Introduction to Artificial Intelligence

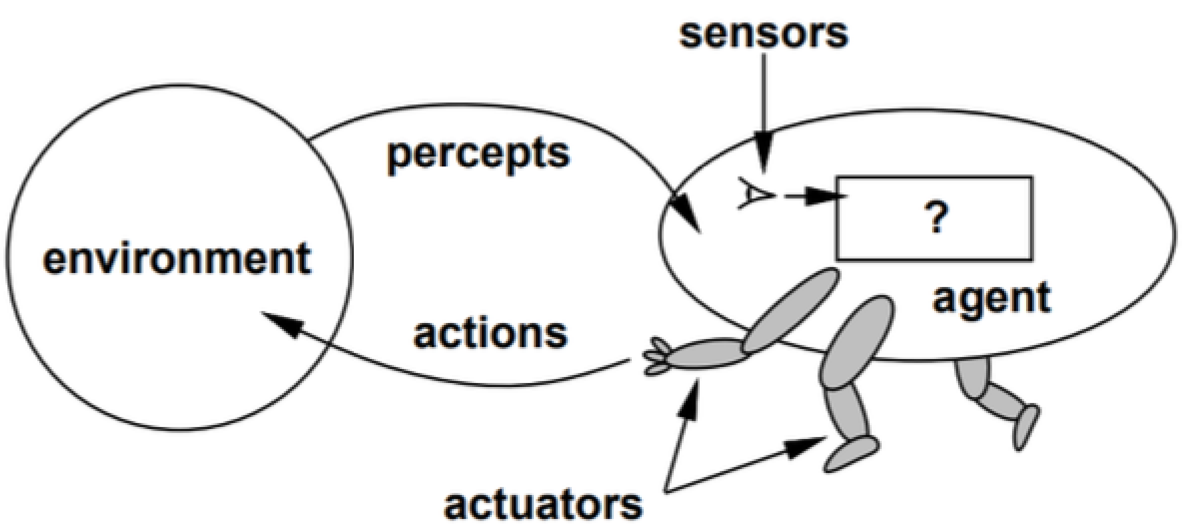
Lecture: Intelligent Agents

Outline

- Agents and Environments
- The Concept of Rationality
- The Nature of Environments
- The Structures of Agents

What is agent?

- AI studies how to make computers do things that people are better at if they could.
- Such systems are called Agents.
- An agent **perceives** its environment through **sensors** and **acts** upon that environment through **actuators.**



Examples of Agents

Agent	Sensors	Actuators
Human agent	eyes, ears, and other organs.	hands, legs, vocal tract, etc.
Robotic agent	cameras, infrared range finders, etc.	levels, motors, etc.
Software agent	keystrokes, file contents, network packets, etc.	displaying on screen, writing files, sending network packets, etc.

