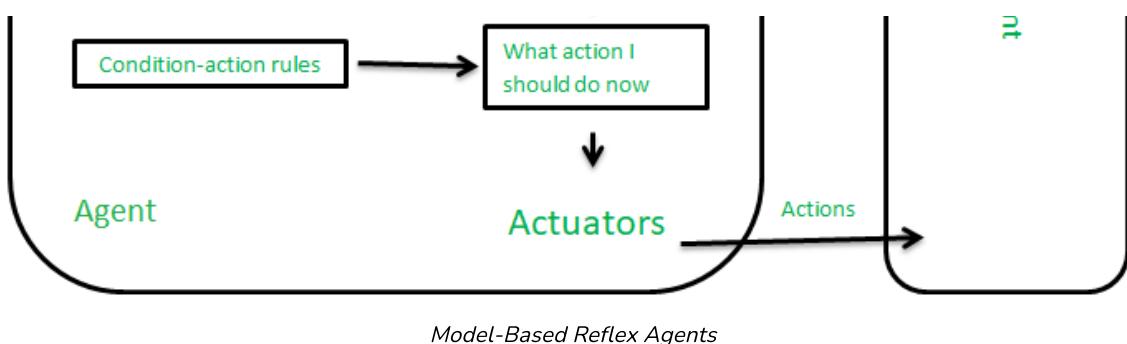
Updating the state requires information about:

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



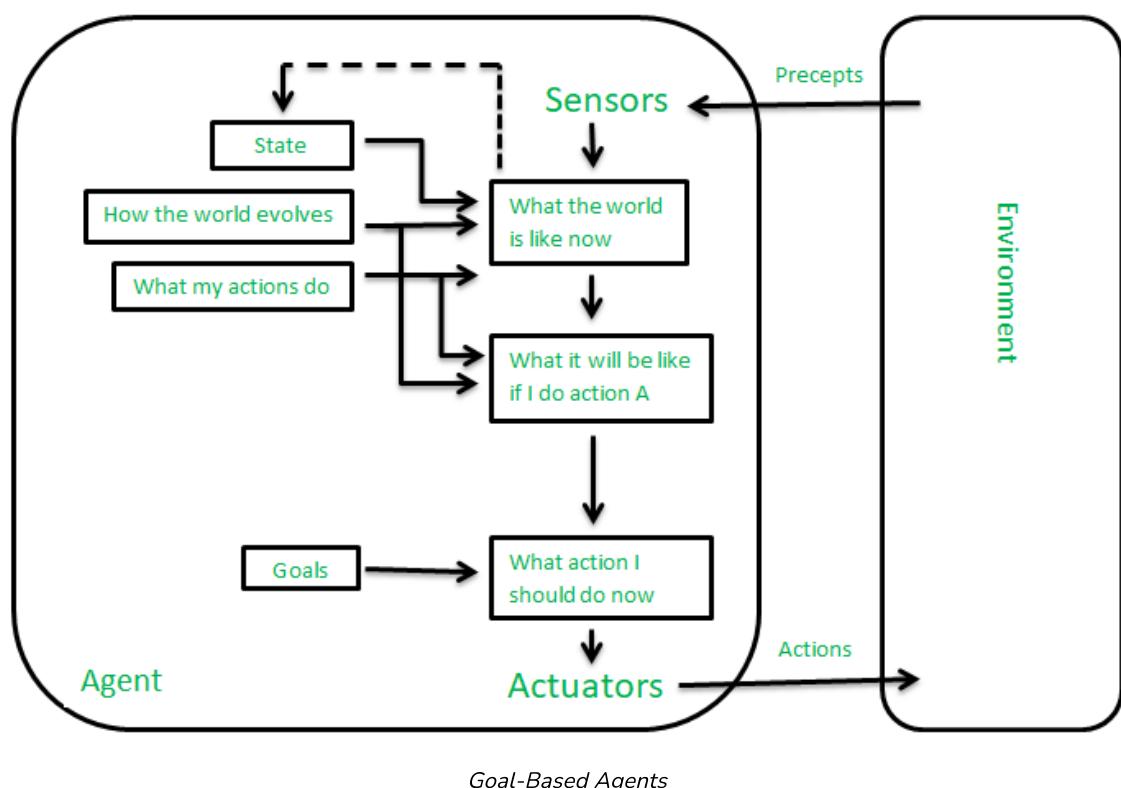
We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

