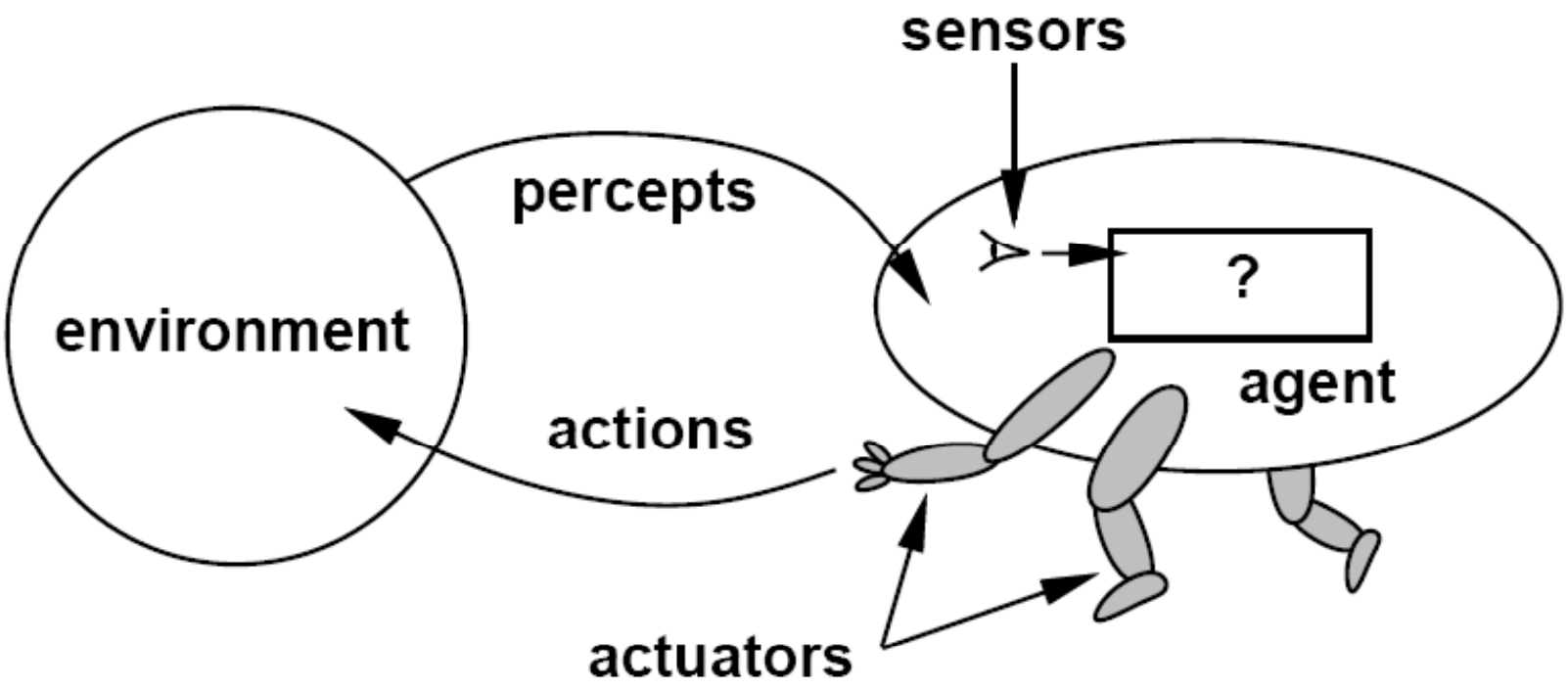
Rational Agents (Chapter 2)

Outline

- Agent function and agent program
- Rationality
- PEAS (Performance measure, Environment, Actuators, Sensors)
- Environment types
- Agent types

Agents

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators



Agent function

- The agent function maps from *percept histories* to *actions*

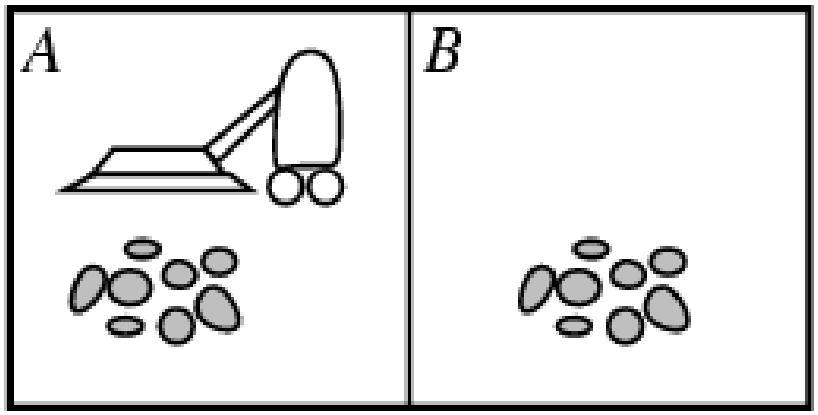
$$P^* \longrightarrow A$$

- The agent program runs on the physical architecture to produce the agent function