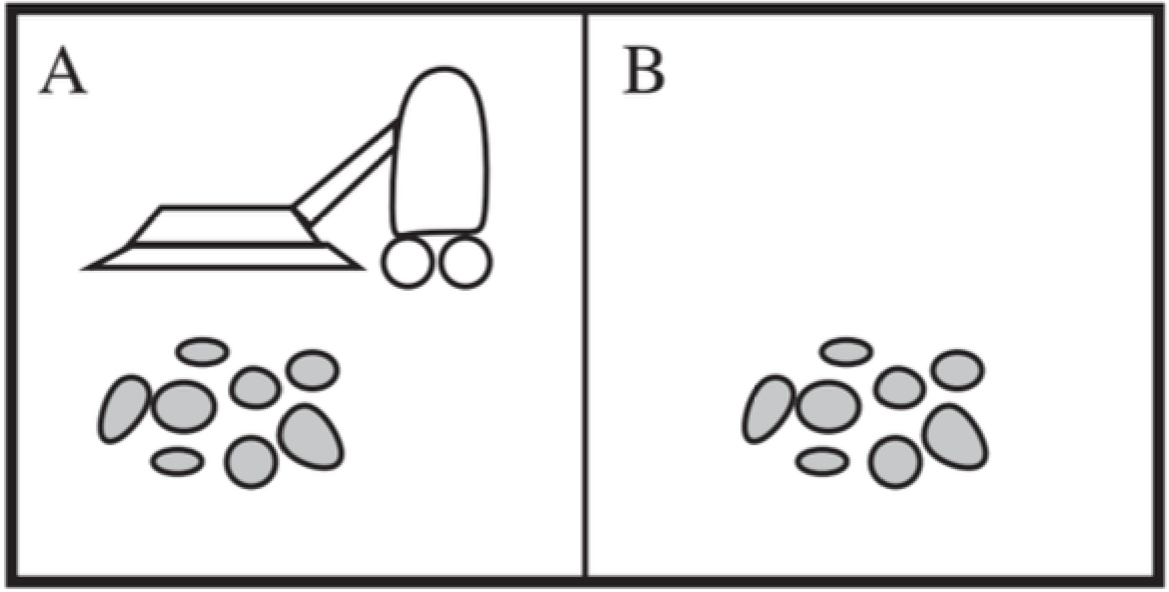
Agent’s Behavior

- Percept: the agent’s perceptual inputs at any given instant
- Percept sequence: the complete history of everything the agent has ever perceived
- An agent’s behavior is described by the agent function that maps any given percept sequence to an action.

$$f: P^* \rightarrow A$$

Example: Vacuum Agent

- Percepts: location and contents, e.g., [A,Dirty]
- Actions: Left, Right, Suck, Do Nothing



Vacuum Agent

Percept sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
…	…
[A, Clean], [A, Clean], [A, Clean]	Right
[A, Clean], [A, Clean], [A, Dirty]	Suck

Vacuum Agent

```
function VACUUM-AGENT([location,status])
returns an action
   if status = Dirty then return Suck
   else if location = A then return Right
   else if location = B then return Left
```

Importance of agents

- A tool for analyzing systems
- All areas of engineering can be seen as designing artifacts that interact with the world.
- AI designs artifacts that have significant computational resources and the task environment requires nontrivial decision making.

Rational agents

- A rational agent is one that does the right thing.
- We need ways to measure success → Performance measure.
- Performance measure evaluates any given sequence of environment states.
- General rule: Design performance measures according to
    - What one actually wants in the environment
    - Not how one thinks the agent should behave
- Example: vacuum agent
    - The amount of dirt cleaned up in a single eight-hour shift
    - The floor clean, no matter how the agent behaves

Rationality

- Performance measure
  - Define the criterion of success
- Prior knowledge
  - What the agent knows about the environment
- Percept sequence
  - The agent’s percept to date
- Actions
  - What the agent can perform

Definition of rational agents

- 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.

Example: Vacuum machine

- Performance measure
  - Award one point for each clean square at each time step, over 10000 timesteps
- Prior knowledge about the environment
  - The geography of the environment (2 squares)
  - The effect of the actions
- Actions that can perform
  - Left, Right, Suck and Do Nothing
- Percept sequences
  - Where is the agent?
  - Whether the location contains dirt?
- The agent is rational.

Omniscience, Learning, Autonomy

Omniscience	• Knows the actual outcome of its actions in advance • No other possible outcomes • Impossible in real world
Learning	• A rational agent must learn as much as possible from what it perceives
Autonomy	• A rational agent should be autonomous – Learn what it can to compensate for partial or incorrect prior knowledge.

Examples

- [Hide and seek]
- [Apple collector]
- https://youtu.be/shccS5kJvtQ
- https://youtu.be/FIz8ycRCSw0

Task environment

- Task environments are essentially the "problems" to which rational agents are the "solutions".
- PEAS:
  - **P**erformance measure
  - **E**nvironment
  - Agent's **A**ctuators
  - Agent's **S**ensors
- In designing an agent, the first step must always be to specify the task environment (PEAS) as fully as possible.