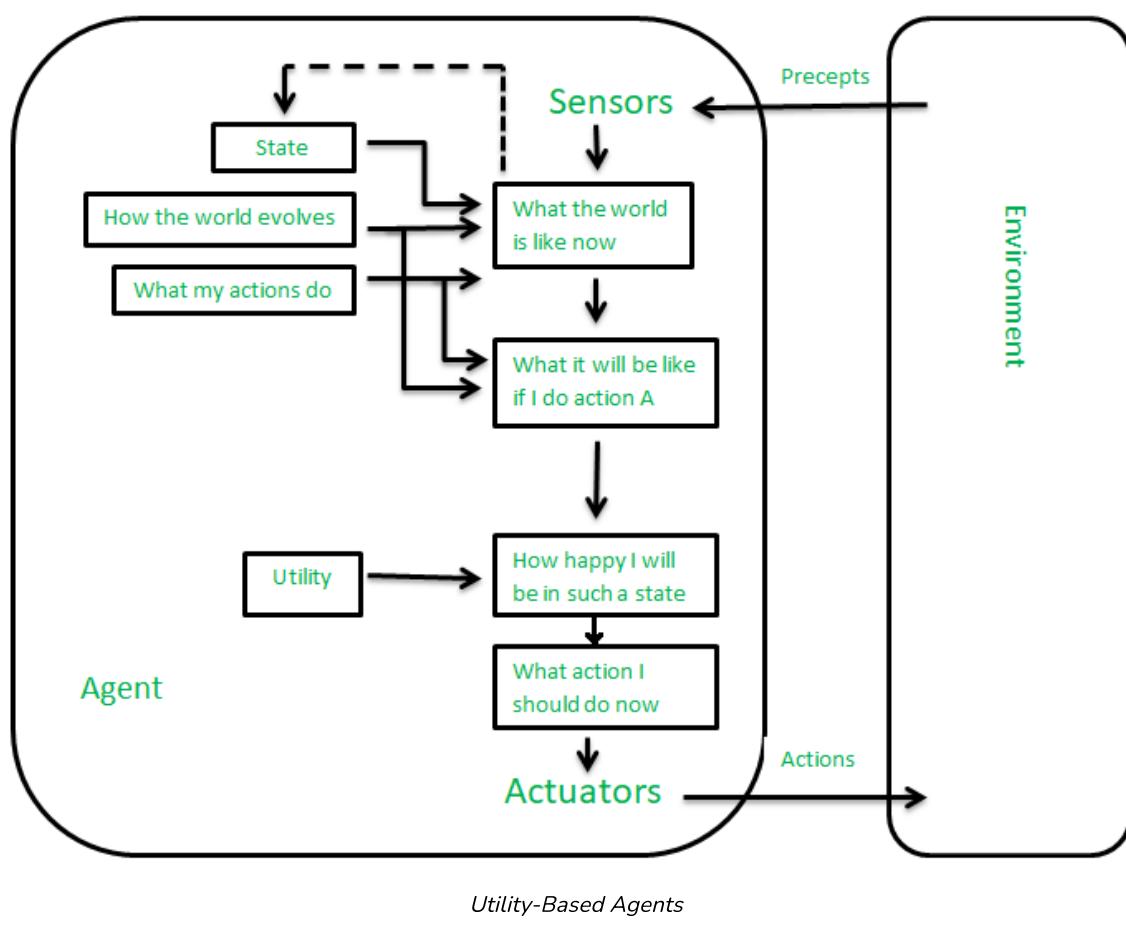


Goal-Based Agents

These kinds of agents take decisions based on how far they are currently from their **goal**(description of desirable situations). Their every action is intended to reduce their distance from the goal. This allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state. The knowledge that supports its decisions is represented explicitly and can be modified, which makes these agents more flexible. They usually require search and planning. The goal-based agent's behavior can easily be changed.



are called utility-based agents. When there are multiple possible alternatives, then to decide which one is best, utility-based agents are used. They choose actions based on a **preference (utility)** for each state. Sometimes achieving the desired goal is not enough. We may look for a quicker, safer, cheaper trip to reach a destination. Agent happiness should be taken into consideration. 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.