- agent = architecture + program

Vacuum-cleaner world

- **Percepts:**
  Location and status,
  e.g., [A,Dirty]
- **Actions:**
  Left, Right, Suck, NoOp



Example vacuum agent program:

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

Rational agents

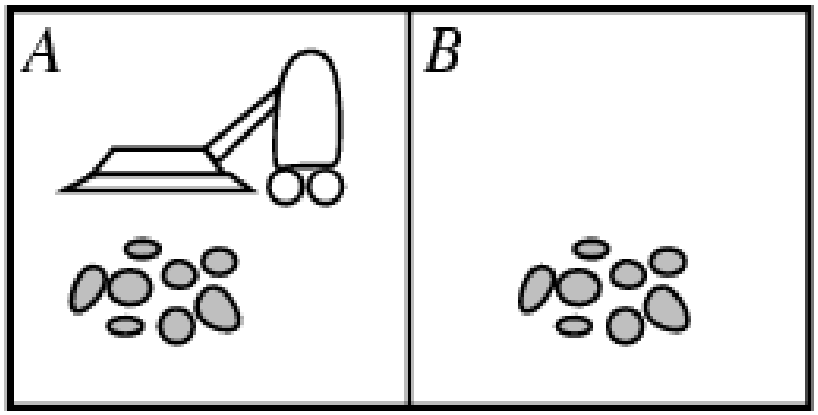
- For each possible percept sequence, a **rational agent** should select an action that is expected to maximize its **performance measure**, *given the evidence provided by the percept sequence and the agent's built-in knowledge*

- **Performance measure (utility function):** An *objective* criterion for success of an agent's behavior

- Can a rational agent make mistakes?

Back to vacuum-cleaner world

- **Percepts:**
  Location and status,
  e.g., [A,Dirty]
- **Actions:**
  Left, Right, Suck, NoOp



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

- Is this agent rational?
  - Depends on performance measure, environment properties

Specifying the task environment

- Problem specification: **Performance measure, Environment, Actuators, Sensors (PEAS)**

- **Example: automated taxi driver**
  - **Performance measure**
    - Safe, fast, legal, comfortable trip, maximize profits
  - **Environment**
    - Roads, other traffic, pedestrians, customers
  - **Actuators**
    - Steering wheel, accelerator, brake, signal, horn
  - **Sensors**
    - Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard

Agent: Spam filter

- Performance measure
  - Minimizing false positives, false negatives
- Environment
  - A user’s email account, email server
- Actuators
  - Mark as spam, delete, etc.
- Sensors
  - Incoming messages, other information about user’s account

Environment types

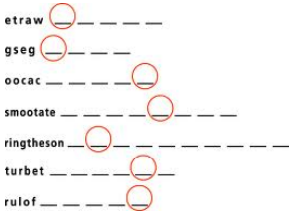
- **Fully observable (vs. partially observable):** The agent's sensors give it access to the complete state of the environment at each point in time
- **Deterministic (vs. stochastic):** The next state of the environment is completely determined by the current state and the agent's action
  - **Strategic:** the environment is deterministic except for the actions of other agents
- **Episodic (vs. sequential):** The agent's experience is divided into atomic "episodes," and the choice of action in each episode depends only on the episode itself

Environment types

- **Static (vs. dynamic):** The environment is unchanged while an agent is deliberating
  - **Semidynamic:** the environment does not change with the passage of time, but the agent's performance score does
- **Discrete (vs. continuous):** The environment provides a fixed number of distinct percepts, actions, and environment states
  - Time can also evolve in a discrete or continuous fashion
- **Single agent (vs. multi-agent):** An agent operating by itself in an environment