Example: Automated taxi driver

- Agent type: taxi driver
- Performance measure: safe, fast, legal, comfortable trip, maximize profits
- Environment: roads, other traffic, pedestrians, passengers.
- Sensors: cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, etc.
- Actuators: steering, accelerator, brake, signal, horn, display.

Properties of Task Environment

Fully observable	Partially observable
Single agent	Multiagent
Deterministic	Stochastic
Episodic	Sequential
Static	Dynamic
Discrete	Continuous
Known	Unknown

Properties of Task Environment

- Fully observable: The agent’s sensory gives it access to the complete state of the environment.
  - The agent need not maintain internal state to keep track of the world.
- Partially observable
  - Noisy and inaccurate sensors
  - Parts of the state are simply missing from the sensor data, e.g., a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares
- Unobservable: The agent has no sensors at all

Properties of Task Environment

- Single agent: An agent operates by itself in an environment.
  - E.g., solving crossword → single-agent,
  - playing chess → two-agent
- Competitive vs. Cooperative multiagent environment
  - E.g., playing chess → competitive,
  - driving on road → cooperative

Properties of Task Environment

- Deterministic: The next state of the environment is completely determined by the current state and the action executed by the agent.
  - E.g., the vacuum world → deterministic,
  - driving on road → stochastic
- Most real situations are so complex that they must be treated as stochastic.

Properties of Task Environment

- Episodic: The agent’s experience is divided into atomic episodes, in each of which the agent receives a percept and then performs a single action.
  - Quality of action depends just on the episode itself
  - Do not need to think ahead
- Sequential: A current decision could affect future decisions.
  - E.g., spotting defective parts on an assembly line vs. playing chess

Properties of Task Environment

- Static: The environment is unchanged while an agent is deliberating.
  - E.g., crossword puzzles → static, taxi driving → dynamic
- Dynamic: The agent is continuous asked what it wants to do
  - If it has not decided yet, that counts as deciding to do nothing.
- Semi-dynamic: The environment itself does not change with the passage of time but the agent's performance score does
  - E.g., chess playing with a clock

Properties of Task Environment

- Discrete vs. continuous
  - The distinction applies to the state of the environment, to the way time is handled, and to the agent’s percepts and actions
  - E.g., the chess has a finite number of distinct states, percepts and actions; while the vehicles’ speeds and locations sweep through a range of continuous values smoothly over time.
- Known vs. unknown
  - Known environment: the outcomes (or outcome probabilities if the environment is stochastic) for all actions are given.
  - Unknown environment: the agent needs to learn how it works to make good decisions.

Structures of Agents

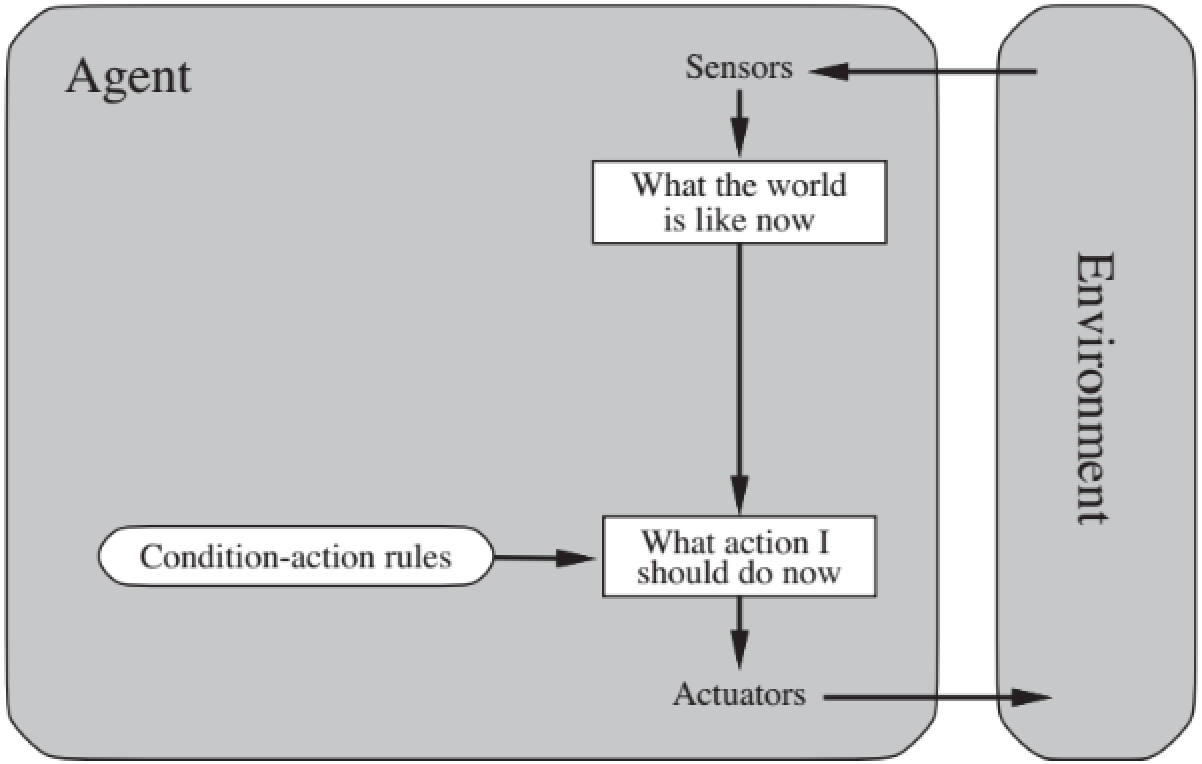
- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents

Simple reflex agents

- Select actions based on the current percept, ignoring the rest of the percept history
- The connection from percept to action is represented by condition-action rules.

IF current percept **THEN** action

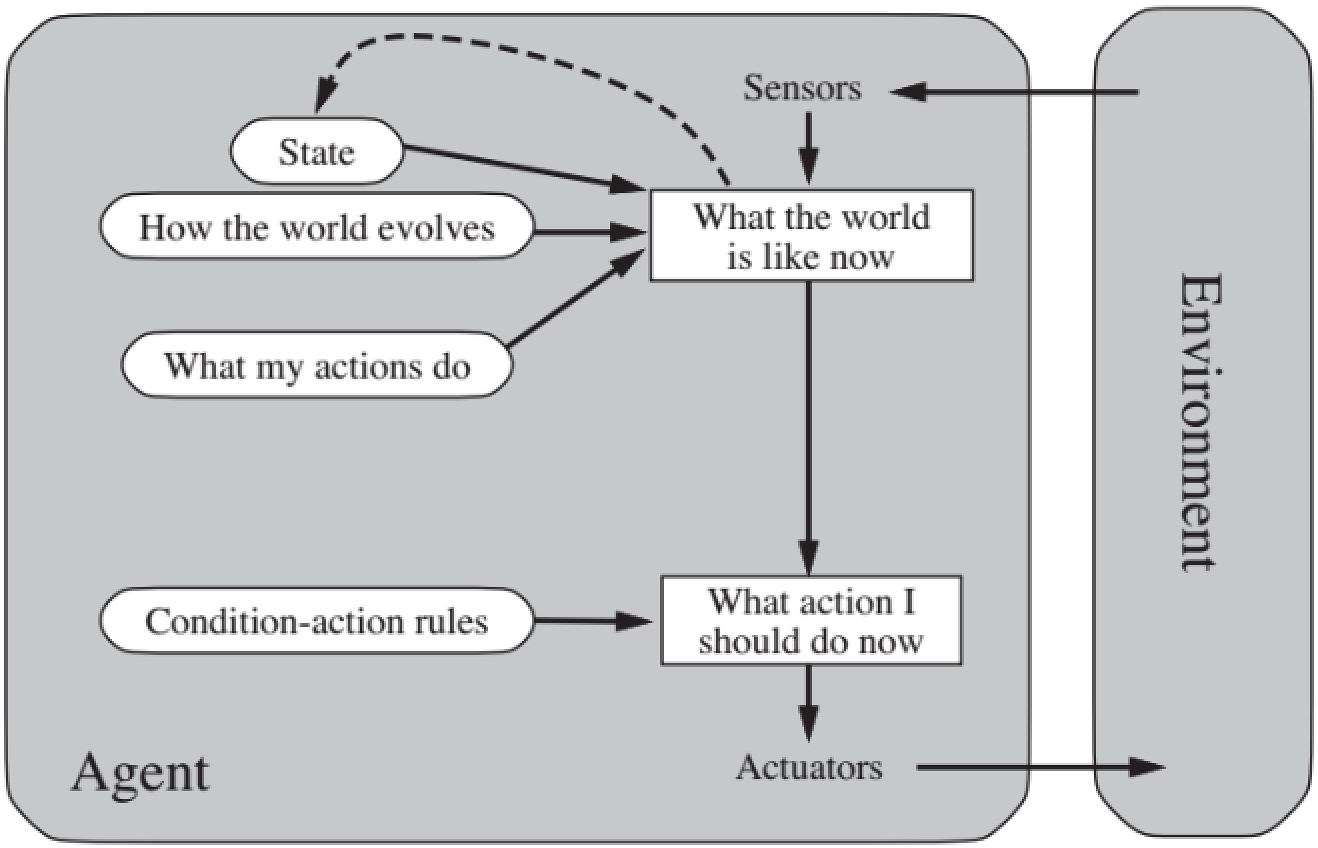
Simple reflex agents



Model-based reflex agents

- Partially observable → the agent must keep track of an internal state
  - It depends on the percept history and reflects some of the unobserved aspects
  - E.g., driving a car and changing lane
- The agent program updates the internal state information as time goes by by encoding two kinds of knowledge
  - How the world evolves independently of the agent
  - How the agent's actions affect the world

