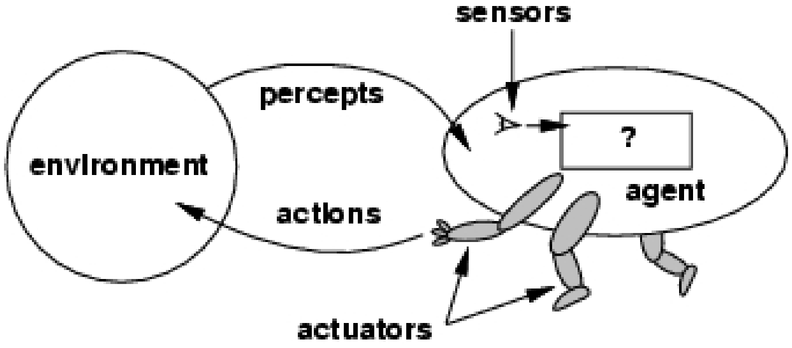
Review

- **Conceptions**
  - **AI**
  - Agent/Intelligent Agent/MAS
  - Rational Agent vs Complete Agent



- **Questions:**
  1. *What is “Turing Test”?*
  2. ***What is “Chinese Room”?***
  3. *Is AI science or engineering?*
  4. *.......*

AI's Hall of Fame

-Edward Albert Feigenbaum/

Review

2018 Turing Award Owners

(Yoshua Bengio、Yann LeCun、Geoffrey Hinton)

Lecture 3: Intelligent Agents

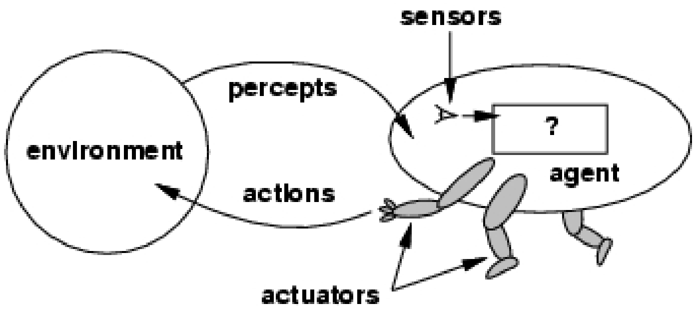
LI Xiaoan (Dustin) 李孝安, Dr. & Associate Prof.
MARS-lab
School of Computer Science and Engineering
Northwestern Polytechnical University

E-mail: dustinli@nwpu.edu.cn
WeChat group: AI-2021-Li
Cell phone: 18629662731

Outline

- Agents and environments
- 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

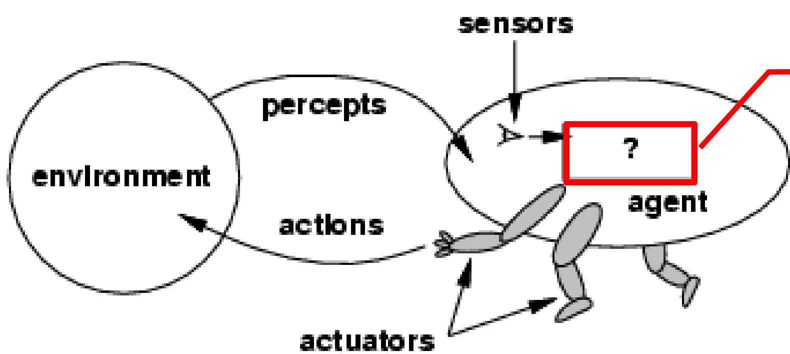
QnA: *examples of Agents: natural or artificial?*

- Human agent:
  - Sensors: eyes, ears, and other organs
  - Actuators: hands, legs, mouth, and other body parts
- Robotic agent:
  - Sensors: cameras and infrared range finders
  - Actuators: various motors

Agents

- Perceiving
  - Awareness (to become aware of through the senses)
  - Consciousness (to become conscious of）
- Cognizing
  - Extent of perception, knowledge, experience, or ability
- Acting
  - perform or work out
- Human agent vs Robotic agent
  - *Differences?*
  - *Cognition/Perception?*
  - Challenging problems for AI*: directions for developing?

