

What is a model-based reflex agent?

Model-based reflex agents, defined

A model-based reflex agent is a type of [artificial intelligence \(AI\)](#) agent that maintains and refers to an internal model of the world. The internal model contains its memory of past perceptions and inferred knowledge about its environment. This model helps the agent make informed decisions in situations where it cannot perceive everything directly.

Model-based reflex agents are designed to operate in dynamic or partially observable environments. The agent's internal reference model updates as a result of the agent's actions, allowing the model to infer other aspects of its environment. In turn, the agent refers to its internal state to inform future actions.

Model-based reflex agents versus simple reflex agents

The difference between model-based reflex agents and [simple reflex agents](#) is that while both [types of AI agents](#) can use [AI agent perception](#) to react to changing environments in real time, model-based reflex agents have a persistent memory of their environment. Simple reflex agents are a less complex form of [agentic AI](#) that make decisions based solely on the environment's current state.

With only their current perception available, simple reflex agents are limited in terms of informed decision-making: they cannot recall previous information nor anticipate how their environment will change, either as a result of their own actions or other external factors.

Unlike simple reflex agents, model-based reflex agents maintain a symbolic internal state that records past perceptions and inferred environmental facts. The internal model allows them to handle partially observable or dynamic environments more effectively. These agents can predict how actions might cause their environment to change, enabling adaptive decision-making processes.

Industry newsletter

The latest AI trends, brought to you by experts

Get curated insights on the most important—and intriguing—AI news. Subscribe to our weekly Think newsletter. See the [IBM Privacy Statement](#).

We use your email to validate you are who you say you are, to create your IBMid, and to contact you for account related matters.

Business email

Your subscription will be delivered in English. You will find an unsubscribe link in every newsletter. You can manage your subscriptions or unsubscribe [here](#). Refer to our [IBM Privacy Statement](#) for more information.

Your subscription will be delivered in English. You will find an unsubscribe link in every newsletter. You can manage your subscriptions or unsubscribe [here](#). Refer to our [IBM Privacy Statement](#) for more information.

Model-based reflex agent components

Model-based reflex agents are composed of four primary components:

- Sensors
- Internal model
- Reasoning component
- Actuators

Sensors

Sensors allow the agent to perceive environmental stimuli and information. Sensory input can include direct physical information such as a camera feed or temperature reading, or virtual data such as API connections or simulated feeds. Without sensors, the agent would not be able to harvest and learn from environmental data.

Internal model

The internal model represents the agent's current understanding of its environment, including past experiences, inferred unseen states and predictions of how possible actions might change that environment.

Reasoning component

A reasoning component makes decisions based on the data in the internal model. The agent's reasoning is often rule-based, such as with the condition-action rule that uses predefined rules that tell the agent how to act.

The condition-action rule establishes a series of if-then statements that tell an agent what to do in a specific situation. For example, with a self-driving car, a condition-action rule might tell the agent to brake when an obstacle is detected in front of the autonomous vehicle. Or if the agent is controlling a smart home, it might turn on the air conditioning when the thermostat registers a specified temperature.

Other reasoning components use the state-action model, which outlines a system's possible states and any actions or events that might cause it to move between those states. In healthcare, state-action models can predict how a patient's condition might improve or worsen with possible treatments.

Actuators

Actuators, or performance elements, are the means by which the agent affects its environment. Actions can be physical or virtual, such as software outputs or API calls. In robotics, an actuator is the physical robot that the agent controls. In a smart home setting, one actuator is the home's climate control

system. For a virtual assistant, the actuators might be API connections to relevant tools, such as a calendar.

The model-based reflex agent workflow

Most model-based reflex agents operate according to a four-stage behavioral loop:

1. Sensing
2. Internal modeling
3. Decision-making
4. Action

After each action, the model restarts the loop, reassessing its environment, updating its model and making a new choice. These loops are explicitly non-learning: the agent's internal model is updated, but the agent's rules are not.

1. Sensing

The agent uses its sensors to perceive its environment and take in observable data. For example, an agent that operates a fleet of delivery bots can use their cameras to detect pedestrians and street traffic.

