Artificial Intelligence

Hai Thi Tuyet Nguyen

Nội dung môn học

CHAPTER 1: INTRODUCTION (CHAPTER 1)

CHAPTER 2: INTELLIGENT AGENTS (CHAPTER 2)

CHAPTER 3: SOLVING PROBLEMS BY SEARCHING (CHAPTER 3)

CHAPTER 4: INFORMED SEARCH (CHAPTER 3)

CHAPTER 5: LOGICAL AGENT (CHAPTER 7)

CHAPTER 6: FIRST-ORDER LOGIC (CHAPTER 8, 9)

CHAPTER 7: QUANTIFYING UNCERTAINTY(CHAPTER 13)

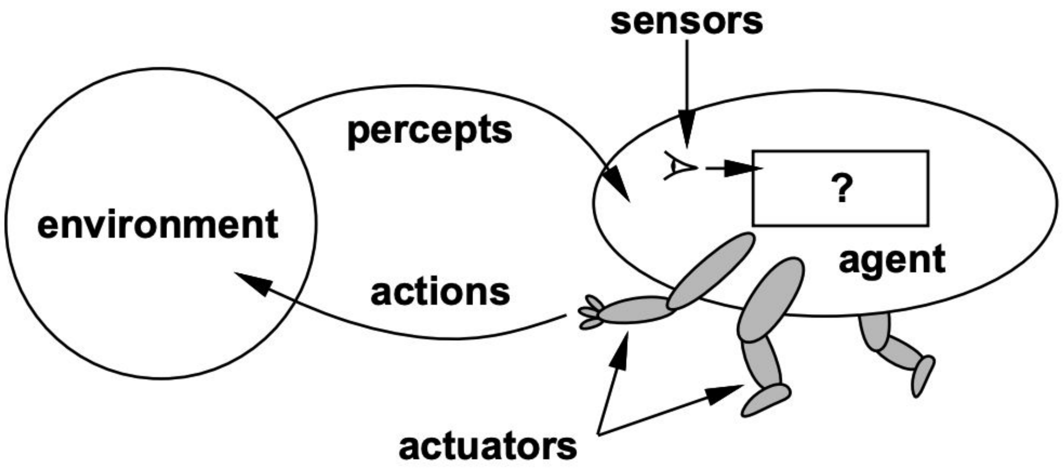
CHAPTER 8: PROBABILISTIC REASONING (CHAPTER 14)

CHAPTER 9: LEARNING FROM EXAMPLES (CHAPTER 18)

CHAPTER 2: INTELLIGENT AGENTS

2.1 Agents And Environments

- An agent is something that
  - **perceives** its environment through **sensors**
  - **acts** upon that environment through **actuators**.
- A human agent has eyes, ears, … for sensors and hands, legs, … for actuators.

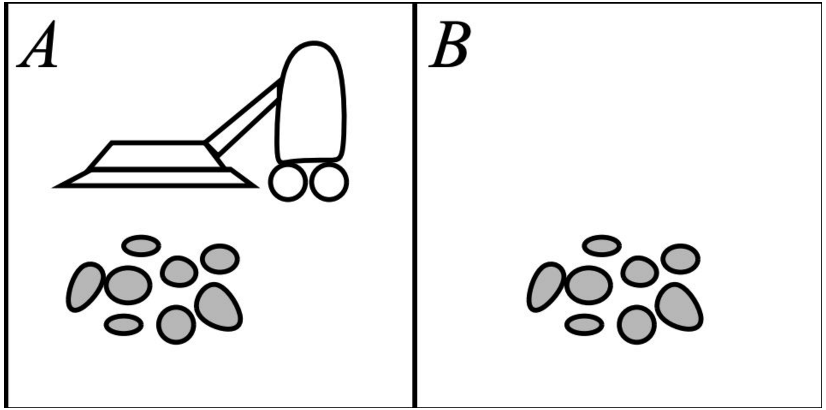


2.1 Agents And Environments

- Agent’s percept: the agent’s perceptual inputs at any given instant.
- Agent’s percept sequence: the complete history of everything the agent has perceived.
- Agent’s behavior (~agent function): maps any percept sequence to an action.
- Agent program: the concrete implementation of the agent action

2.1 Agents And Environments

- Percepts: location and contents, e.g., [A, Dirty]
- Actions: Left, Right, Suck, NoOp



2.1 Agents And Environments

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

function REFLEX-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*

2.2 The Concept Of Rationality

- A rational agent chooses the action that maximizes the expected value of the performance measure given the percept sequence.