Agents and environments



Function
vs
Program

- The agent function maps from percept histories to actio

Agent function vs agent program ?

$$[f: \mathcal{P}^* \rightarrow \mathcal{A}]$$

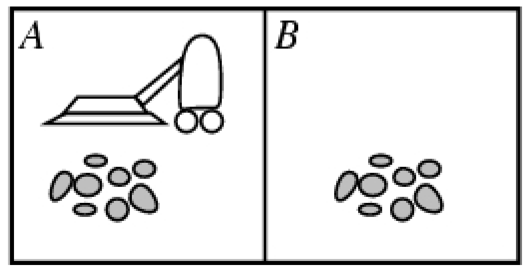
- The agent program runs on the physical ***architecture*** to produce *f*

Physical or
Virtual*

- The agent structure

agent = architecture + program

Example: Vacuum-cleaner world



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

A vacuum-cleaner agent

- \input{tables/vacuum-agent-function-table}

Rational agents

- An agent should strive to "do the right thing", based on what it can perceive and the actions it can perform. The right action is the one that will cause the agent to be most successful

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

- E.g., performance measure of a vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.

Rational agents

- 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.

Rational agents

- Rationality is distinct from ***omniscience*** (all-knowing with infinite knowledge) Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, ***exploration***) An agent is autonomous if its behavior is determined by its own experience (with ability to **learn** and **adapt**) --*Autonomy*

PEAS

- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent design
  Consider, e.g., PEAS for the task of designing an ***automated taxi driver***:
  - Performance measure
  - Environment
  - Actuators
  - Sensors

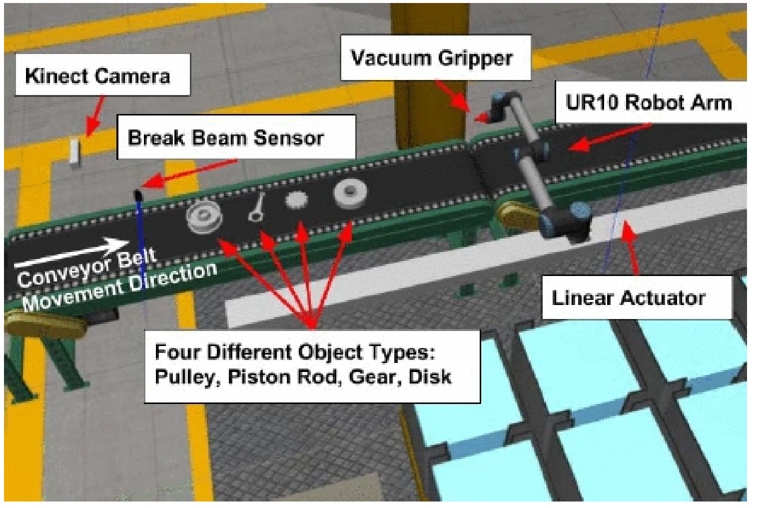
PEAS

- Consider, PEAS for the task of designing an **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

PEAS

- Agent: Medical diagnosis system
  - Performance measure: Healthy patient, minimize costs, lawsuits
  - Environment: Patient, hospital, staff
  - Actuators: Screen display (questions, tests, diagnoses, treatments, referrals)
  - Sensors: Keyboard (entry of symptoms, findings, patient's answers)