2. Internal modeling

Using the environmental data it perceives, the agent creates or updates the internal model of its real-world environment. The model uses observable data to infer the state of unobservable aspects of the environment. Internal models also provide potential actions for the agent to take and predict how those actions might affect its environment.

3. Decision-making

The agent uses the information in its internal model along with the logic of its reasoning component to decide on an action. If a delivery robot detects a walk signal and no oncoming traffic, it might decide to cross a street.

4. Action

In the action phase, the agent uses its actuators to carry out the decision made in the decision-making phase. Our delivery bot would engage its motor to send power to its wheels and cross the street, trundling along on its way.

Model-based reflex agent use cases

Model-based reflex agents are effective in environments that require continuous sensing and real-time decision-making, such as robotics, autonomous vehicles and real-time game AI.

With robotics and autonomous vehicles, model-based reflex agents navigate dynamic environments to minimize harm and guide the robot or vehicle to its destination. Game-playing agents can predict user actions and react to provide the degree of challenge needed by the game designer.

Large-scale [enterprise automation](#) use cases see model-based reflex agents maximizing efficiency, tweaking production and inventory in response to changing conditions.

Model-based reflex agents versus other types of agents

Reflex agents aren't the only types of agents out there. Other types of agents include:

- Goal-based agents
- Utility-based agents
- Learning agents
- Hierarchical agents

Goal-based agents

Goal-based agents work toward a specific goal. While reflex agents typically operate reactively under a set of if-then conditionals, goal-based agents use problem-solving techniques to figure out how to achieve specific objectives. The plan consists of a series of actions and specific tasks that are likely to lead to the completion of the goal.

Goal-based agents use search and planning [machine learning algorithms](#) such as A* or the Stanford Research Institute Problem Solver (STRIPS) to generate a sequence of actions intended to achieve the goal.

Utility-based agents

Utility-based agents are similar to goal-based agents but with a focus on efficiency. A utility function analyzes potential future states for maximum effectiveness and assigns a numerical value to outcomes to mitigate uncertainty. The optimal strategy is one that maximizes the expected utility of the agent's choices. Utility-based agents are focused on obtaining the best long-term outcomes.

Learning agents

Learning agents feature a learning element that optimizes the agent's behavior and decision-making processes based on the results of its past choices. A problem generator component proposes new exploratory strategies, helping the agent discover better behaviors rather than repeating old ones. These intelligent agents improve over time.

Learning agents use methods such as [supervised learning](#), [unsupervised learning](#) and [reinforcement learning](#) to refine their behavior.

For example, a learning agent asked to create a dynamic pricing system will analyze consumer response and demand to tweak future prices. Learning agents can adapt to a single user's preferences, using [natural language processing \(NLP\)](#) and a [chatbot](#) interface to process inputs and feedback.

Hierarchical agents

Hierarchical agents are used in [multi-agent systems](#): higher-level agents break complex tasks into subtasks, then assign them to lower-level agents to complete. Higher-level agents can be more complex [large language models \(LLMs\)](#), while lower-level agents use simpler architectures to carry out assignments. However, hierarchical agentic systems can use agents of any level of complexity.

The two groups of agents work together, one group's outputs feeding the other's inputs and vice versa. The [agentic workflow](#) coordinates the communication and cooperation between the tiers of agents in the AI system.

Model-based reflex agent limitations

While model-based reflex agents offer considerable flexibility, they aren't a good fit for every application. The limits of model-based reflex agents include:

- **Compute requirements:** Maintaining and continuously updating the internal model can consume significant computational resources, which can reduce the agent's ability to react quickly in time-sensitive environments.
- **Internal model dependence:** A model-based reflex agent is only as effective as its internal model. If that model diverges from real-world conditions, the agent may make suboptimal or even incorrect decisions.
- **Limited adaptability:** Model-based reflex agents cannot independently learn and update their ruleset. Should the environment change beyond the conditions the agent was programmed to handle, it will fail to adapt.
- **No long-term planning:** Model-based reflex agents are reactive by nature. Unlike goal-based or learning agents, they cannot develop long-term strategies to pursue broader objectives.
- **Single-focus purpose:** Model-based reflex agents are designed to achieve one specific function and cannot balance multiple or conflicting directives.