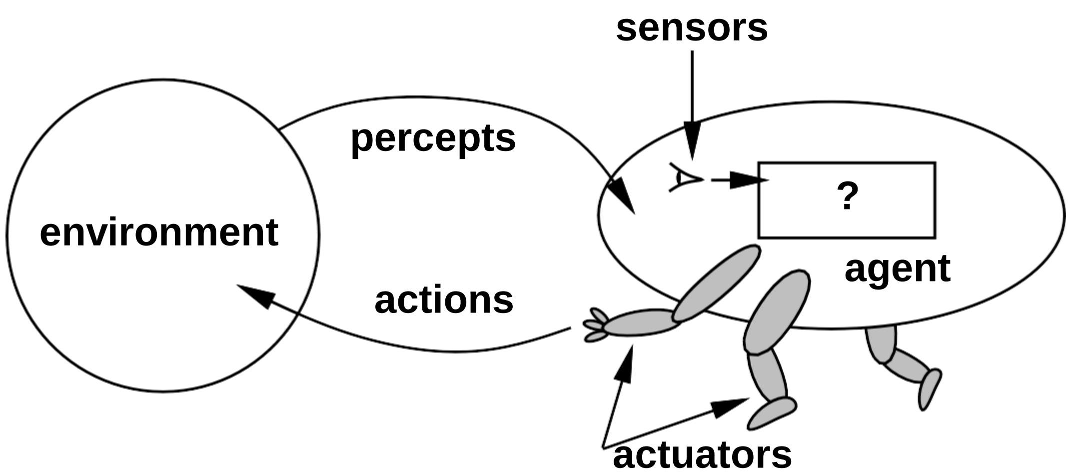
Intelligent Agents

Chapter 2

Outline

- Agents and environments
- Rationality
- PEAS (Performance measure, Environment, Actuators, Sensors)
- Environment types
- The structure of agents

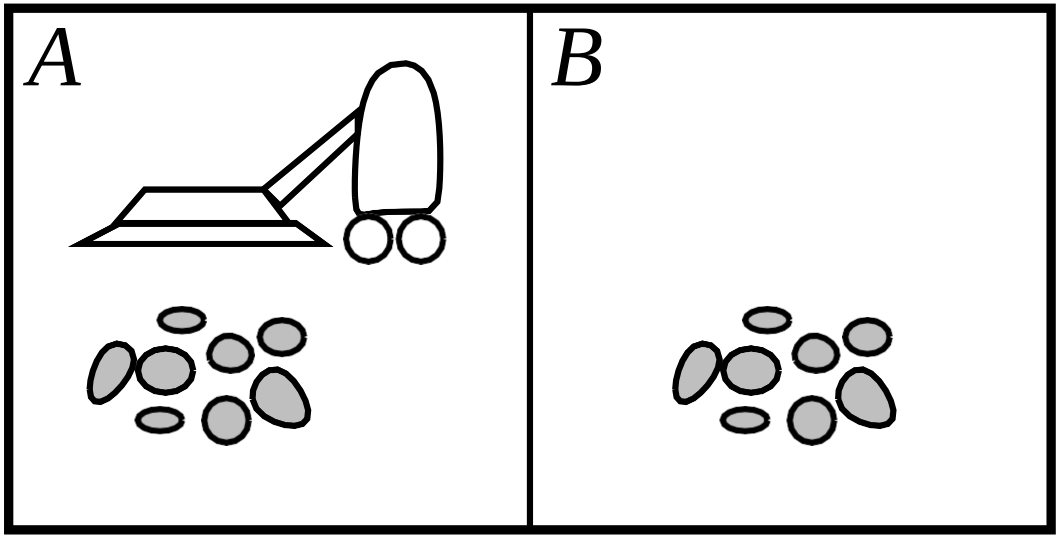
Agents and environments



Agents: An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators include humans, robots, softbots, thermostats, etc.

An *agent* is a computer system that is *situated* in some *environment*, and that is capable of *autonomous action* in this environment in order to meet its design objectives". Example: Taxi Driver is an agent and road is an environment.

Vacuum-cleaner world



Percepts: location and contents, e.g., [*A*, *Dirty*]

Actions: *Left*, *Right*, *Suck*, *NoOp*

A vacuum-cleaner agent

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

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*

GOOD BEHAVIOR: THE CONCEPT OF RATIONALITY

- A rational agent is one that does the right thing—conceptually speaking, every entry in the table for the agent function is filled out correctly.