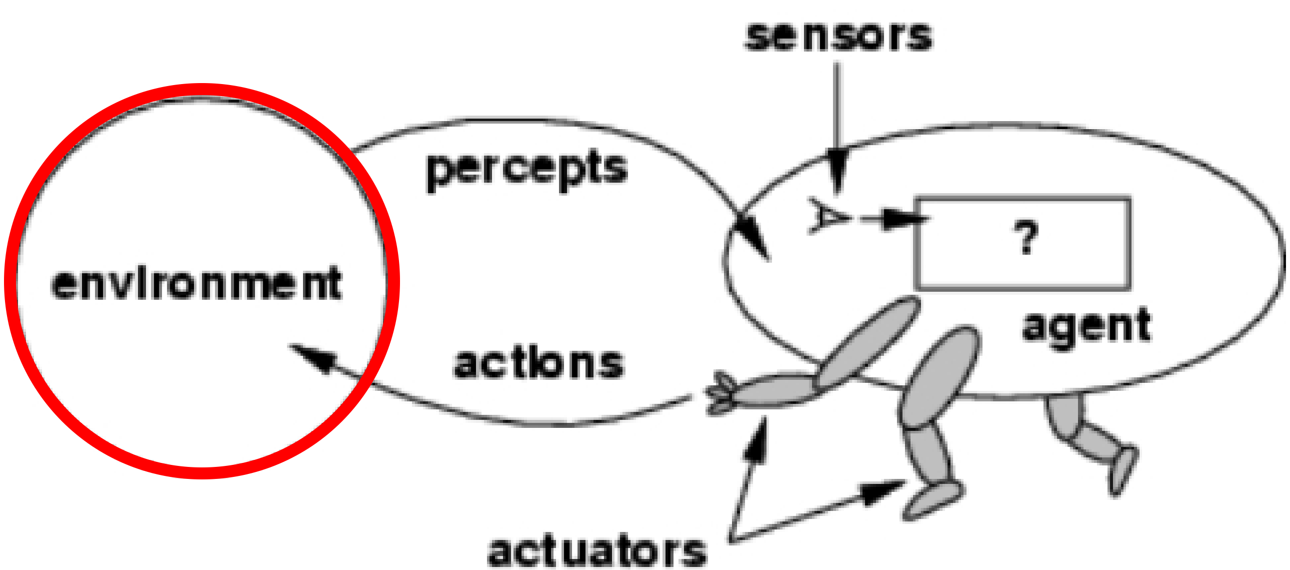
PEAS



- Agent: Part-picking robot
  - Performance measure: Percentage of parts in correct bins
  - Environment: Conveyor belt with parts, bins
  - Actuators: Jointed arm and hand
  - Sensors: Camera, joint angle sensors

PEAS

- Agent: Interactive English tutor
  - Performance measure: Maximize student's score on test
  - Environment: Set of students
  - Actuators: Screen display (exercises, suggestions, corrections)
  - Sensors: Keyboard


sensors
percepts
?
environment
agent
actions
actuators

Environment types

- Fully observable (vs. partially observable)

An 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 action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic)

- Episodic (vs. sequential)

The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), 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. (The environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does)

- Discrete (vs. continuous)

A limited number of distinct, clearly defined percepts and actions.

- Single agent (vs. multiagent)

An agent operating by itself in an environment.

- **QnA:** *Real World vs Virtual World?*

Environment types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

- The environment type largely determines **the agent design**

- **QnA:** *Real World vs Virtual World?*
  - The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent
  - Question: The virtual world?

Agent functions and programs*

- An agent is completely specified by the agent function mapping percept sequences to actions
- One agent function (or a small equivalence class) is <u>rational</u>

- Aim: find a way to implement the rational agent function concisely

Table-lookup agent

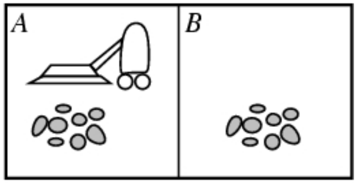
function TABLE-DRIVEN-AGENT(*percept*) **returns** an action
persistent: *percepts*, a sequence, initially empty
table, a table of actions, indexed by percept sequences, initially fully specified

append *percept* to the end of *percepts*
action ← LOOKUP(*percepts*, *table*)
return *action*

Figure 2.7 The TABLE-DRIVEN-AGENT program is invoked for each new percept and returns an action each time. It retains the complete percept sequence in memory.

- Drawbacks:
  - Huge table
  - Take a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries

Agent program for a vacuum-cleaner agent



function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action

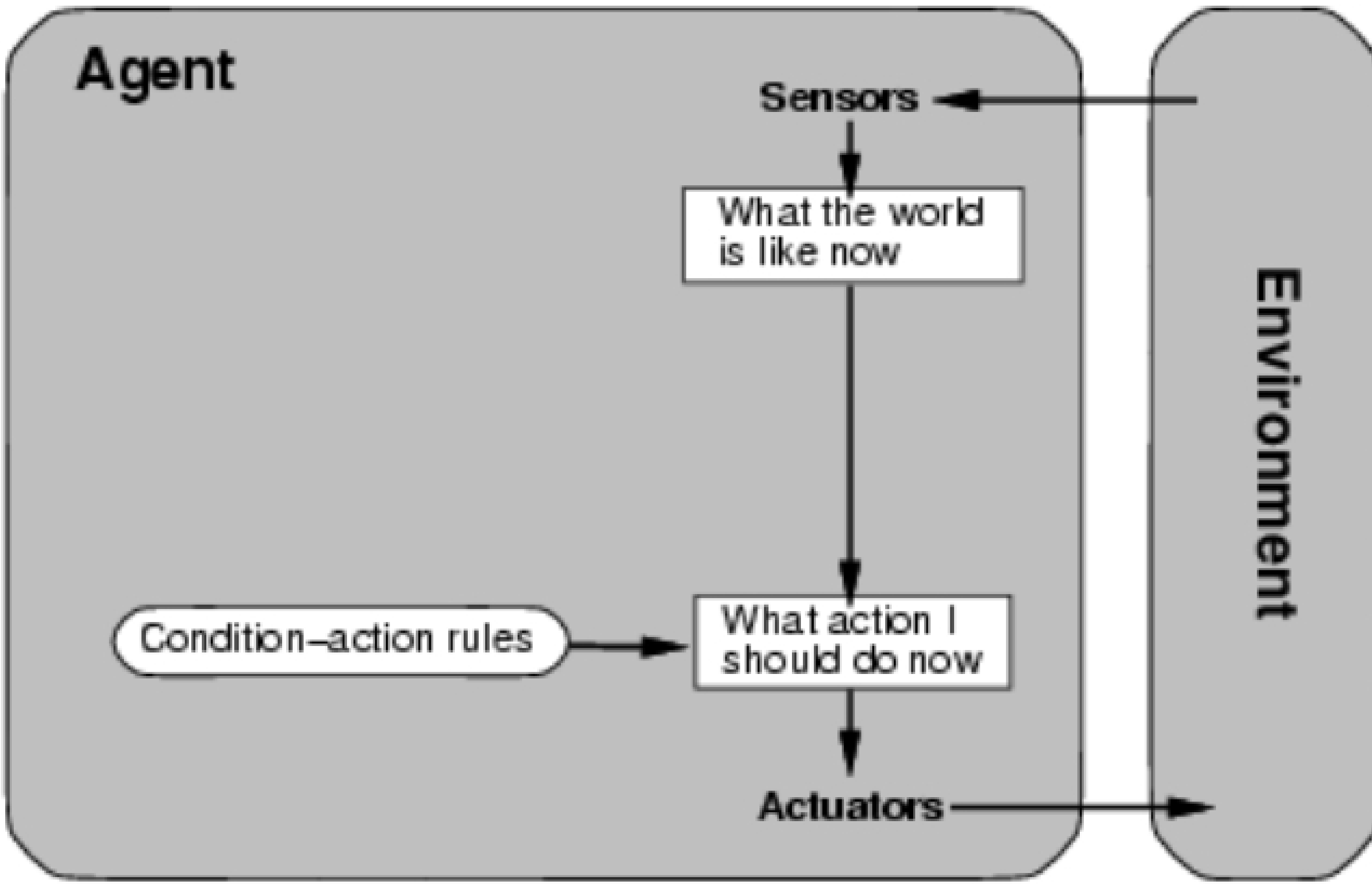
    if *status* = *Dirty* **then return** *Suck*
    else if *location* = *A* **then return** *Right*
    else if *location* = *B* **then return** *Left*

Figure 2.8 The agent program for a simple reflex agent in the two-state vacuum environment. This program implements the agent function tabulated in Figure 2.3.

Agent types

- Four basic types in order of increasing generality:
  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents

Simple reflex agents



Simple reflex agents

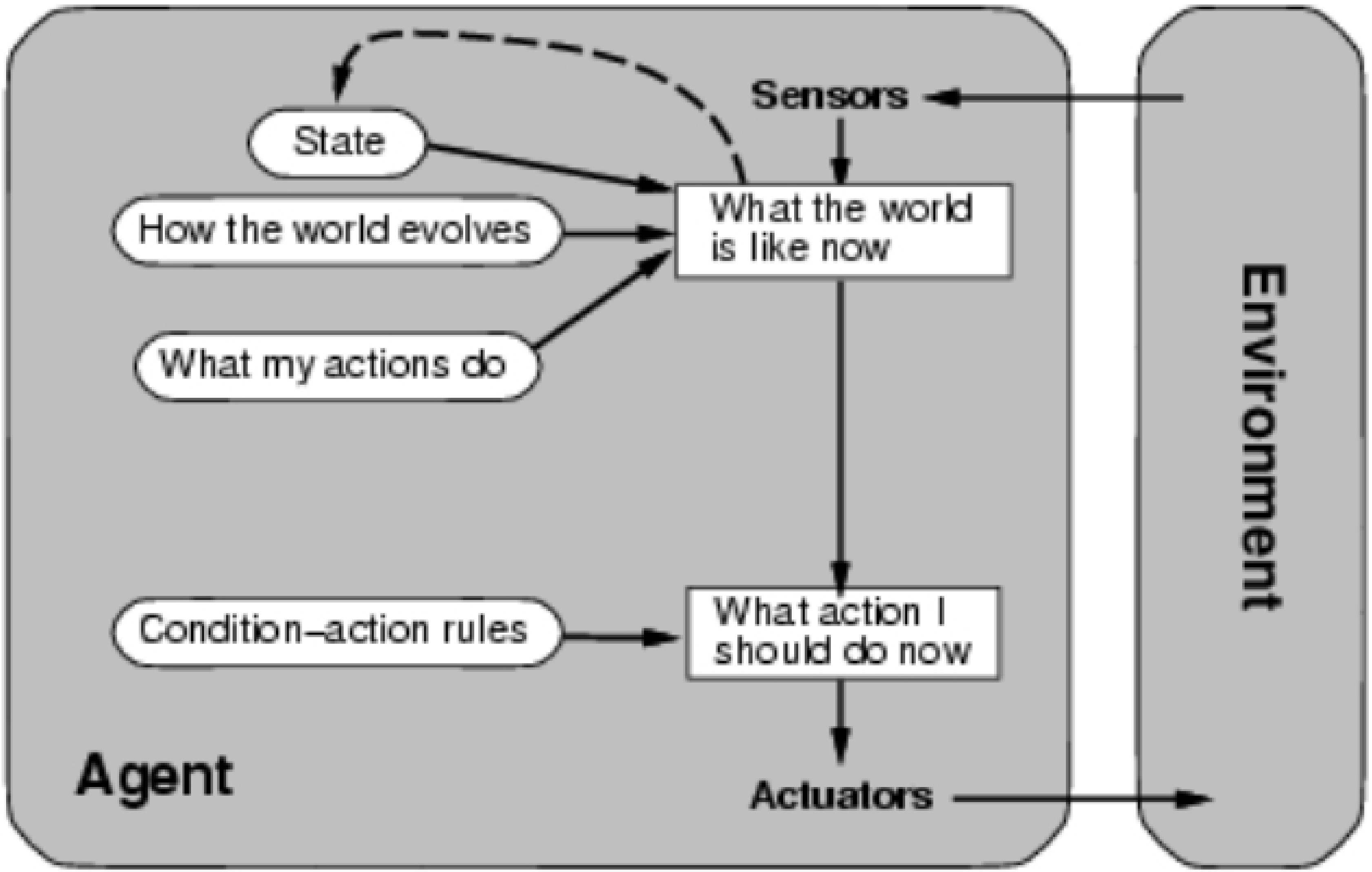
- Agent program

function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
persistent: *rules*, a set of condition–action rules

state ← INTERPRET-INPUT(*percept*)
rule ← RULE-MATCH(*state*, *rules*)
action ← *rule*.ACTION
return *action*

Figure 2.10 A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

Model-based reflex agents



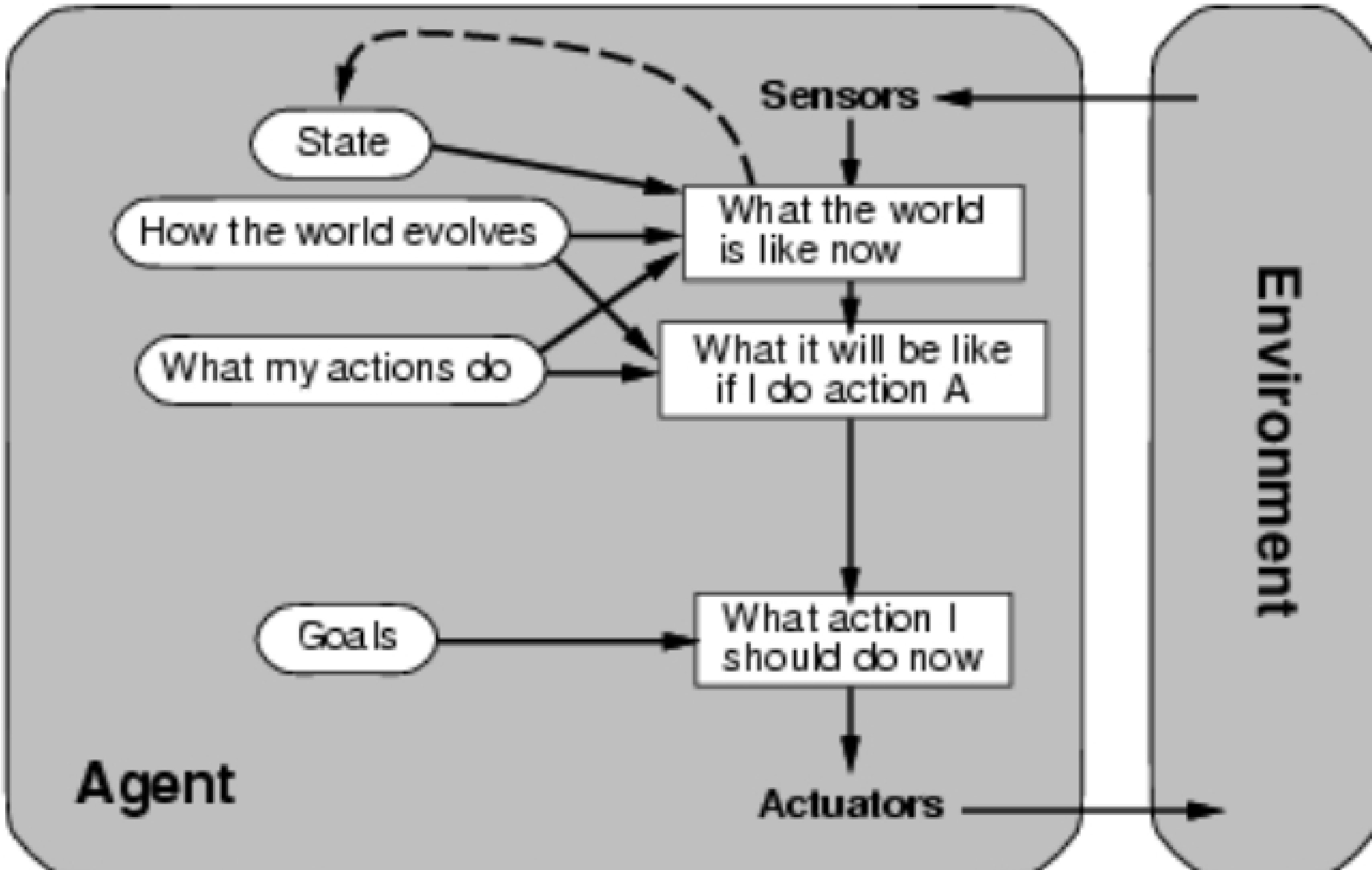
Model-based reflex agents

- Agent program

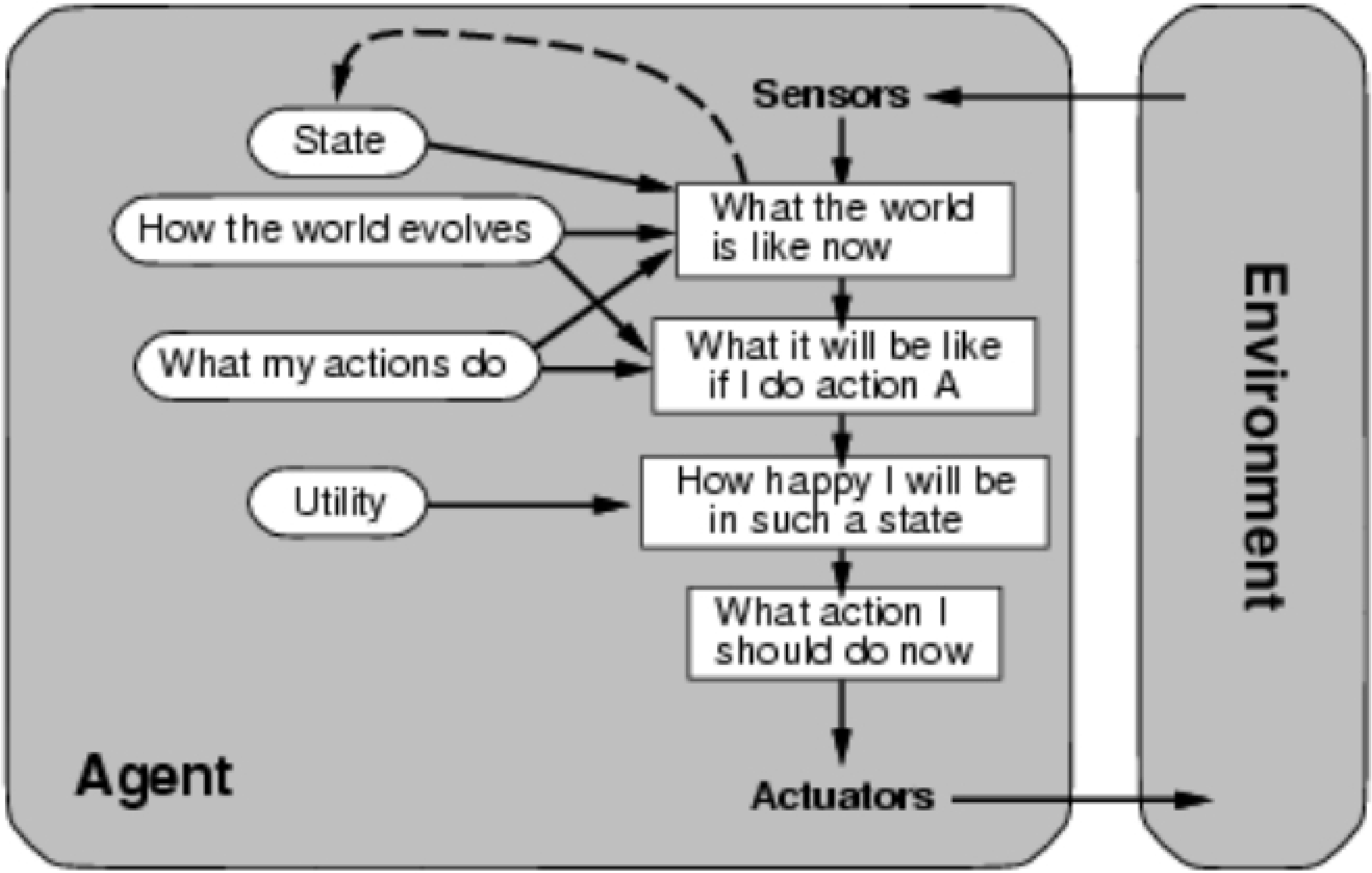
```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
              model, a description of how the next state depends on current state and action
              rules, a set of condition–action rules
              action, the most recent action, initially none

  state ← UPDATE-STATE(state, action, percept, model)
  rule ← RULE-MATCH(state, rules)
  action ← rule.ACTION
  return action
```

Figure 2.12 A model-based reflex 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 reflex agent.