2.3 The Nature Of Environments

2.3.1 Specifying the task environment

- To design a rational agent, we must specify the environment task.
- Task environment or PEAS (Performance, Environment, Actuators, Sensors):

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.

2.3 The Nature Of Environments

2.3.2 Properties of task environments

- **Fully observable, partially observable, unobservable**
  - fully observable if the sensors detect all relevant aspects.
  - partially observable because of noisy and inaccurate sensors or parts of the state are missing from the sensor data
  - unobservable if the agent has no sensors at all
- **Single agent vs. multiagent**
  - single-agent: an agent solving a crossword puzzle by itself
  - two- agent:
    - Competitive: chess
    - Cooperative: taxi-driving

2.3 The Nature Of Environments

2.3.2 Properties of task environments

- **Deterministic vs. stochastic**
  - Deterministic: if the next state of the environment is completely determined by the current state and the action executed by the agent
  - Stochastic: otherwise; generally implies that uncertainty about outcomes is quantified in terms of probabilities
- **Episodic vs. sequential**
  - Episodic: the agent's experience is divided into atomic episodes; the next episode does not depend on the actions taken in previous episodes.
  - Sequential: the current decision could affect all future decisions; e.g., chess and taxi driving

2.3 The Nature Of Environments

2.3.2 Properties of task environments

- **Static vs. dynamic:**
    - dynamic: if the environment can change while an agent is deliberating; e.g., taxi driving
    - static: otherwise; e.g., crossword puzzles
    - semi-dynamic: if the environment does not change with the passage of time but the agent's performance score does, e.g., chess, when played with a clock
- **Discrete vs. continuous:** applies to the *state* of the environment, to the way *time* is handled, and to the *percepts* and *actions* of the agent
    - the chess environment has a finite number of distinct *states*, a discrete set of percepts and *actions*
    - taxi driving is a continuous-*state* and continuous-*time* problem, its *actions* are also continuous
- **Known vs. unknown:**
    - In a known environment, the outcomes for all actions are given.

2.4 The Structure Of Agents

- The job of AI is to design *an agent program* that implements *the agent function*
- This program runs on computing device with physical sensors, actuators: *architecture*

AGENT = ARCHITECTURE + PROGRAM

2.4 The Structure Of Agents

2.4.1 Agent programs

- They take the current percept as input and return an action.
- Four basic types in order of increasing generality
    - simple reflex agents
    - reflex agents with state
    - goal-based agents
    - utility-based agents

2.4 The Structure Of Agents

2.4.2 Simple reflex agents:

- the simplest type
- the agents select actions on the basis of the *current* percept

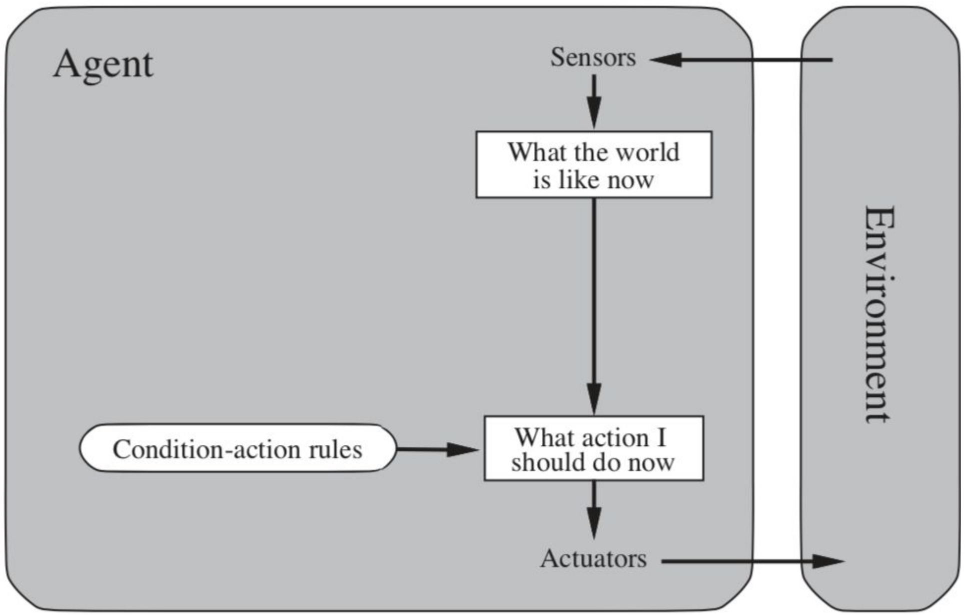


Figure 2.9 Schematic diagram of a simple reflex agent.

2.4 The Structure Of Agents

2.4.2 Simple reflex agents:

function REFLEX-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*

2.4 The Structure Of Agents

2.4.3 Model-based reflex agents

- the agents maintain internal state to track aspects of the world that are not evident in the current percept

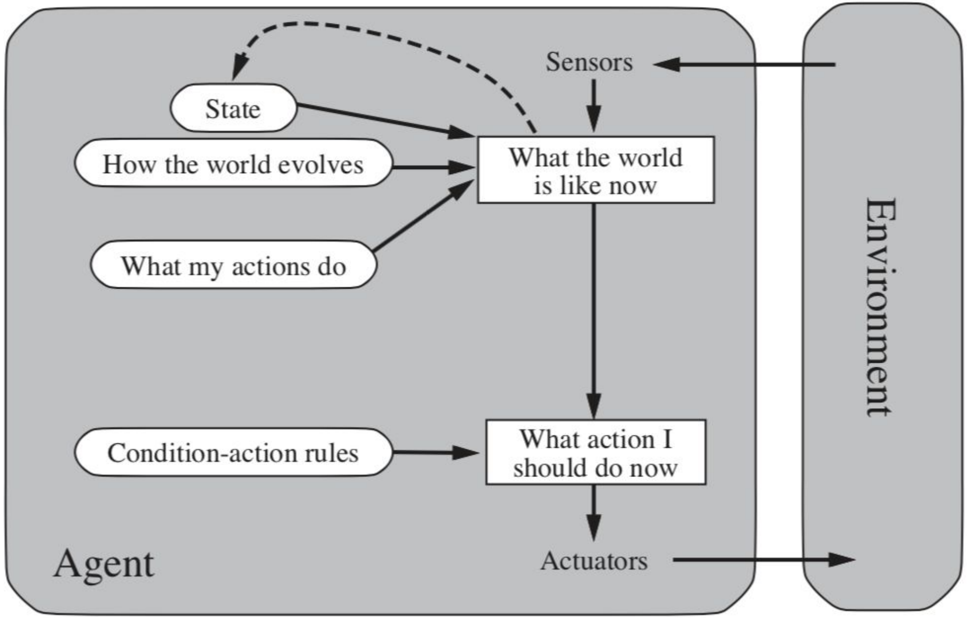


Figure 2.11 A model-based reflex agent.

2.4 The Structure Of Agents

2.4.4 Goal-based agents

- the agent needs some sort of goal information
- the agent program combines the goal information with the model to choose right actions

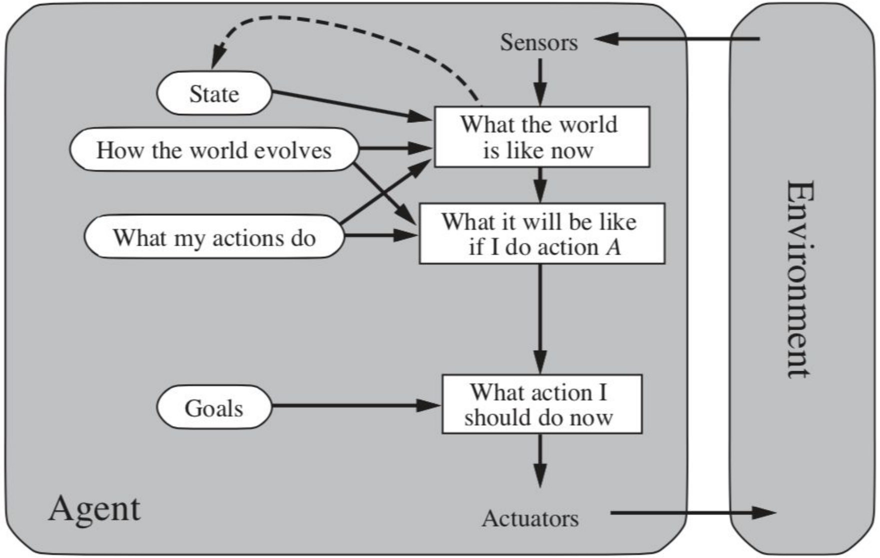


Figure 2.13 A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

2.4 The Structure Of Agents

2.4.5 Utility-based agents

- Goals provide a binary distinction between “happy” and “unhappy” states.
- A performance measure should score exactly *how happy* they would make the agent.
- An agent’s utility function: an internalization of the performance measure.

2.4 The Structure Of Agents

2.4.5 Utility-based agents

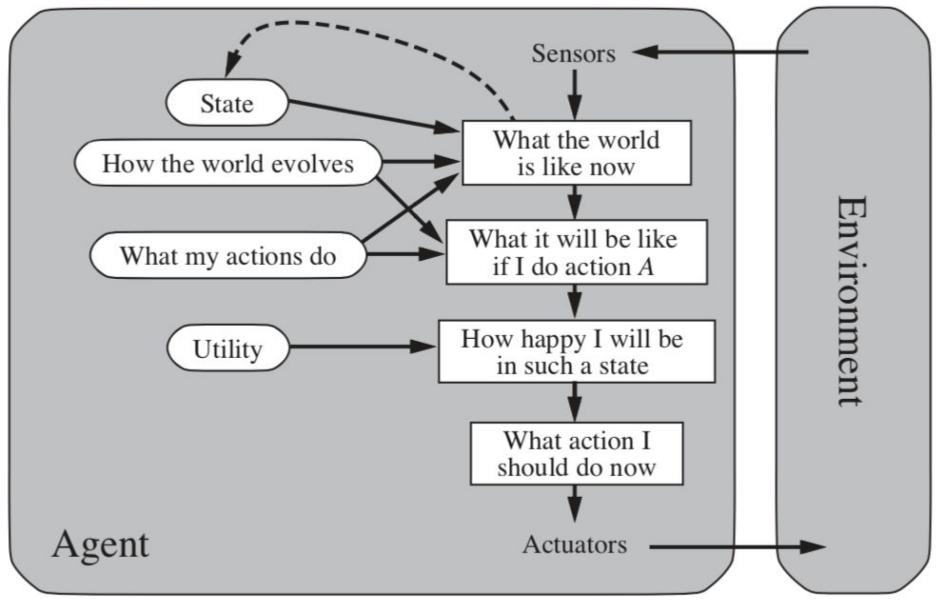


Figure 2.14 A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome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

2.4 The Structure Of Agents

2.4.6 Learning agents

- All agents can improve their performance through learning.

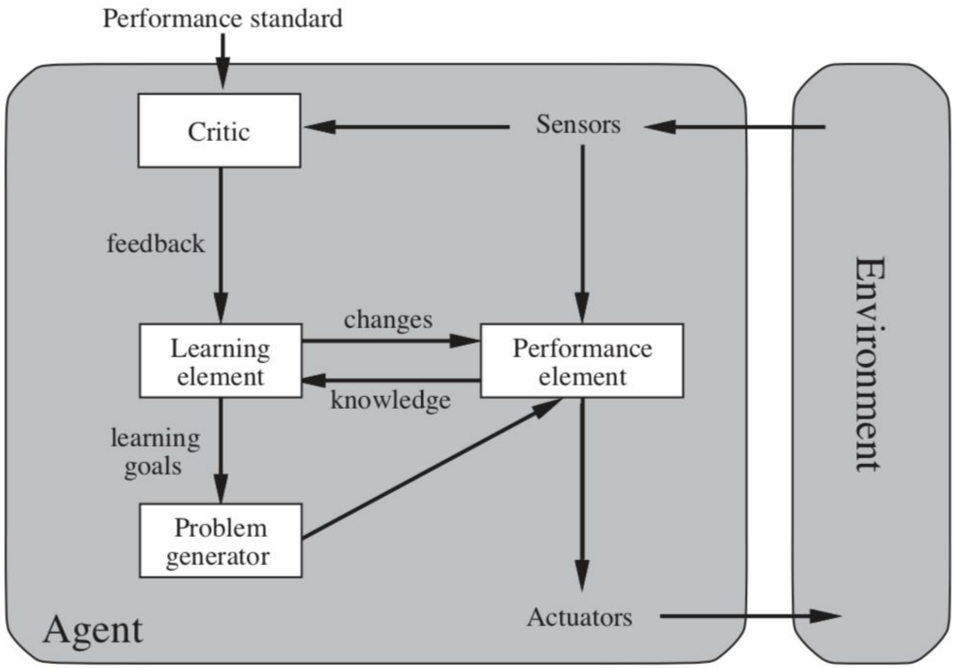


Figure 2.15 A general learning agent.

2.4 The Structure Of Agents

2.4.6 Learning agents

- A learning agent can be divided into four conceptual components:
  - **learning element** uses feedback and determines how the *performance element* should be modified to do better => making improvements
  - **performance element** takes in percepts and decides on actions => selecting external actions
  - **problem generator**: suggesting actions that will lead to new and informative experiences.