- Obviously, doing the right thing is better than doing the wrong thing, but what does it mean to do the right thing.

- We answer this age-old question in an age-old way: by considering the consequences of the agent's behavior.

- When an agent is plunked down in an environment, it generates a sequence of actions according to the percepts it receives.

- This sequence of actions causes the environment to go through a sequence of states

An AI system is composed of an **agent and its environment**. The agents act in their environment. The environment may contain other agents.

An agent is anything that can be viewed as :

- perceiving its environment through **sensors** and
- acting upon that environment through **actuators**

What is Ideal Rational Agent?

- An ideal rational agent is the one, which is capable of doing expected actions to maximize its performance measure, on the basis of –
- Its percept sequence
- Its built-in knowledge base

Rationality of an agent depends on the following –

- The **performance measures**, which determine the degree of success.
- Agent's **Percept Sequence** till now.
- The agent's **prior knowledge about the environment**.
- The **actions** that the agent can carry out.

Rationality

Fixed performance measure evaluates any sequence of environment states

–the amount of dirt cleaned in time T
–one point per each square cleaned up at each time step T?
–one point per clean square per time step, minus one per move?
–penalize for $> k$ dirty squares, electricity, noise?
–having a clean floor

Design performance measures according to what one actually wants in the environment, rather than to how one thinks the agent should behave.

A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date

Rationality

What is rational at any given time depends on four things:

–The performance measure that defines the criterion of success.
–The agents prior knowledge of the environment.
–The actions that the agent can perform.
–The agents percept sequence to date.

♦The performance measure awards one point for each clean square at each time step, over a lifetime of 1000 time steps.

♦The geography of the environment is known a priori but the dirt distri- bution and the initial location of the agent are not. Clean squares stay clean and sucking cleans the current square. The Left and Right actions move the agent left and right except when this would take the agent outside the environment, in which case the agent remains where it is.

♦The only available actions are Left, Right, and Suck .

♦The agent correctly perceives its location and whether it contains dirt.

Omniscience, learning, autonomy

An omniscient agent knows the actual outcome of its actions and can act accordingly; but omniscience is impossible in reality.

learn as much as possible from what it perceives

Autonomy: A rational agent should be learn what it can to compensate for partial or incorrect prior knowledge.

For example, a vacuum-cleaning agent that learns to foresee where and when additional dirt will appear will do better than one that does not

The Nature of Environments

An environment in artificial intelligence is the surrounding of the agent. The agent takes input from the environment through sensors and delivers the output to the environment through actuators.

- Some programs operate in the entirely **artificial environment** confined to keyboard input, database, computer file systems and character output on a screen.
- In contrast, some software agents (software robots or softbots) exist in rich, unlimited softbots domains. The simulator has a **very detailed, complex environment**.

- The software agent needs to choose from a long array of actions in real time. A softbot designed to scan the online preferences of the customer and show interesting items to the customer works in the **real** as well as an **artificial** environment.

- The most famous **artificial environment** is the **Turing Test environment**, in which one real and other artificial agents are tested on equal ground. This is a very challenging environment as it is highly difficult for a software agent to perform as well as a human.

THE NATURE OF ENVIRONMENTS

1. **Specifying the task environment**: Specify the performance measure. the environment, and the agent's actuators and sensors. We group all these under the heading of the task environment

- (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 PEAS description of the task environment for an automated taxi.

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts: bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, beaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises. suggestions, corrections	Keyboard entry

Figure Examples of agent types and their PEAS descriptions.

Properties of Environment

- The environment has multifold properties –
- **Discrete / Continuous** – If there are a limited number of distinct, clearly defined, states of the environment, the environment is discrete (For example, chess); otherwise it is continuous (For example, driving).
- **Observable / Partially Observable** – If it is possible to determine the complete state of the environment at each time point from the percepts it is observable; otherwise it is only partially observable.
- **Static / Dynamic** – If the environment does not change while an agent is acting, then it is static; otherwise it is dynamic.