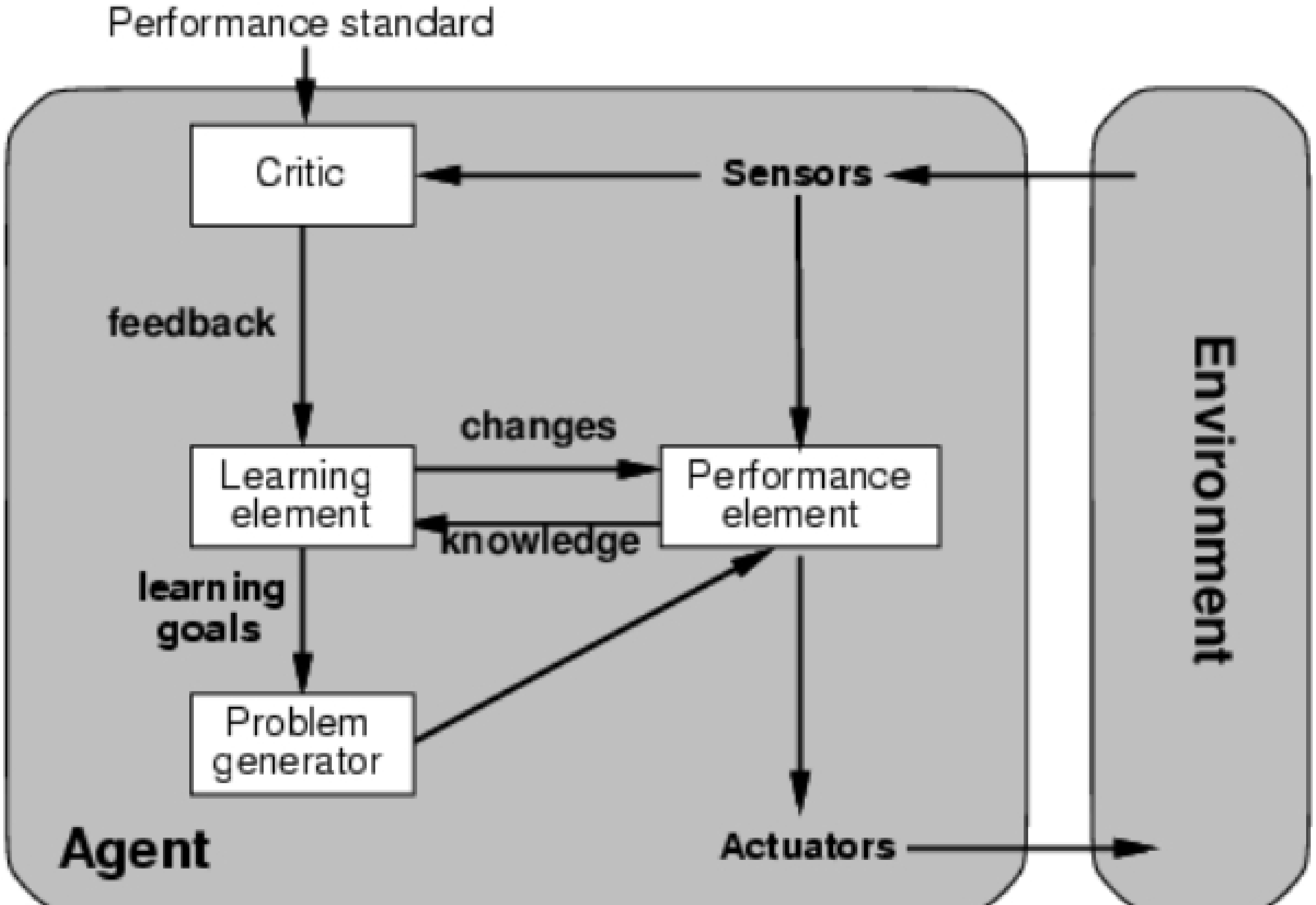
Goal-based agents



Utility-based agents



Learning agents



Summary

- **Agents** interact with environments through actuators and sensors
- The **agent function** describes what the agent does in all circumstances
- The **performance measure** evaluates the environment sequence
- A **perfectly rational agent** maximizes expected performance
- **Agent programs** implement (some) agent functions
- **PEAS** descriptions dene task environments
- **Environments** are categorized along several dimensions:

  observable? deterministic? episodic? static? discrete? single-agent?
- Several basic agent architectures exist:

  reex, reex with state, goal-based, utility-based

Assignment

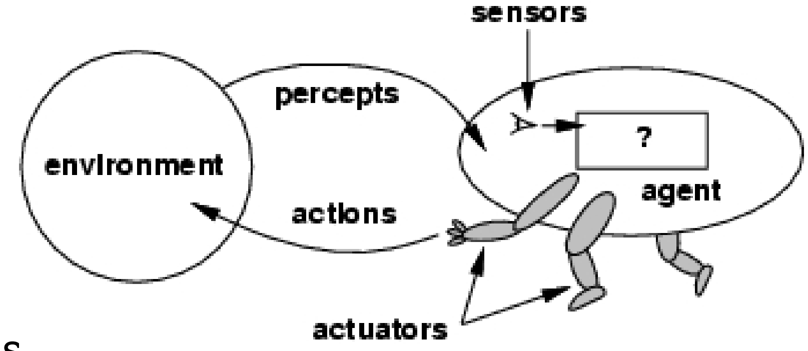
– Chap 2: exercise 2.3, 2.5
– Exercise 2.6

*Handed in next Tuesday

Review

- **Agents**
- **agent function** **vs** **agent program**
- A **perfectly rational agent**
- **PEAS**
- **Environments**
  - are categorized along several dimensions.

  observable? deterministic? episodic? static? discrete? single-agent?
  - **Real world vs Virtual world**



- Several basic agent architectures exist:

  reex, reex with state, goal-based, utility-based
- **Questions:**

  1. *Why design of a rational intelligent agent or agents is usually difficult? Where the difficulties come from?*

  2. *Is memory necessary for intelligent agent/agents?*