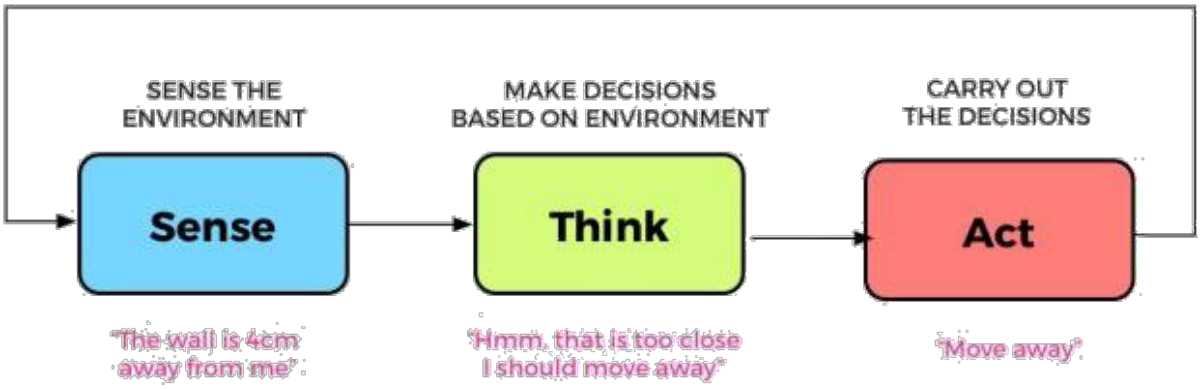
1 AI Systems: Agents and Environments

Definition: AI System

An AI system is defined as the study of (rational) **agents** and their **environments**. The system has two main parts:

1. **Agent:** Anything that can be viewed as perceiving its environment through **sensors** and acting upon that environment through **actuators**.
2. **Environment:** The surroundings or conditions in which the agent lives or operates. This can be real (e.g., streets for a self-driving car) or artificial (e.g., a chessboard).

The agent follows a continuous **Sense** → **Think** → **Act** loop.



1.1 Rationality

Rationality

- A **rational agent** is one that "does the right thing".
- A **rational action** is one that maximizes the agent's performance and yields the best positive outcome.
- **Key Point:** Rationality maximizes **expected** performance, not necessarily the *optimal* outcome. E.g., not playing the lottery is rational (positive expected outcome), even if playing could lead to the optimal outcome (winning).
- Rationality is **not** omniscient. An omniscient agent would know the *actual* outcome of its actions, which is impossible in reality.

A **performance measure** is a function that evaluates a sequence of actions.

General Rule for Design

Design the performance measure based on the **desired outcome**, not the desired agent behaviour.

1.2 Characteristics of Environments

The design of an agent heavily depends on the type of environment it operates in. Environments are characterized along several key dimensions.

Environment Dimensions

- **Discrete vs. Continuous:** Does the environment have a limited, countable number of distinct states (e.g., chess) or is it continuous (e.g., position and speed of a self-driving car)?
- **Observable vs. Partially/Unobservable:** Can the agent's sensors determine the *complete* state of the environment at each time point? If not, it is **partially observable** (e.g., a taxi cannot know pedestrian intentions, poker agent cannot see opponent's cards).
- **Static vs. Dynamic:** Does the environment change while the agent is acting/deliberating? A crossword puzzle is **static**; taxi driving is **dynamic** (other cars move).
- **Single Agent vs. Multiple Agents:** Is the agent operating by itself? Or does the environment contain other agents (e.g., other drivers, poker players)?
- **Accessible vs. Inaccessible:** Can the agent obtain *complete and accurate* information about the environment's state?
- **Deterministic vs. Non-deterministic (Stochastic):** Is the next state of the environment completely determined by the current state and the agent's action? Chess is **deterministic**. A self-driving car is **non-deterministic** (turning the wheel can have slightly different effects due to road friction, wind, etc.).
- **Episodic vs. Sequential:** In an **episodic** environment, the agent's experience is divided into "episodes". The quality of its action depends only on the current episode (perceive → act). In a **sequential** environment, the agent requires memory of past actions to make the best decision.

Key Distinction: Observable vs. Accessible

- **Accessibility** concerns the environment itself: whether the information exists and can *in principle* be obtained.
- **Observability** concerns the agent's *sensors*: whether they can actually perceive that information.