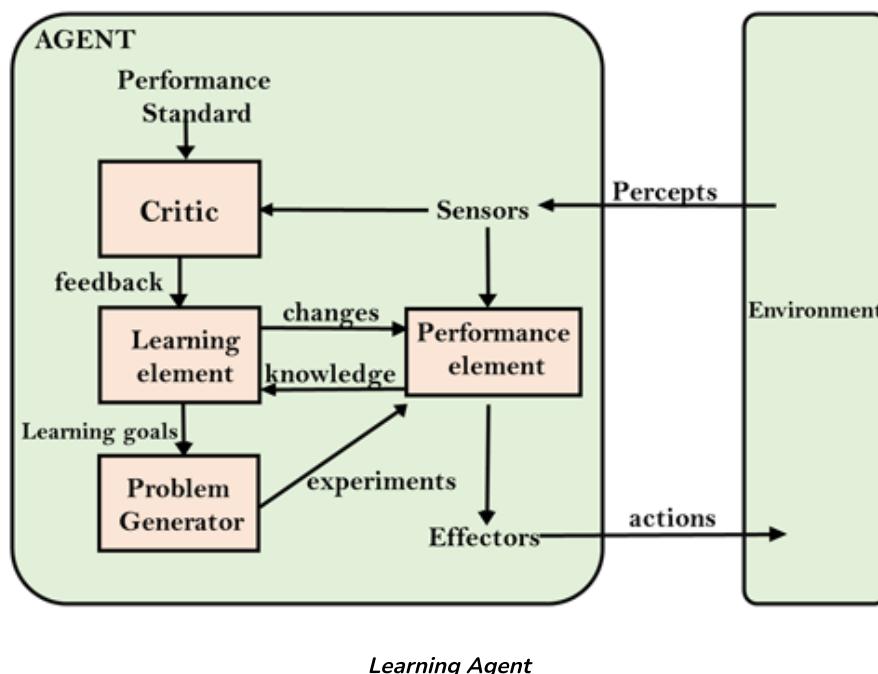
Learning Agent

A learning agent in AI is the type of agent that can learn from its past experiences or it has learning capabilities. It starts to act with basic knowledge and then is able to act and adapt automatically through learning. A learning agent has mainly four conceptual components, which are:

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

2. **Critic:** The learning element takes feedback from critics which describes how well the agent is doing with respect to a fixed performance standard.
3. **Performance element:** It is responsible for selecting external action.
4. **Problem Generator:** This component is responsible for suggesting actions that will lead to new and informative experiences.



Multi-Agent Systems

These agents interact with other agents to achieve a common goal. They may have to coordinate their actions and communicate with each other to achieve their objective.

A multi-agent system (MAS) is a system composed of multiple interacting agents that are designed to work together to achieve a common goal.

These agents may be autonomous or semi-autonomous and are capable of perceiving their environment, making decisions, and taking action to achieve the common objective.

MAS can be used in a variety of applications, including transportation systems, robotics, and social networks. They can help improve efficiency, reduce costs, and increase flexibility in complex systems. MAS can be

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

are cooperative or competitive, and whether the agents are homogeneous or heterogeneous.

- In a homogeneous MAS, all the agents have the same capabilities, goals, and behaviors.
- In contrast, in a heterogeneous MAS, the agents have different capabilities, goals, and behaviors.

This can make coordination more challenging but can also lead to more flexible and robust systems.

Cooperative MAS involves agents working together to achieve a common goal, while competitive MAS involves agents working against each other to achieve their own goals. In some cases, MAS can also involve both cooperative and competitive behavior, where agents must balance their own interests with the interests of the group.

MAS can be implemented using different techniques, such as [game theory](#), [machine learning](#), and agent-based modeling. Game theory is used to analyze strategic interactions between agents and predict their behavior. Machine learning is used to train agents to improve their decision-making capabilities over time. Agent-based modeling is used to simulate complex systems and study the interactions between agents.