- **Single agent / Multiple agents** − The environment may contain other agents which may be of the same or different kind as that of the agent.
- **Accessible / Inaccessible** − If the agent's sensory apparatus can have access to the complete state of the environment, then the environment is accessible to that agent.
- **Deterministic / Non-deterministic** – If the next state of the environment is completely determined by the current state and the actions of the agent, then the environment is deterministic; otherwise it is non-deterministic.

- **Episodic / Non-episodic** – In an episodic environment, each episode consists of the agent perceiving and then acting. The quality of its action depends just on the episode itself. Subsequent episodes do not depend on the actions in the previous episodes. Episodic environments are much simpler because the agent does not need to think ahead

Examples of task environments and their characteristics

Task Env.	Observable	Agents	Deterministic	Episodic	Static	Discrete

Task Env.	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete

Examples of task environments and their characteristics

Task Env.	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete

Task Env.	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete

Task Env.	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete

Task Env.	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continous

Task Env.	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continous
Medical diagnosis	Partially	Multi	Stochastic	Sequential	Dynamic	Continous

Task Env.	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle Chess with clock	Fully Fully	Single Multi	Deterministic Deterministic	Sequential Sequential	Static Semi	Discrete Discrete
Poker Backgammon	Partially Fully	Multi Multi	Stochastic Stochastic	Sequential Sequential	Static Static	Discrete Discrete
Taxi driving Medical diagnosis	Partially Partially	Multi Multi	Stochastic Stochastic	Sequential Sequential	Dynamic Dynamic	Continous Continous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continous

Task Env.	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle Chess with clock	Fully Fully	Single Multi	Deterministic Deterministic	Sequential Sequential	Static Semi	Discrete Discrete
Poker Backgammon	Partially Fully	Multi Multi	Stochastic Stochastic	Sequential Sequential	Static Static	Discrete Discrete
Taxi driving Medical diagnosis	Partially Partially	Multi Multi	Stochastic Stochastic	Sequential Sequential	Dynamic Dynamic	Continous Continous
Image analysis Part picking robot	Fully Partially	Single Single	Deterministic Stochastic	Episodic Episodic	Semi Dynamic	Continous Continous

Task Env.	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle Chess with clock	Fully Fully	Single Multi	Deterministic Deterministic	Sequential Sequential	Static Semi	Discrete Discrete
Poker Backgammon	Partially Fully	Multi Multi	Stochastic Stochastic	Sequential Sequential	Static Static	Discrete Discrete
Taxi driving Medical diagnosis	Partially Partially	Multi Multi	Stochastic Stochastic	Sequential Sequential	Dynamic Dynamic	Continous Continous
Image analysis Part picking robot	Fully Partially	Single Single	Deterministic Stochastic	Episodic Episodic	Semi Dynamic	Continous Continous
Refinery con-Troller Interactive. English tutor	Partially Partially	Single Multi	Stochastic Stochastic	Sequential Sequential	Dynamic Dynamic	Continuous Discrete

The structure of agents:

- Agent’s structure can be viewed as −
- Agent = Architecture + Agent Program
- Architecture = the machinery that an agent executes on.
- Agent Program = an implementation of an agent function.

The structure of agents

Agent : An agent is just something that acts (agent comes from the Latin agere, to do). Of course, all computer programs do something, but computer agents are expected to do more: operate autonomously, perceive their environment, persist over a prolonged time period, adapt to change, and create and pursue goals.

A rational agent is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome

agent= architecture + program

run on some sort of computing device with physical sensors and actuators—we call this the architecture

1.Agent program is an implementation of an agent function

Four basic types in order of increasing generality:

- simple reflex agents
- reflex agents with state
- goal-based agents
- utility-based agents