Environment	Discrete?	Observable?	Static?	Single Agent?	Accessible?	Deterministic?	Episodic?
Chess	Discrete	Observable	Static	Multi-Agent	Accessible	Deterministic	Sequential
Solitaire	Discrete	Observable	Static	Single Agent	Accessible	Deterministic	Sequential
Poker	Discrete	Partially Observable	Static	Multi-Agent	Partially Accessible	Stochastic	Sequential
Self-Driving	Continuous	Partially Observable	Dynamic	Single Agent	Inaccessible	Stochastic	Sequential
Medical Diagnosis	Discrete	Partially Observable	Static	Single Agent	Inaccessible	Stochastic	Episodic

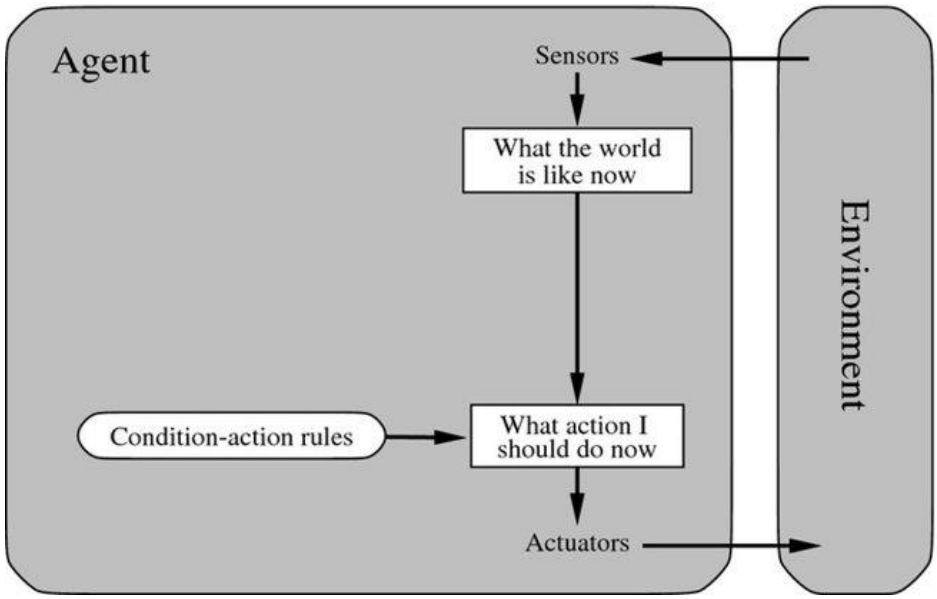
Characteristics of various environments

1.3 Types of Agents

Agents are categorized based on their perceived intelligence and complexity.

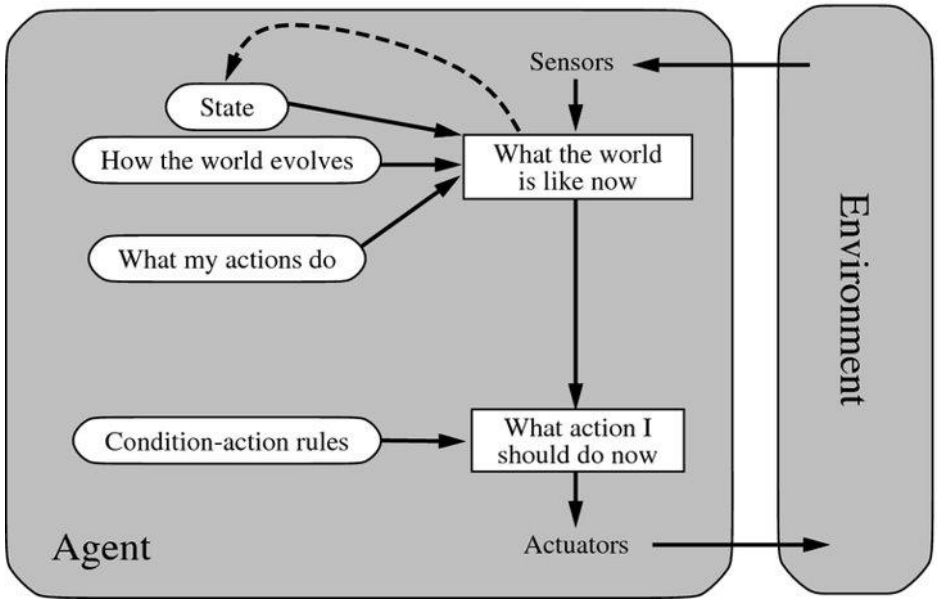
1.3.1 Reflex Agent

- Selects actions based **only on the current percept**, ignoring the percept history.
- Implemented with simple **condition-action rules**.
- Example: A thermostat (IF temp $< 20°C$ → turn on heater).
- **Problem:** Very limited. No knowledge of anything it cannot actively perceive.



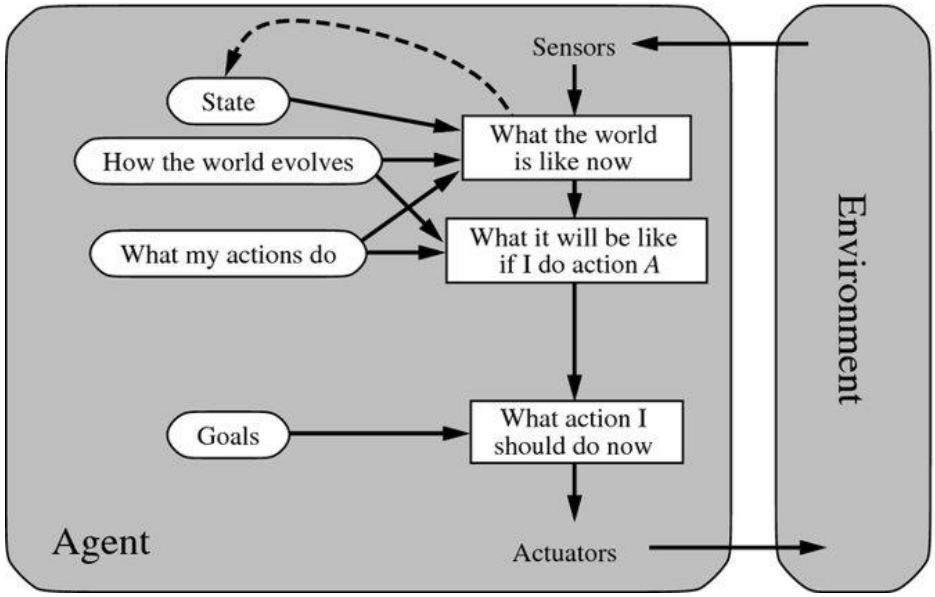
1.3.2 Model-based Agent

- These agents **keep track of the world state**.
- They maintain an **internal state (a world model)** that describes how the world evolves and how the agent's actions affect it.
- This allows the agent to handle partially observable environments.
- Example: A warehouse robot tracking inventory positions.



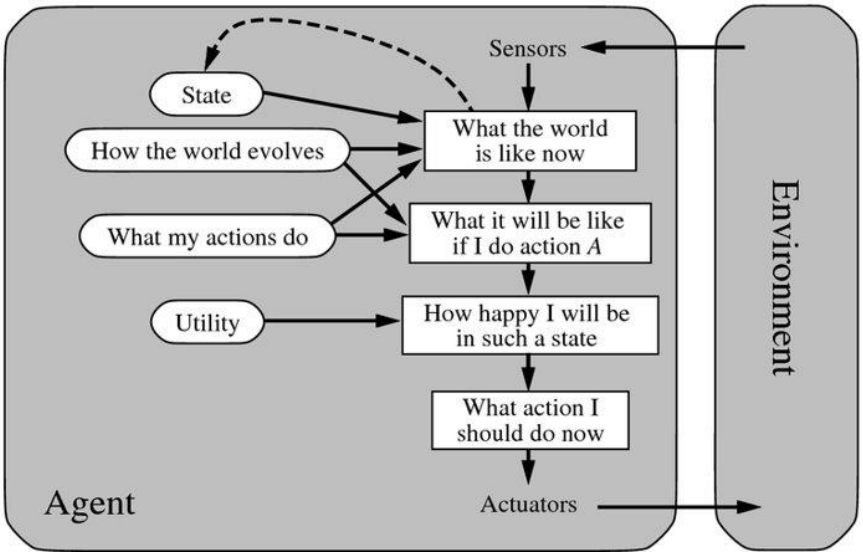
1.3.3 Goal-based Agent

- Builds on a model-based agent, but also knows what states are **desirable** (i.e., it has **goals**).
- This allows the agent to make decisions by considering the future, asking "What will happen if I do action A?" and "Will that action achieve my goal?".
- Example: A chess agent whose goal is to checkmate the opponent.



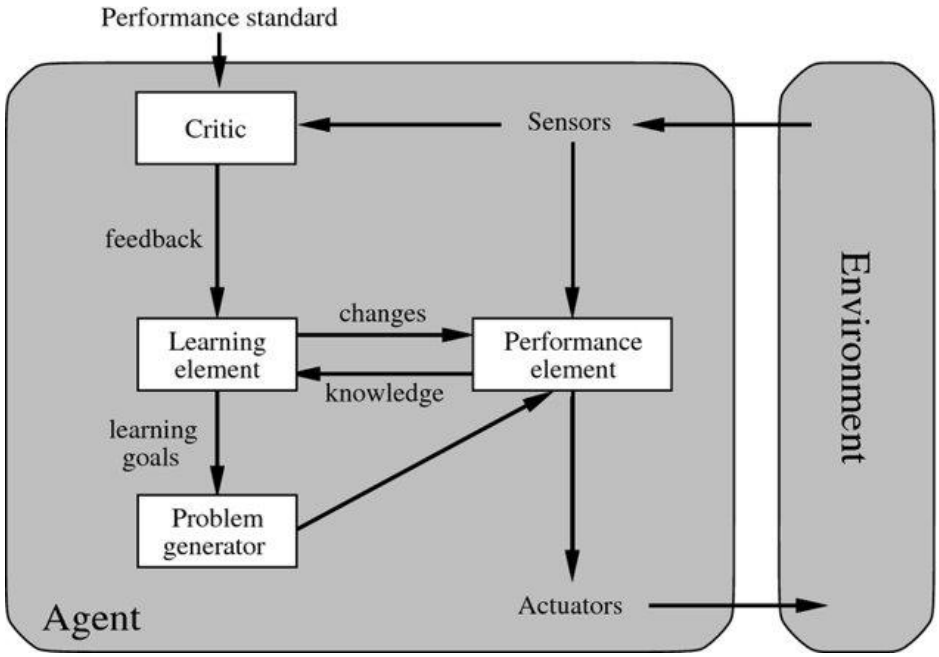
1.3.4 Utility-based Agent

- Goals provide a binary distinction (achieved / not-achieved). A **utility function** provides a continuous scale, rating each state based on the desired result ("how happy" the agent is).
- This is crucial for resolving **conflicting goals** (e.g., is speed or safety more important for a self-driving car?).
- Allows the agent to choose the action that maximizes its **expected utility**.



1.3.5 Learning Agent

- Employs a **learning element** to gradually improve and become more knowledgeable over time.
- Can learn from its past experiences and adapt automatically.
- More robust in unknown environments.



Four Components of a Learning Agent

1. **Learning Element:** Responsible for making improvements by learning from the environment.
2. **Critic:** Gives feedback on how well the agent is doing with respect to a fixed performance standard.
3. **Performance Element:** Responsible for selecting actions (this is the "agent" part).
4. **Problem Generator:** Responsible for suggesting actions that will lead to new (and potentially informative) experiences.

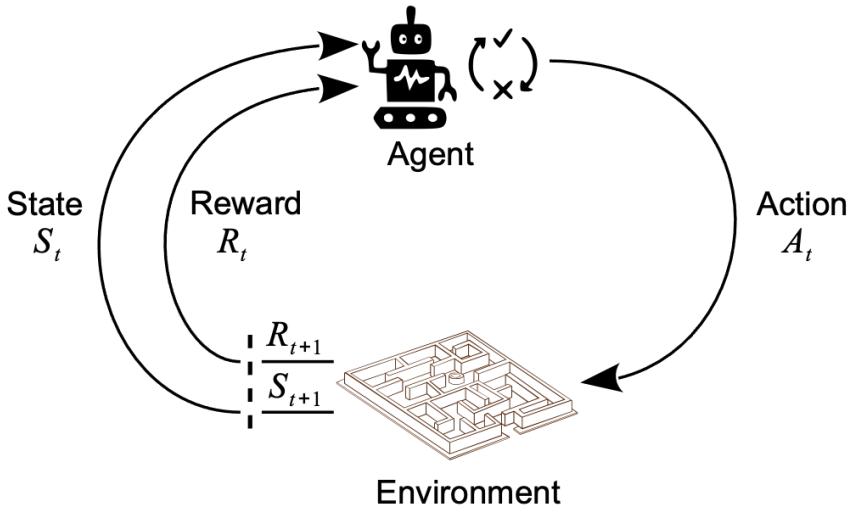
Agent Types Summary

- **Reflex agent:** reacts.
- **Model-based agent:** remembers.
- **Goal-based agent:** plans.
- **Utility-based agent:** optimizes.
- **Learning agent:** improves itself over time.

1.4 How to Make Agents Intelligent

There are several high-level approaches to selecting intelligent actions:

- **Search Algorithms:** Understand "finding a good action" as a search problem and use tree-based algorithms to find a solution (path to a goal).
- **Reinforcement Learning (RL):** Based on trial and error, similar to animal conditioning. The agent receives **rewards** (positive) or **pain/punishments** (negative) from the environment and learns to choose actions that maximize its cumulative reward.



- **Genetic Algorithms (GAs):** Inspired by Darwinian evolution ("survival of the fittest"). A **population** of agents is generated, evaluated by a **performance function**, and the best ones are "bred" (using **crossover** and **mutation**) to create a new, potentially better, generation.