Overall, multi-agent systems are a powerful tool in artificial intelligence that can help solve complex problems and improve efficiency in a variety of applications.

Hierarchical Agents

These agents are organized into a hierarchy, with high-level agents overseeing the behavior of lower-level agents. The high-level agents provide goals and constraints, while the low-level agents carry out specific tasks. Hierarchical agents are useful in complex environments with many tasks and sub-tasks.

- Hierarchical agents are agents that are organized into a hierarchy, with high-level agents overseeing the behavior of lower-level agents. The high-level agents provide goals and constraints, while the low-level

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

- Hierarchical agents can be implemented in a variety of applications, including robotics, manufacturing, and transportation systems. They are particularly useful in environments where there are many tasks and sub-tasks that need to be coordinated and prioritized.
- In a hierarchical agent system, the high-level agents are responsible for setting goals and constraints for the lower-level agents. These goals and constraints are typically based on the overall objective of the system. For example, in a manufacturing system, the high-level agents might set production targets for the lower-level agents based on customer demand.
- The low-level agents are responsible for carrying out specific tasks to achieve the goals set by the high-level agents. These tasks may be relatively simple or more complex, depending on the specific application. For example, in a transportation system, low-level agents might be responsible for managing traffic flow at specific intersections.
- Hierarchical agents can be organized into different levels, depending on the complexity of the system. In a simple system, there may be only two levels: high-level agents and low-level agents. In a more complex system, there may be multiple levels, with intermediate-level agents responsible for coordinating the activities of lower-level agents.
- One advantage of hierarchical agents is that they allow for more efficient use of resources. By organizing agents into a hierarchy, it is possible to allocate tasks to the agents that are best suited to carry them out, while avoiding duplication of effort. This can lead to faster, more efficient decision-making and better overall performance of the system.

Overall, hierarchical agents are a powerful tool in artificial intelligence that can help solve complex problems and improve efficiency in a variety of applications.

Uses of Agents

Agents are used in a wide range of applications in artificial intelligence, including:

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

lighting, and other systems in smart homes and buildings, optimizing energy use and improving comfort.

- **Transportation systems:** Agents can be used to manage traffic flow, optimize routes for autonomous vehicles, and improve logistics and supply chain management.
- **Healthcare:** Agents can be used to monitor patients, provide personalized treatment plans, and optimize healthcare resource allocation.
- **Finance:** Agents can be used for automated trading, fraud detection, and risk management in the financial industry.
- **Games:** Agents can be used to create intelligent opponents in games and simulations, providing a more challenging and realistic experience for players.
- **Natural language processing:** Agents can be used for language translation, question answering, and chatbots that can communicate with users in [natural language](#).
- **Cybersecurity:** Agents can be used for intrusion detection, malware analysis, and network security.
- **Environmental monitoring:** Agents can be used to monitor and manage natural resources, track climate change, and improve environmental sustainability.
- **Social media:** Agents can be used to analyze social media data, identify trends and patterns, and provide personalized recommendations to users.

Overall, agents are a versatile and powerful tool in [artificial intelligence](#) that can help solve a wide range of problems in different fields.

Here's a complete roadmap for you to become a developer: **Learn DSA -> Master Frontend/Backend/Full Stack -> Build Projects -> Keep Applying to Jobs**

And why go anywhere else when our [DSA to Development: Coding Guide](#) helps you do this in a single program! Apply now to our [DSA to](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

Got It !

[Privacy Policy](#)

207

[Suggest improvement](#)[Previous](#)[Next](#)

Data Structure for Dictionary and Spell Checker? Difference Between Machine Learning and Artificial Intelligence

[Share your thoughts in the comments](#)[Add Your Comment](#)

Similar Reads

[Difference Between Artificial Intelligence and Business Intelligence](#)[Difference Between Artificial Intelligence and Human Intelligence](#)[Top 5 Programming Languages For Artificial Intelligence](#)[Introduction to Hill Climbing | Artificial Intelligence](#)[Emergence Of Artificial Intelligence](#)[Integration of Artificial Intelligence and BlockChain](#)[Impacts of Artificial Intelligence in everyday life](#)[Game Playing in Artificial Intelligence](#)[Artificial intelligence vs Machine Learning vs Deep Learning](#)[Artificial Intelligence - Temporal Logic](#) [Sahil_Ban...](#)**Article Tags :** [AI-ML-DS](#) , [Artificial Intelligence](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