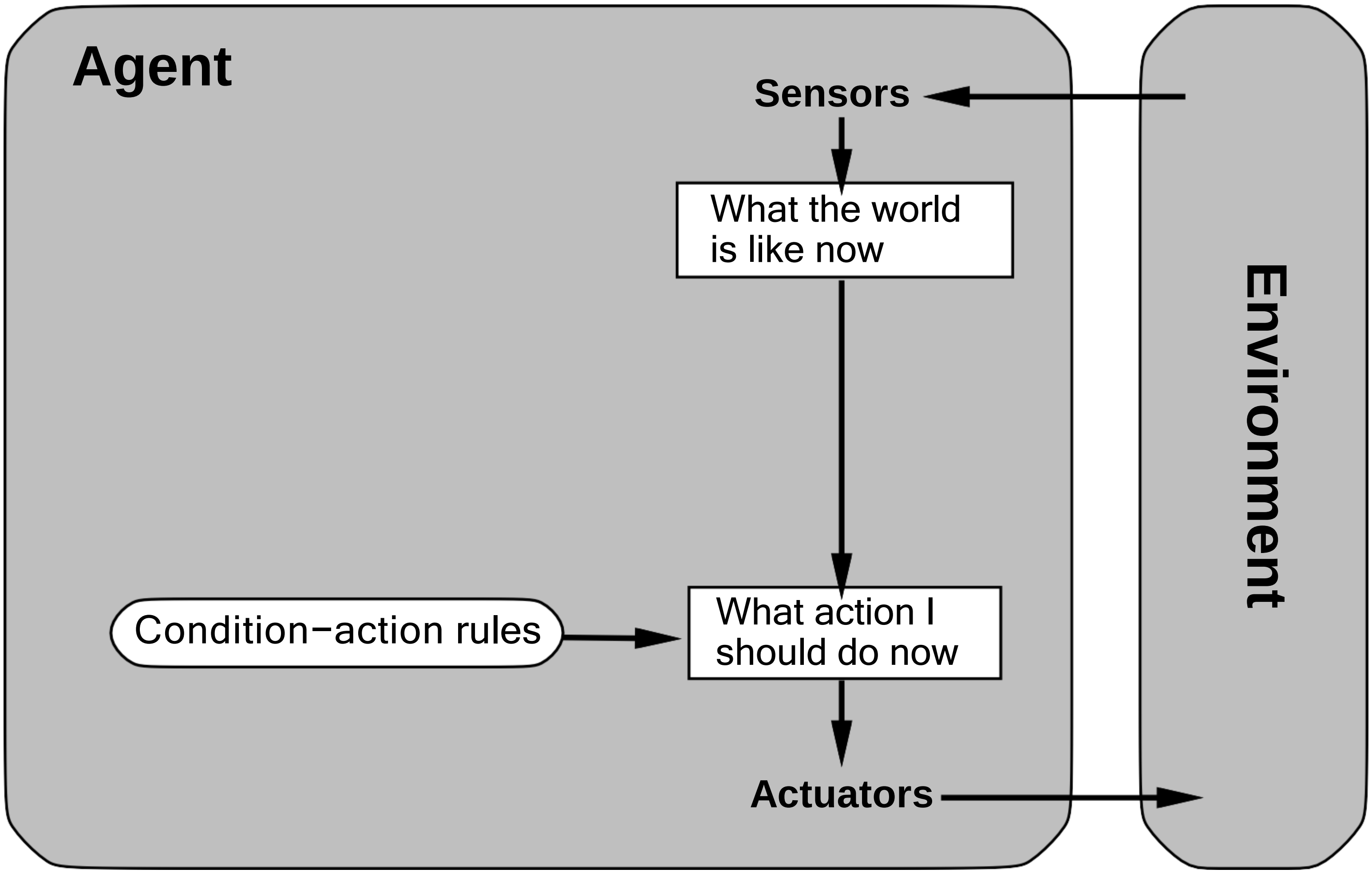
All these can be turned into learning agents