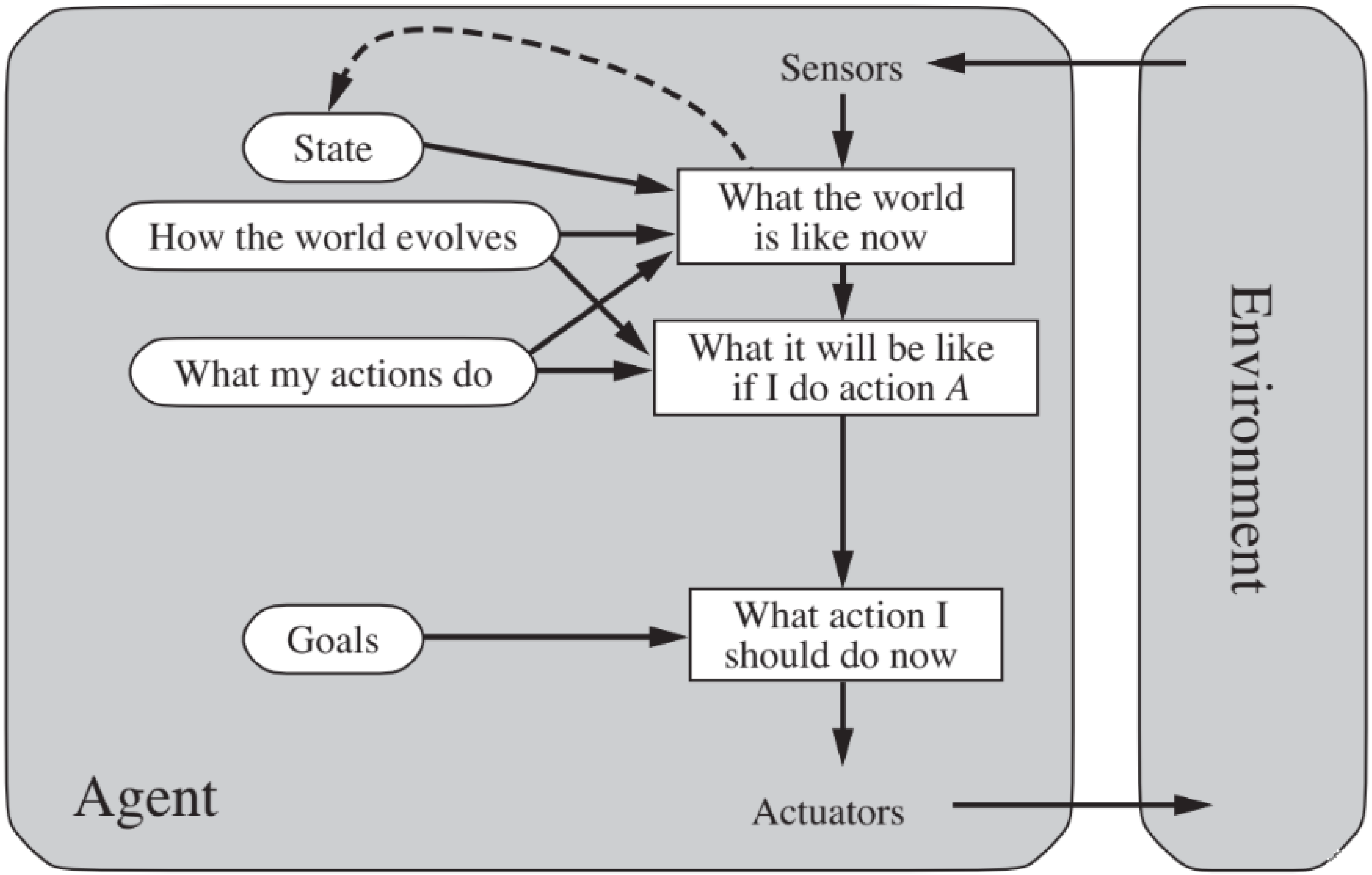
Model-based reflex agents



Goal-based agents

- Current state of the environment is always not enough
- The agent further needs some sort of goal information that describes desired situations.
    - E.g., at a road junction, the taxi can turn left, turn right, or go straight on, depending on where the taxi is trying to get to.
- Less efficient but more flexible
    - Knowledge that supports the decisions is represented explicitly and can be modified