[Got It !](#)[Privacy Policy](#)



Company

- [About Us](#)
- [Legal](#)
- [Careers](#)
- [In Media](#)
- [Contact Us](#)
- [Advertise with us](#)
- [GFG Corporate Solution](#)
- [Placement Training Program](#)

Explore

- [Hack-A-Thons](#)
- [GfG Weekly Contest](#)
- [DSA in JAVA/C++](#)
- [Master System Design](#)
- [Master CP](#)
- [GeeksforGeeks Videos](#)
- [Geeks Community](#)

Languages

- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)
- [SQL](#)
- [R Language](#)
- [Android Tutorial](#)
- [Tutorials Archive](#)

DSA

- [Data Structures](#)
- [Algorithms](#)
- [DSA for Beginners](#)
- [Basic DSA Problems](#)
- [DSA Roadmap](#)
- [Top 100 DSA Interview Problems](#)
- [DSA Roadmap by Sandeep Jain](#)
- [All Cheat Sheets](#)

Data Science & ML

- [Data Science With Python](#)
- [Data Science For Beginner](#)
- [Machine Learning Tutorial](#)
- [ML Maths](#)
- [Data Visualisation Tutorial](#)
- [Pandas Tutorial](#)
- [NumPy Tutorial](#)

HTML & CSS

- [HTML](#)
- [CSS](#)
- [Web Templates](#)
- [CSS Frameworks](#)
- [Bootstrap](#)
- [Tailwind CSS](#)
- [SASS](#)

We use cookies to ensure you have the best browsing experience on our website. By

using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

Got It !

[Privacy Policy](#)

Python Tutorial

- [Python Programming Examples](#)
- [Python Projects](#)
- [Python Tkinter](#)
- [Web Scraping](#)
- [OpenCV Tutorial](#)
- [Python Interview Question](#)

Computer Science

- [Operating Systems](#)
- [Computer Network](#)
- [Database Management System](#)
- [Software Engineering](#)
- [Digital Logic Design](#)
- [Engineering Maths](#)

DevOps

- [Git](#)
- [AWS](#)
- [Docker](#)
- [Kubernetes](#)
- [Azure](#)
- [GCP](#)
- [DevOps Roadmap](#)

Competitive Programming

- [Top DS or Algo for CP](#)
- [Top 50 Tree](#)
- [Top 50 Graph](#)
- [Top 50 Array](#)
- [Top 50 String](#)
- [Top 50 DP](#)
- [Top 15 Websites for CP](#)

System Design

- [High Level Design](#)
- [Low Level Design](#)
- [UML Diagrams](#)
- [Interview Guide](#)
- [Design Patterns](#)
- [OOAD](#)
- [System Design Bootcamp](#)
- [Interview Questions](#)

JavaScript

- [JavaScript Examples](#)
- [TypeScript](#)
- [ReactJS](#)
- [NextJS](#)
- [AngularJS](#)
- [NodeJS](#)
- [Lodash](#)
- [Web Browser](#)

Preparation Corner

- [Company-Wise Recruitment Process](#)
- [Resume Templates](#)
- [Aptitude Preparation](#)
- [Puzzles](#)
- [Company-Wise Preparation](#)

School Subjects

- [Mathematics](#)
- [Physics](#)
- [Chemistry](#)
- [Biology](#)
- [Social Science](#)
- [English Grammar](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

[Got It !](#)

Management	Typing Test
HR Management	Image Editor
Finance	Code Formatters
Income Tax	Code Converters
Organisational Behaviour	Currency Converter
Marketing	Random Number Generator
	Random Password Generator

More Tutorials

Software Development
Software Testing
Product Management
SAP
SEO - Search Engine Optimization
Linux
Excel

GeeksforGeeks Videos

DSA
Python
Java
C++
Data Science
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !