# Artificial Intelligence

AI2002

Rushda Muneer

Spring 2026

# Agents and Architectures

# What is an agent?

- An **agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**
- **Examples:**
- A **human agent** has eyes, ears, and other organs for sensors and hands, legs, vocal tract, and so on for actuators.
- A **robotic agent** might have cameras and infrared range finders for sensors and various motors for actuators.
- A **software agent** 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.

# What is an agent?

- We use the term **percept** to refer to the agent’s perceptual inputs at any given instant.
- An agent’s **percept sequence** is the complete history of everything the agent has ever perceived.
- In general, an agent’s choice of action at any given instant can depend on the entire percept sequence observed to date, but not on anything it hasn’t perceived.

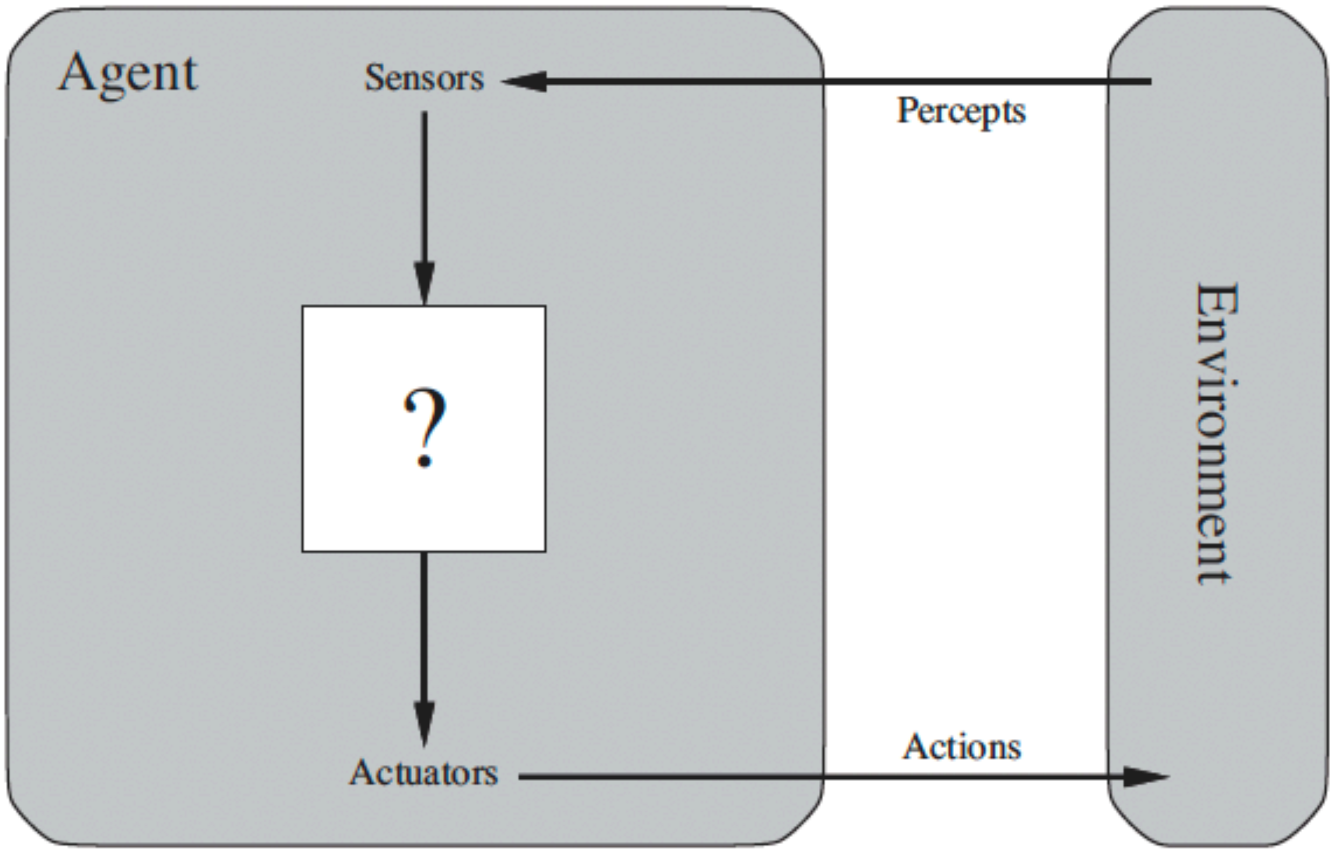


**Figure 2.1** Agents interact with environments through sensors and actuators.

# How to describe an Agent?

- What is the **Environment?**
- What type of **Sensors** it requires?
- Which **Actuators** are required?
- What **Percepts** it is getting via sensors from environment?
  - Percept Sequence
- **Agent Function (map percepts or percept sequence to action)?**
- Mathematically speaking, we say that **an agent's behavior is described** by the **agent function** that **maps any given percept sequence to an action.**
- **Agent Program**
- Internally, the agent function for **an artificial agent will be implemented by an agent program**.
- **Performance Measure:** that evaluated the effect of actions

# Example – Vacuum Agent

- This particular environment has just **two locations**: squares A and B.
- The vacuum agent **perceives** which square it is in and whether there is dirt in the square.
- It can choose to **move** left, move right, suck up the dirt, or do nothing.
- One very simple **agent function** is the following:
    - if the current square is dirty, then suck;
    - otherwise, move to the other square.

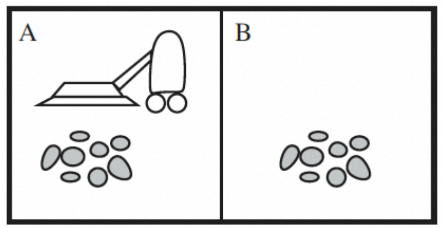


**Figure 2.2** A vacuum-cleaner world with just two locations.

# Example – Vacuum Agent

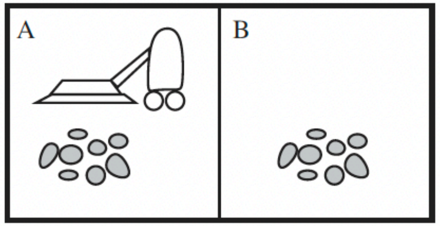


**Figure 2.2** A vacuum-cleaner world with just two locations.

Agent Function Table

| Percept sequence | Action |
|---|---|
| $[A, Clean]$ | *Right* |
| $[A, Dirty]$ | *Suck* |
| $[B, Clean]$ | *Left* |
| $[B, Dirty]$ | *Suck* |
| $[A, Clean], [A, Clean]$ | *Right* |
| $[A, Clean], [A, Dirty]$ | *Suck* |
| $\vdots$ | $\vdots$ |
| $[A, Clean], [A, Clean], [A, Clean]$ | *Right* |
| $[A, Clean], [A, Clean], [A, Dirty]$ | *Suck* |
| $\vdots$ | $\vdots$ |

**Figure 2.3** Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

# Rational Agent

- A rational agent is one that does the right thing—conceptually speaking, every entry in the table for the agent function is filled out correctly.
- If the sequence is desirable, then the agent has performed well. This notion of desirability is captured by a **performance measure** that evaluates any given sequence of environment states.
- As a general rule, it is better to design **performance measures** according to what one actually **wants in the environment**, rather than according to how one thinks the agent should behave.
- Example:
- A vacuum cleaner as a rational agent should not only clean dirt but sustain a clean environment.