- **Known (vs. unknown):** The agent knows the rules of the environment

Examples of different environments



Word jumble solver

Chess with a clock

Scrabble

Taxi driving

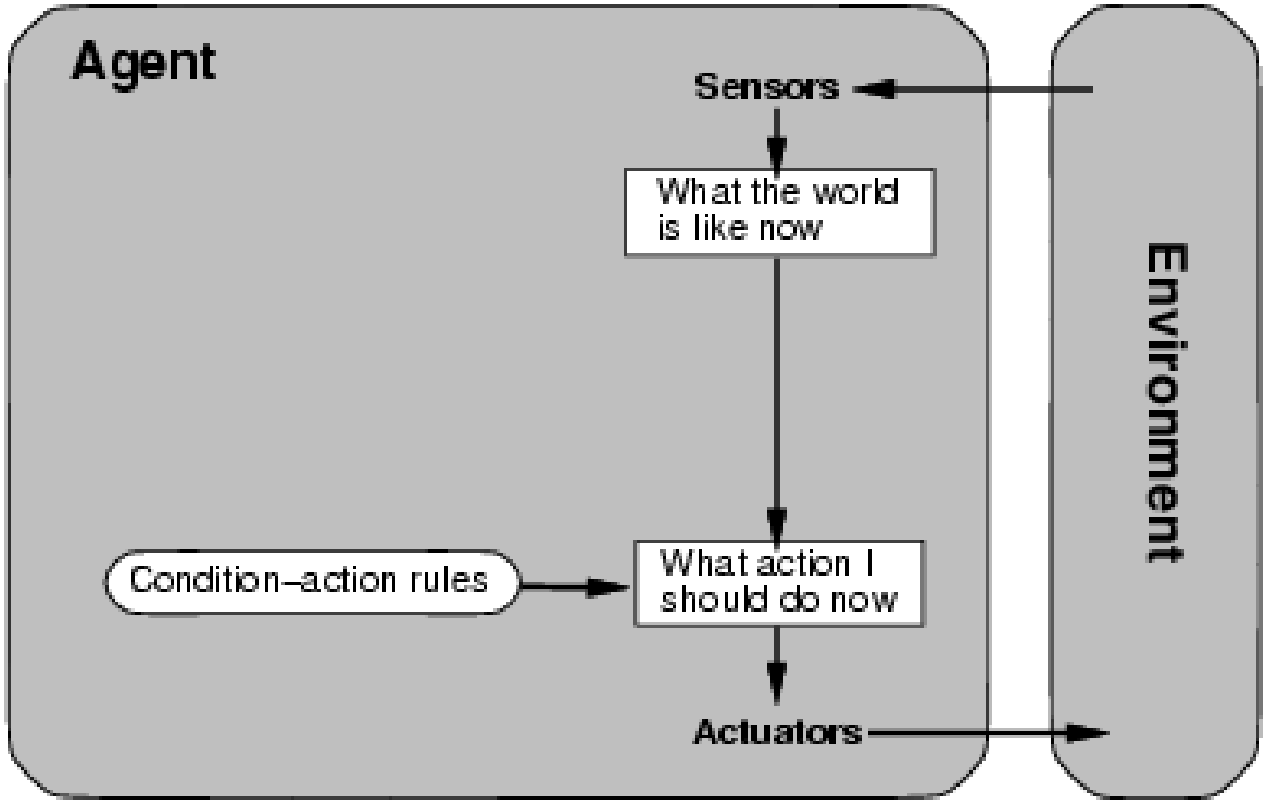
	Word jumble solver	Chess with a clock	Scrabble	Taxi driving
Observable	Fully	Fully	Partially	Partially
Deterministic	Deterministic	Strategic	Stochastic	Stochastic
Episodic	Episodic	Sequential	Sequential	Sequential
Static	Static	Semidynamic	Static	Dynamic
Discrete	Discrete	Discrete	Discrete	Continuous
Single agent	Single	Multi	Multi	Multi

Hierarchy of agent types

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

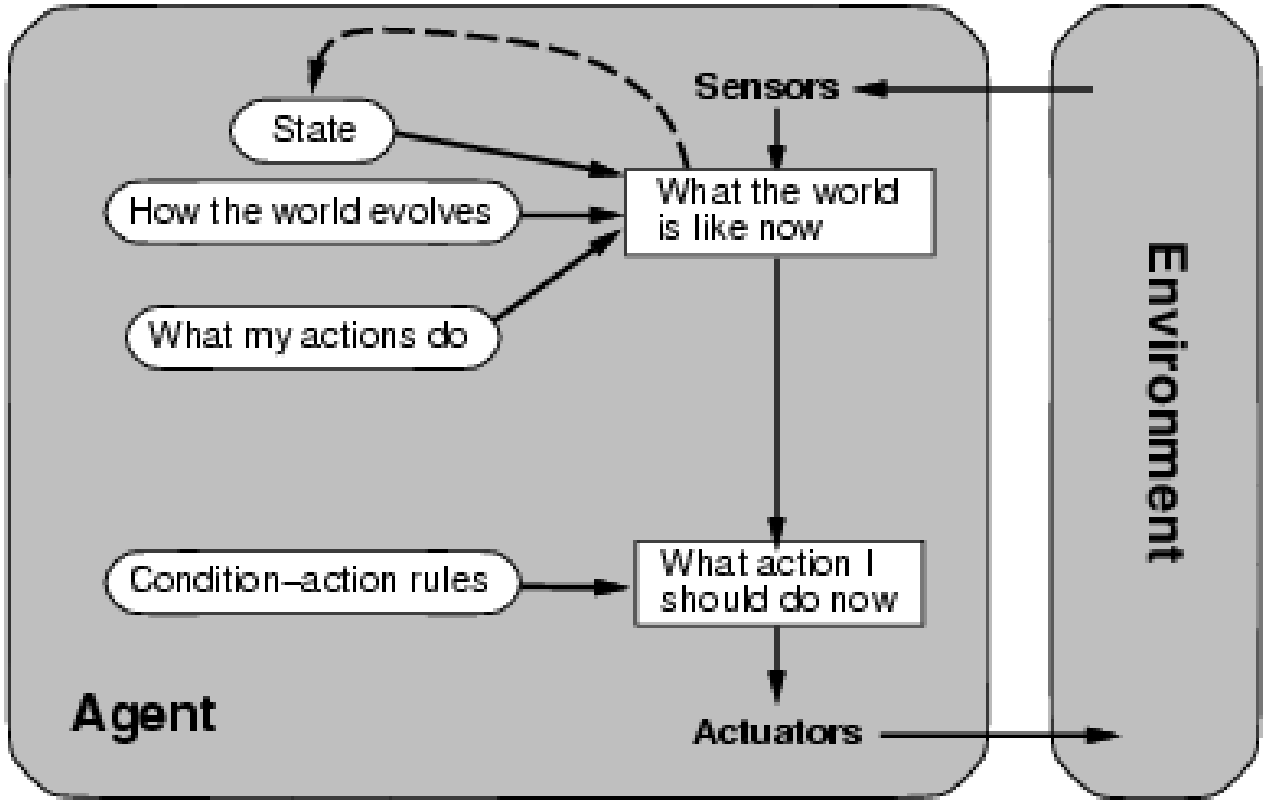
Simple reflex agent

- Select action on the basis of current percept, ignoring all past percepts



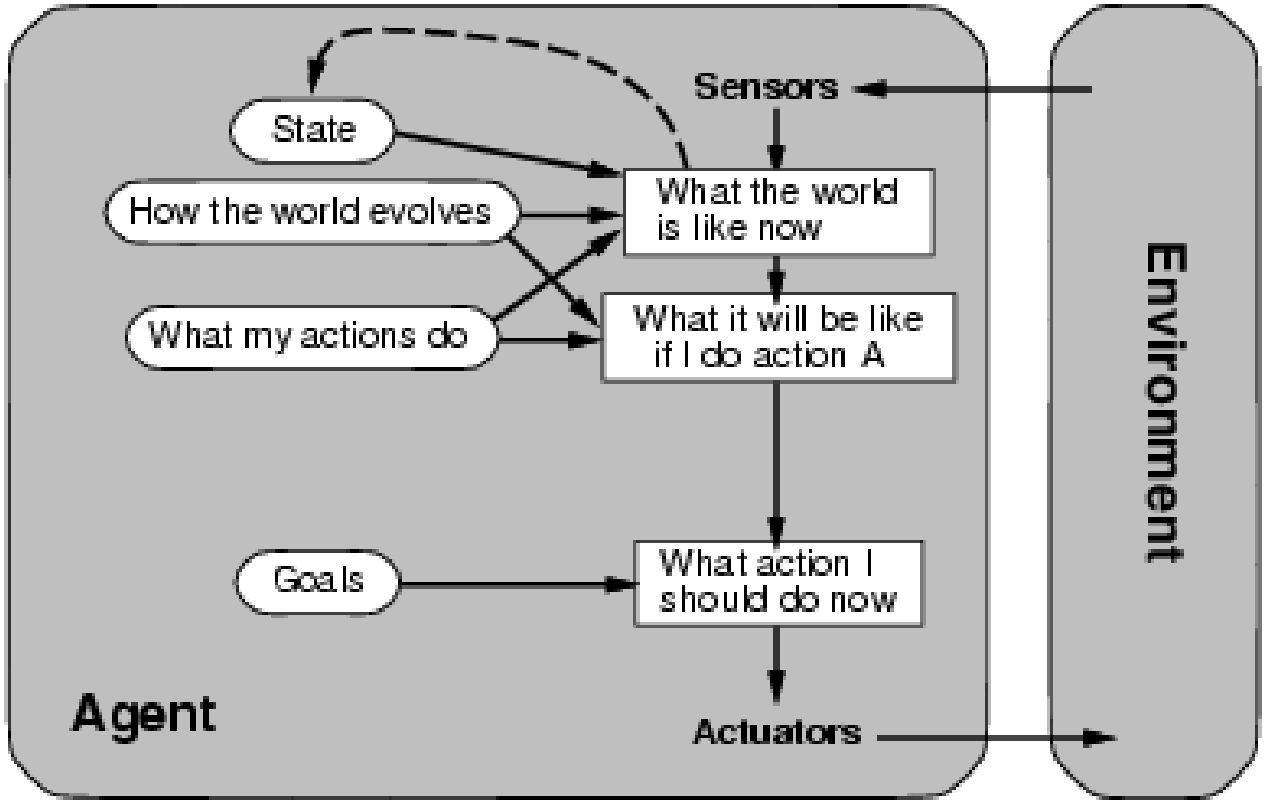
Model-based reflex agent

- Maintains internal state that keeps track of aspects of the environment that cannot be currently observed



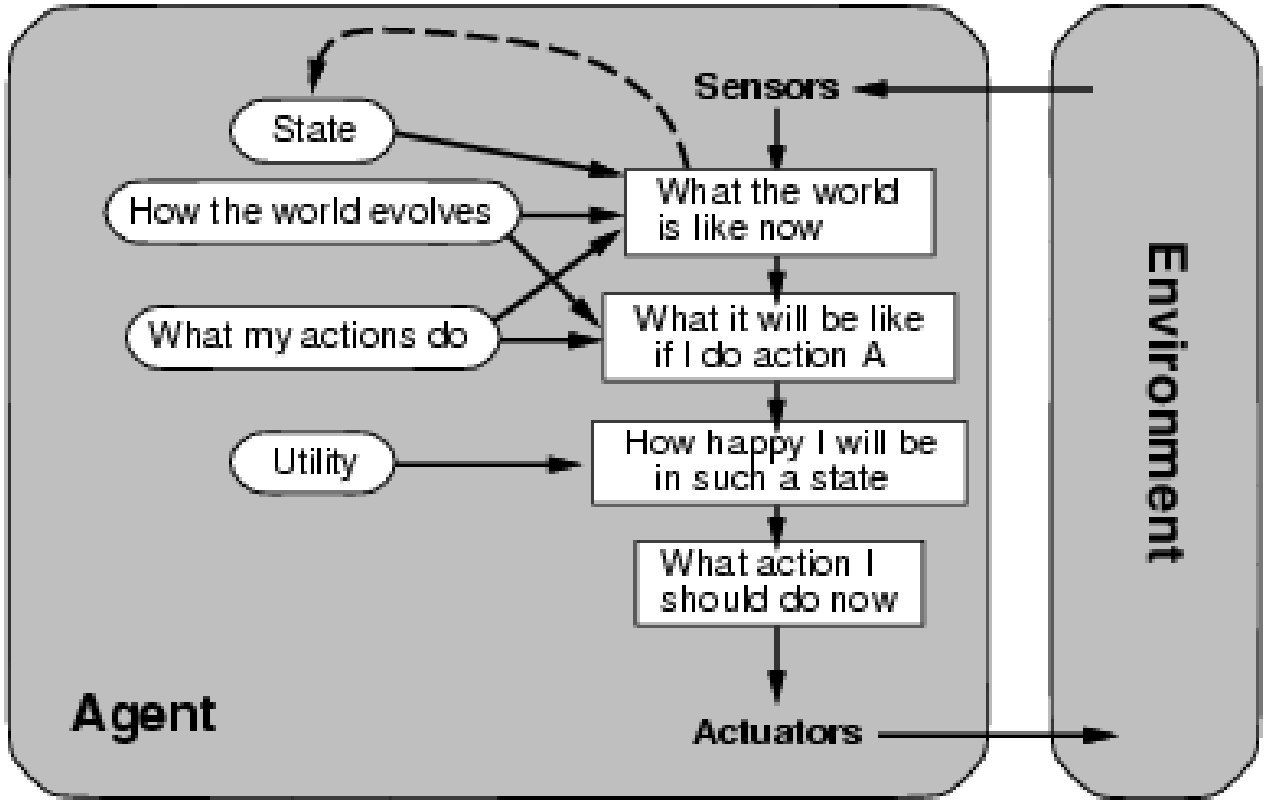
Goal-based agent

- The agent uses goal information to select between possible actions in the current state



Utility-based agent

- The agent uses a utility function to evaluate the desirability of states that could result from each possible action



Where does learning come in?