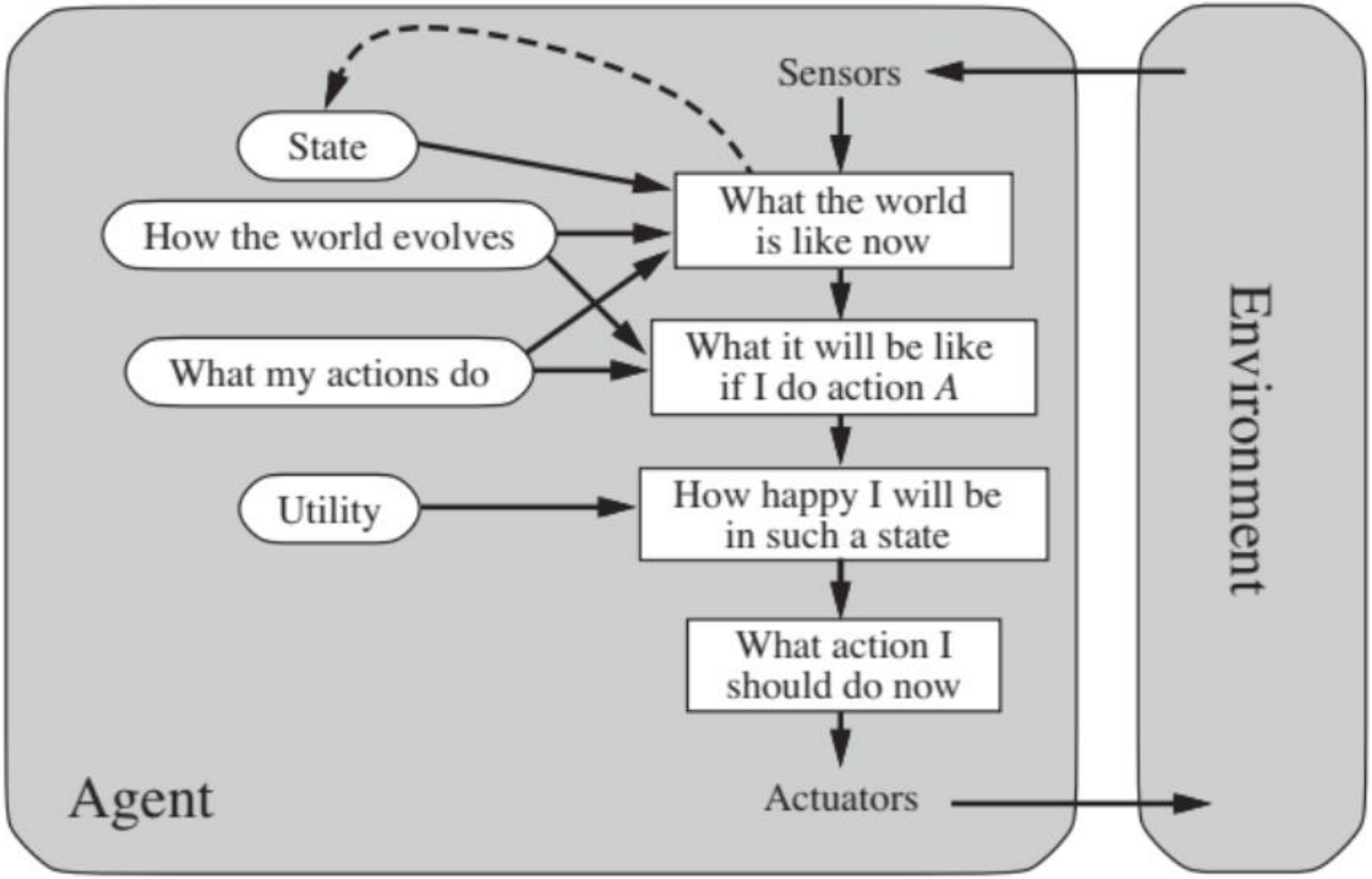
Goal-based agents



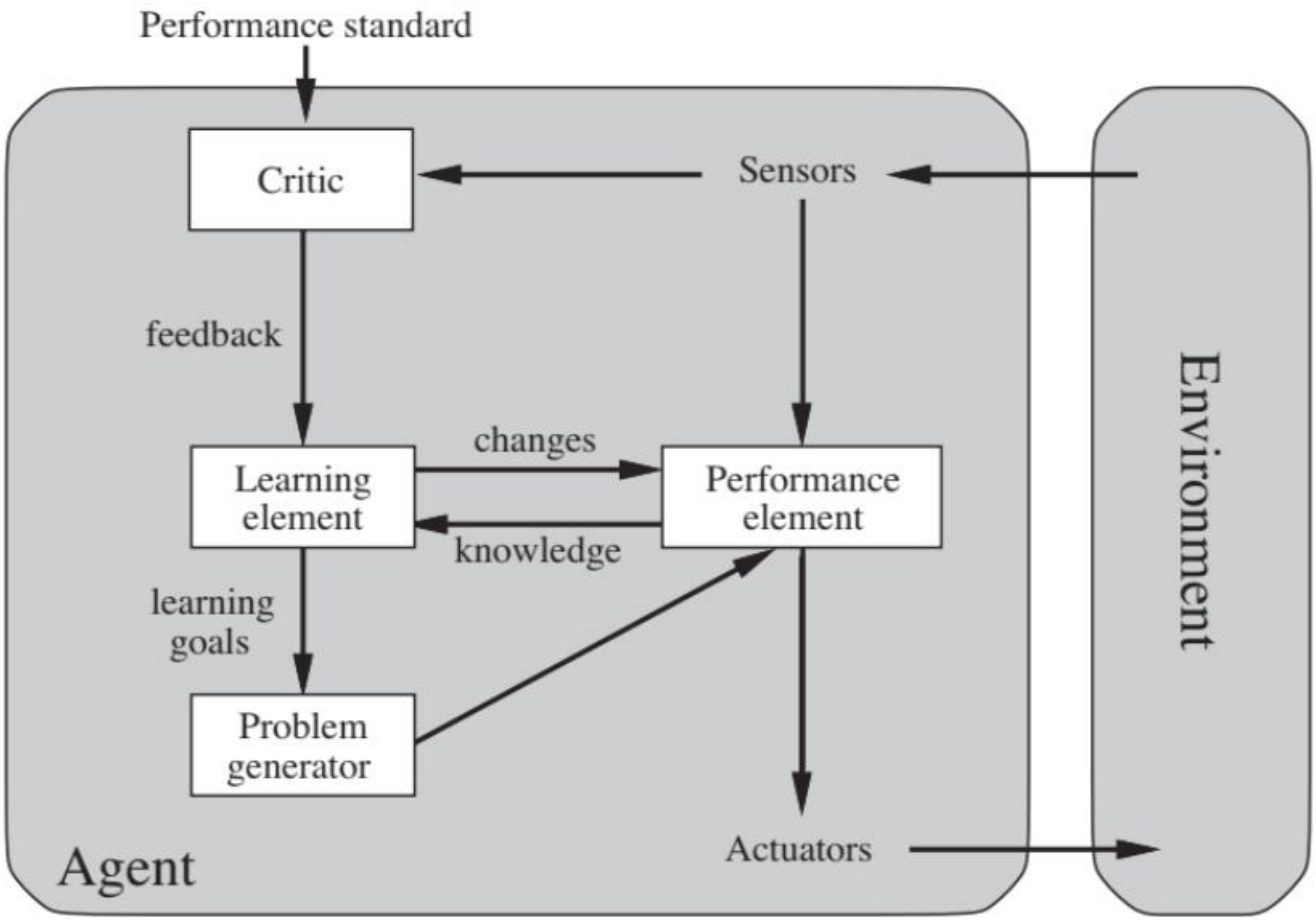
Utility-based agents

- Goals alone are inadequate to generate high-quality behavior in most environments
    - Many action sequences can get the goals, some are better, and some are worse
    - E.g., go home: Vinasun taxi or Grab car?
- An agent's utility function is essentially an internalization of the performance measure.
    - Goal → success, utility → degree of success (how successful it is)
    - If state A is more preferred than others, then A has higher utility

Utility-based agents



Learning Agents



Learning Agents

- Learning in intelligent agents is a process of modification of each component of the agent to bring the components into closer agreement with the available feedback information, thereby improving the overall performance of the agent.
    - Learning element → Making improvement
    - Performance element → Selecting external actions
    - Critic → Tells the Learning element how well the agent is doing with respect to fixed performance standard. (Feedback from user or examples, good or not?)
    - Problem generator → Suggest actions that will lead to new and informative experiences.

References

- Stuart Russell and Peter Norvig. 2009. Artificial Intelligence: A Modern Approach (3rd ed.). Prentice Hall Press, Upper Saddle River, NJ, USA.
- Lê Hoài Bắc, Tô Hoài Việt. 2014. Giáo trình Cơ sở Trí tuệ nhân tạo. Khoa Công nghệ Thông tin. Trường ĐH Khoa học Tự nhiên, ĐHQG-HCM.
- Nguyễn Ngọc Thảo, Nguyễn Hải Minh. 2020. Bài giảng Cơ sở Trí tuệ Nhân tạo. Khoa Công nghệ Thông tin. Trường ĐH Khoa học Tự nhiên, ĐHQG-HCM.