# Rationality

- What is rational at any given time depends on **four things**:
  - The **performance measure** that defines the criterion of success.
  - The **agent's prior knowledge** of the environment.
  - The **actions** that the agent can perform.
  - The **agent's percept sequence** to date.
- **Rational Agent:** For each possible percept sequence, a rational agent should select an action that is expected to **maximize its performance measure**, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

# PEAS

(Performance, Environment, Actuators, Sensors)

# PEAS

- We must think about **task environments**, which are essentially the "problems" to which **rational agents** are the "solutions."
- To specify task environments and design rational agents **PEAS** (Performance, Environment, Actuators, Sensors) description is used.

| Agent Type | Performance Measure | Environment | Actuators | Sensors |
|---|---|---|---|---|
| Taxi driver | Safe, fast, legal, comfortable trip, maximize profits | Roads, other traffic, pedestrians, customers | Steering, accelerator, brake, signal, horn, display | Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard |

**Figure 2.4** PEAS description of the task environment for an automated taxi.

# Properties of Environment

- **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 we say that the task environment is **fully observable**.
- An environment might be **partially observable** because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data

# Properties of Environment

- **Single agent vs. multiagent:** The distinction between single-agent and multiagent environments depends on number of agents.
- **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 we say the environment is deterministic; otherwise, it is stochastic
- **Episodic vs. sequential:**
- In an episodic task environment, the agent's experience is divided into atomic episodes.
- In sequential environments, on the other hand, the current decision could affect all future decisions.

# Properties of Environment

- **Static vs. dynamic:** If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise, it is static.
- **Discrete vs. continuous:** The discrete/continuous distinction applies to the state of the environment, to the way time is handled, and to the percepts and actions of the agent
- **Known vs. unknown:**
- In a known environment, the outcomes (or outcome probabilities if the environment is stochastic) for all actions are given.
- If the environment is unknown, the agent will have to learn how it works in order to make good decisions

# Which combination of task environment is the most difficult for rational agents?

- The hardest case is partially observable, multiagent, stochastic, sequential, dynamic, continuous, and unknown.

# Structure of Agents

- *agent = architecture + program*
- Four basic kinds of agent programs that embody the principles underlying almost all intelligent systems:
  - Simple reflex agents;
  - Model-based reflex agents;
  - Goal-based agents; and
  - Utility-based agents.

# Simple reflex agents

- The simplest kind of agent is the simple reflex agent.
- These agents select actions on the basis of the **current percept**, ignoring the rest of the percept history.
- Example:
  - Vacuum agent

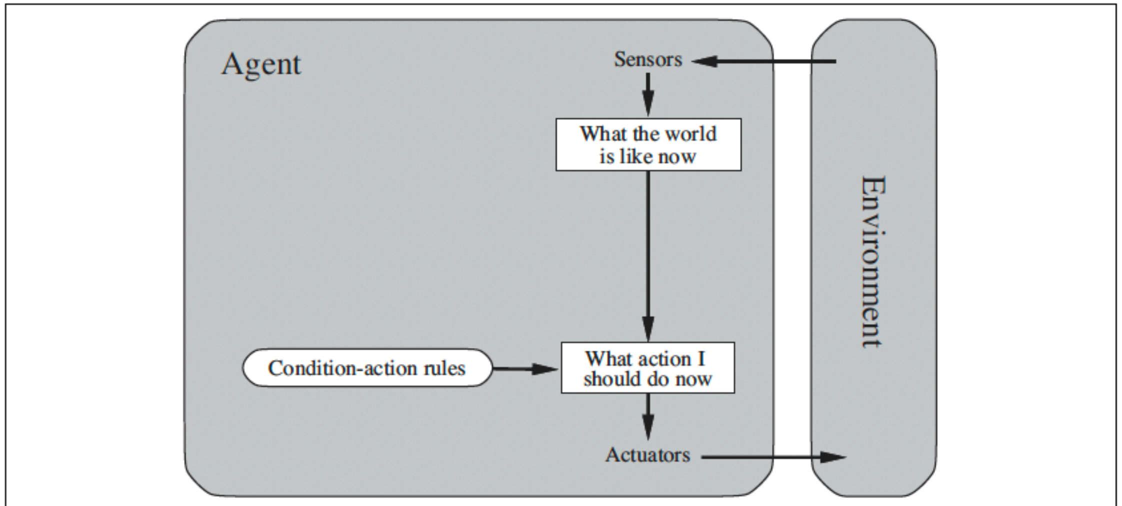


**Figure 2.9** Schematic diagram of a simple reflex agent.

# Model-based reflex agent

- Learns knowledge about “how the world works”—called a model of the world. An agent that uses such a model is called a model-based agent.
- It keeps track of the current state of the world, using an internal model.
- It then chooses an action in the same way as the simple reflex agent.
- Example:
  - Self-driving cars recognizing road conditions