1. Simple reflex agents: The simplest kind of agent is the simple reflex agent. These agents select actions on the basis of the current percept, ignoring the rest of the percept history. It performs actions based on a current situation.
They choose actions only based on the current percept.
They are rational only if a correct decision is made only on the basis of current precept.
Their environment is completely observable.
Condition-Action Rule − It is a rule that maps a state (condition) to an action.

```
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 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.

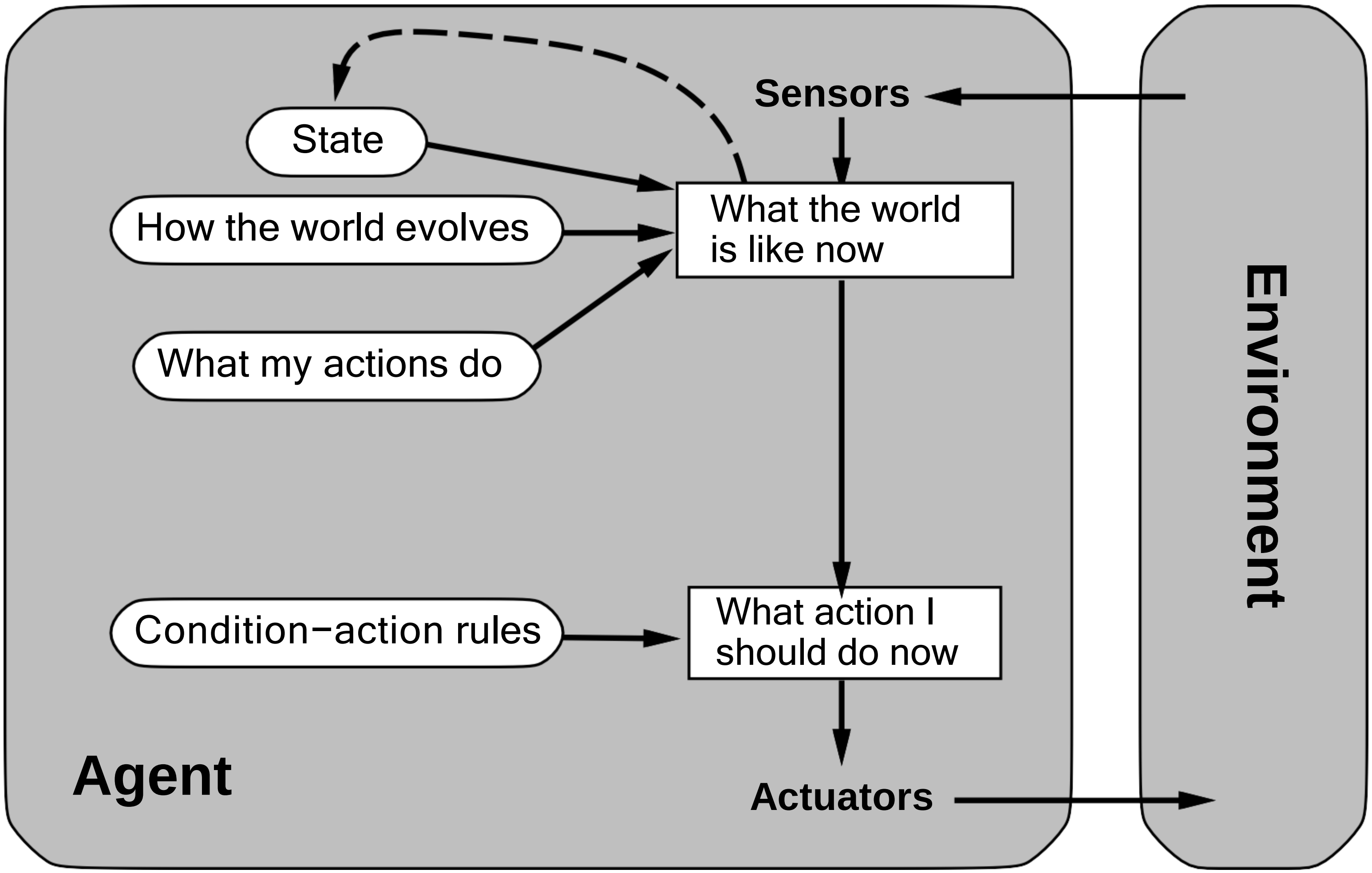
Simple reflex agents



2.Reflex agents with state

- They use a model of the world to choose their actions. They maintain an internal state.
- **Model** – knowledge about "how the things happen in the world".
- **Internal State** – It is a representation of unobserved aspects of current state depending on percept history.
- **Updating the state requires the information about –**
- How the world evolves.
- How the agent's actions affect the world.

Reflex agents with state



Ex: best guess for what the world is like now.

Example

function REFLEX-VACUUM-AGENT([*location*,*status*]) returns an action
static: *last_A, last_B*, numbers, initially ∞

if *status* = *Dirty* then . . .

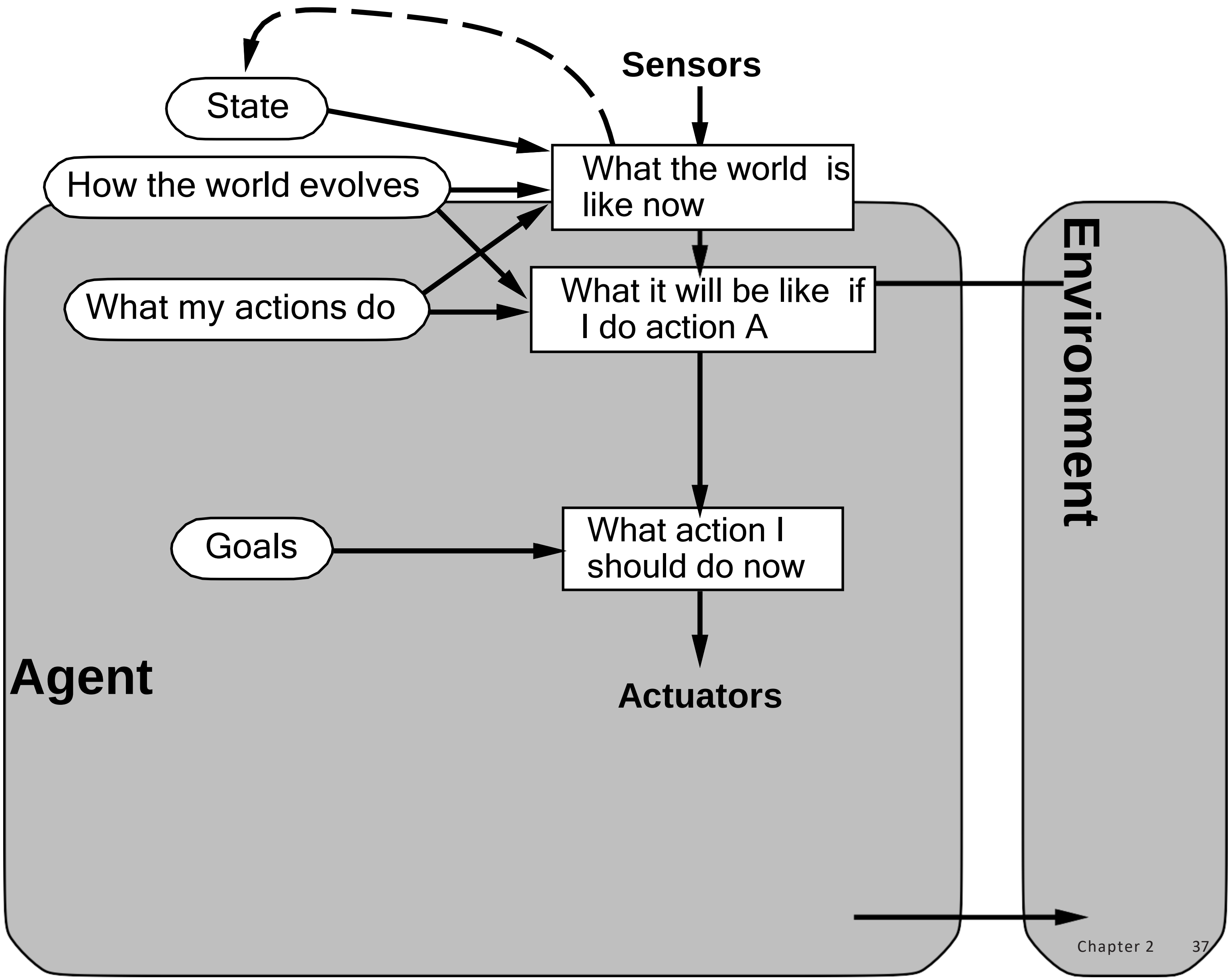
```
(defun make-reflex-vacuum-agent-with-state-program () (let ((last-A
       infinity) (last-B infinity)) #'(lambda (percept)
(let ((location (first percept)) (status (second percept))) (incf last-
   A) (incf last-B)
(cond
((eq status 'dirty)
(if (eq location 'A) (setq last-A 0) (setq last-B 0))
   'Suck)

      ((eq location 'A) (if (> last-B 3) 'Right 'NoOp))
      ((eq location 'B) (if (> last-A 3) 'Left 'NoOp)))))))
```

3.Goal-based agents:

- They choose their actions in order to achieve goals. Goal-based approach is more flexible than reflex agent since the knowledge supporting a decision is explicitly modeled, thereby allowing for modifications.
- **Goal** – It is the description of desirable situations.

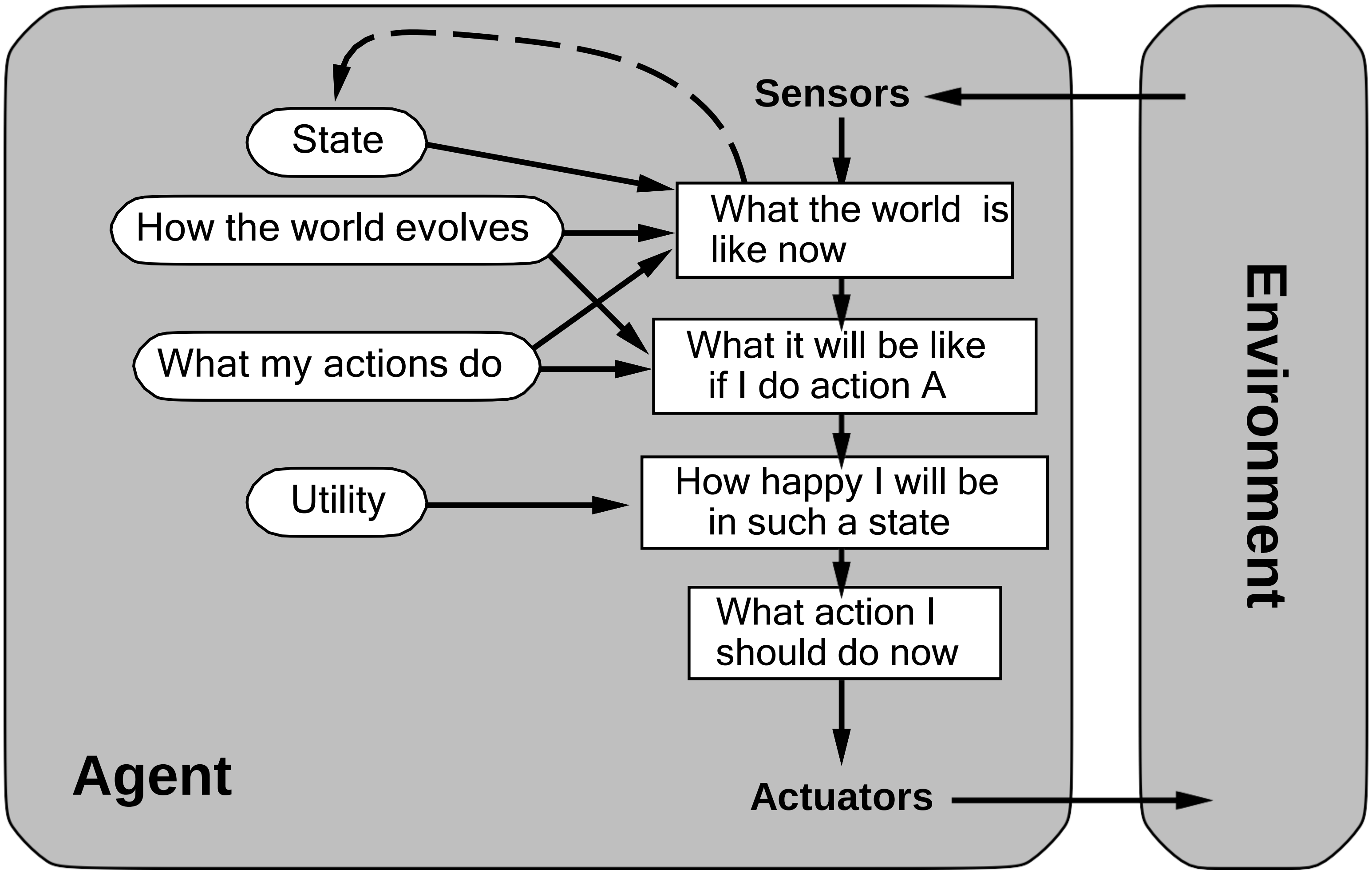
Goal-based agents



4.Utility Based Agents

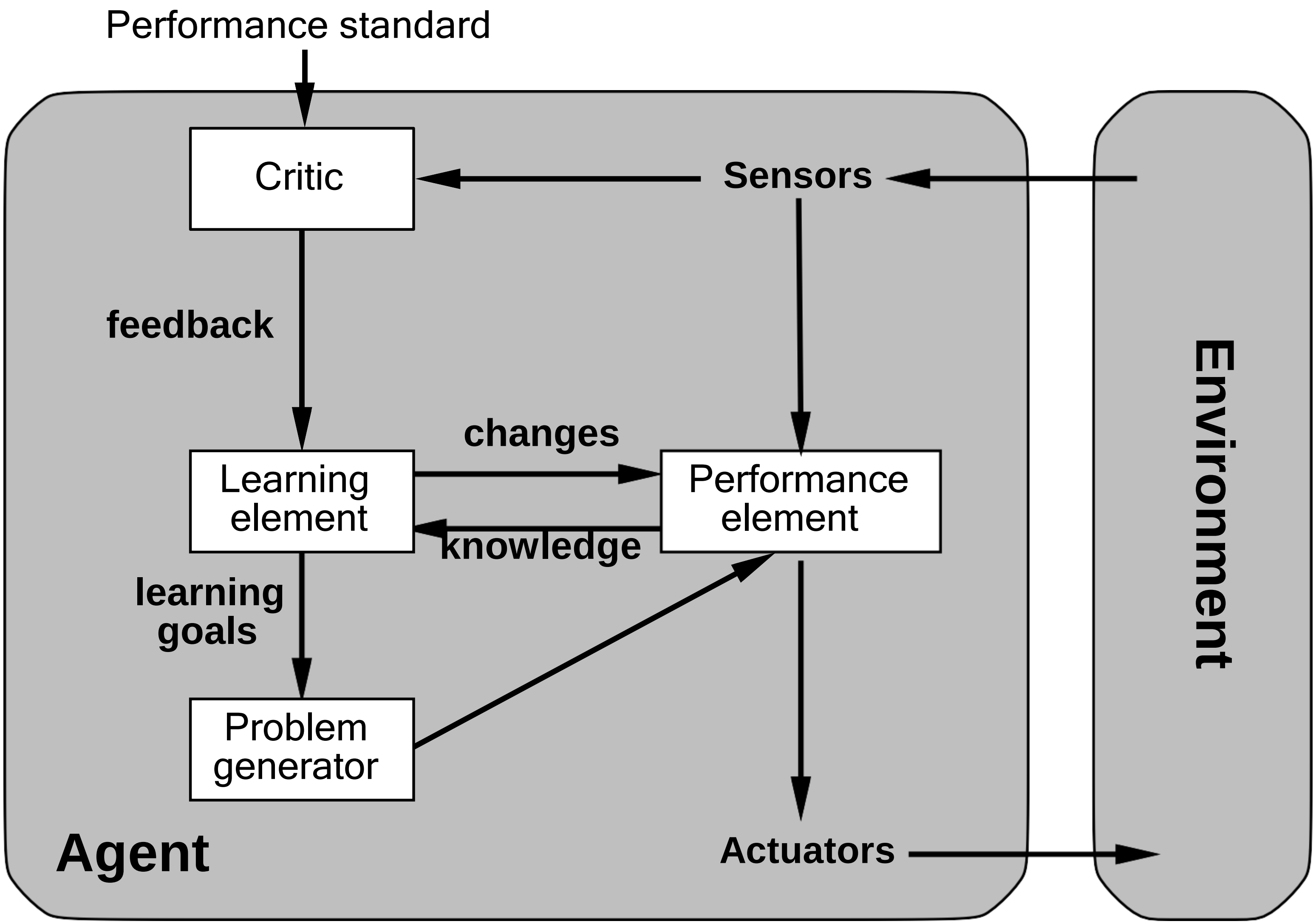
- They choose actions based on a preference (utility) for each state.
- Goals are inadequate when –
- There are conflicting goals, out of which only few can be achieved.
- Goals have some uncertainty of being achieved and you need to weigh likelihood of success against the importance of a goal.

Utility-based agents



Ex: not only happy/unhappy, handle conflicting goals, uncertain goals.

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 define task environments Environments are

categorized along several dimensions:

observable? deterministic? episodic? static? discrete? single-agent?

Several basic agent architectures exist:

reflex, reflex with state, goal-based, utility-based, learning