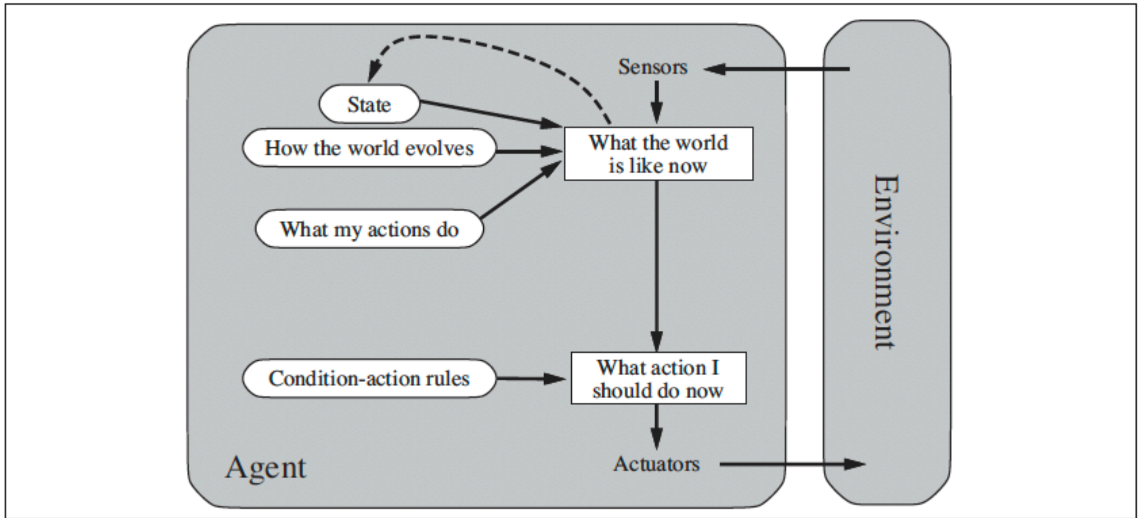


**Figure 2.11** A model-based reflex agent.

# Goal-based agents

- As well as a current state description, the agent needs some sort of goal information that describes situations that are desirable.
- Example:
- Chess-playing AI planning moves.

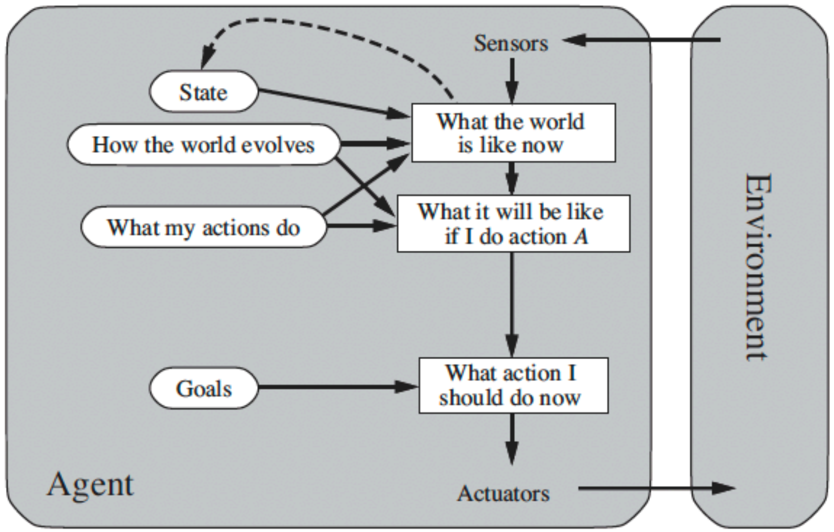


**Figure 2.13** A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

# Utility-based agents

- An agent's utility function is essentially an internalization of the performance measure.
- If the internal utility function and the external performance measure are in agreement, then an agent that chooses actions to maximize its utility will be rational according to the external performance measure.
- Example: Stock trading AI optimizing portfolio gains.

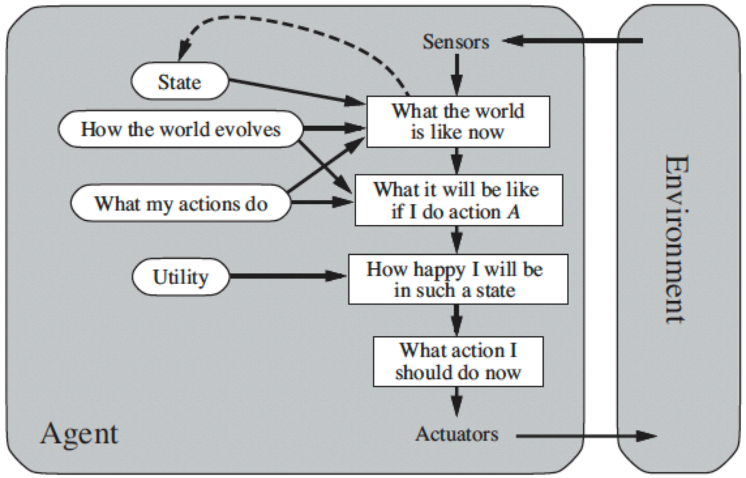


**Figure 2.14** A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.

# Learning agents

- Learns from the environment and feedback on actions.
- Example: AI personal assistants improving recommendations over time.

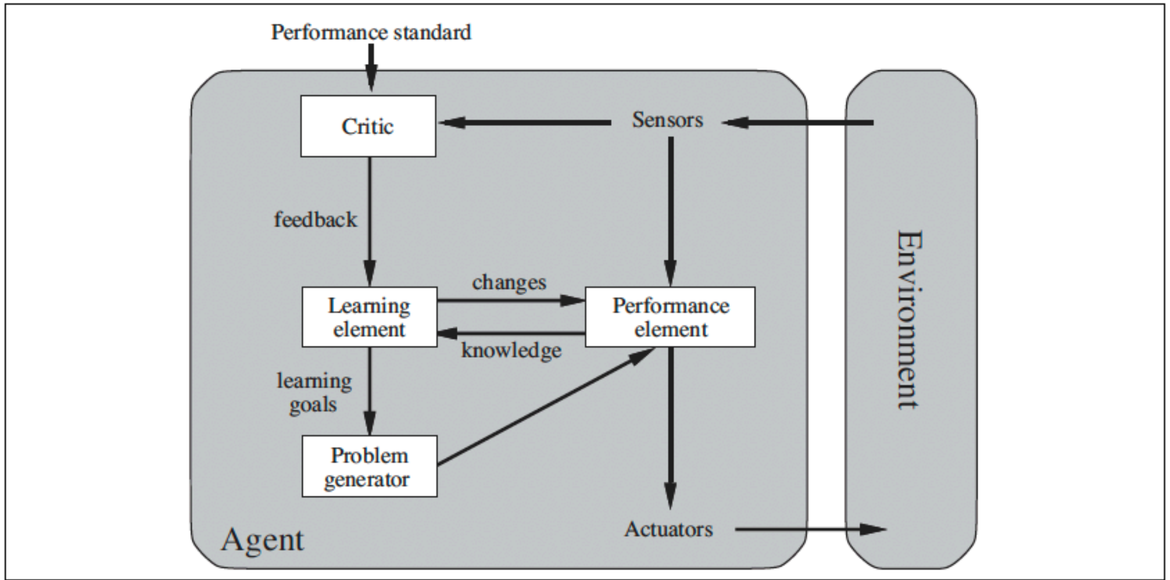


**Figure 2.15** A general learning agent.

# Homework

Read Chapter 2