Intelligent Agents

Chapter 2, Sections 1–4

Outline

♢ Agents and environments

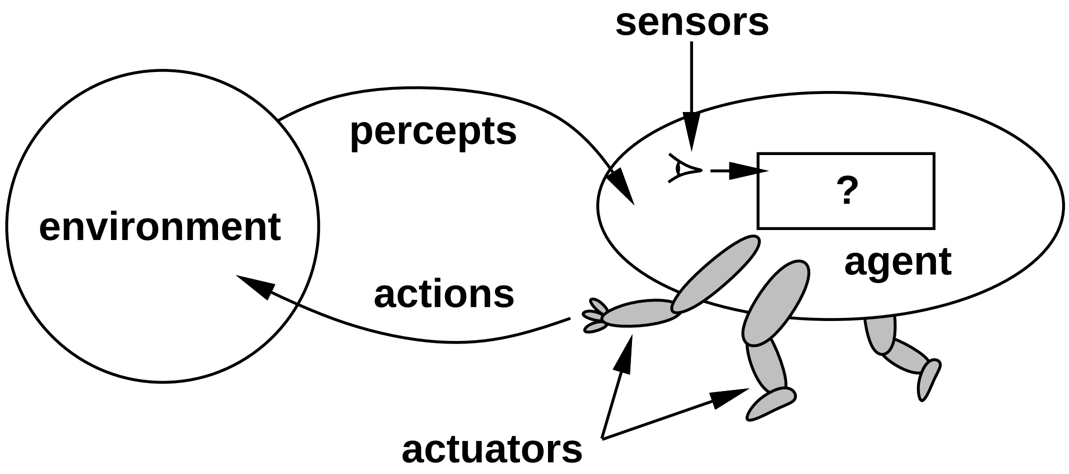
♢ Rationality

♢ PEAS (Performance measure, Environment, Actuators, Sensors)

♢ Environment types

♢ Agent types

Agents and environments



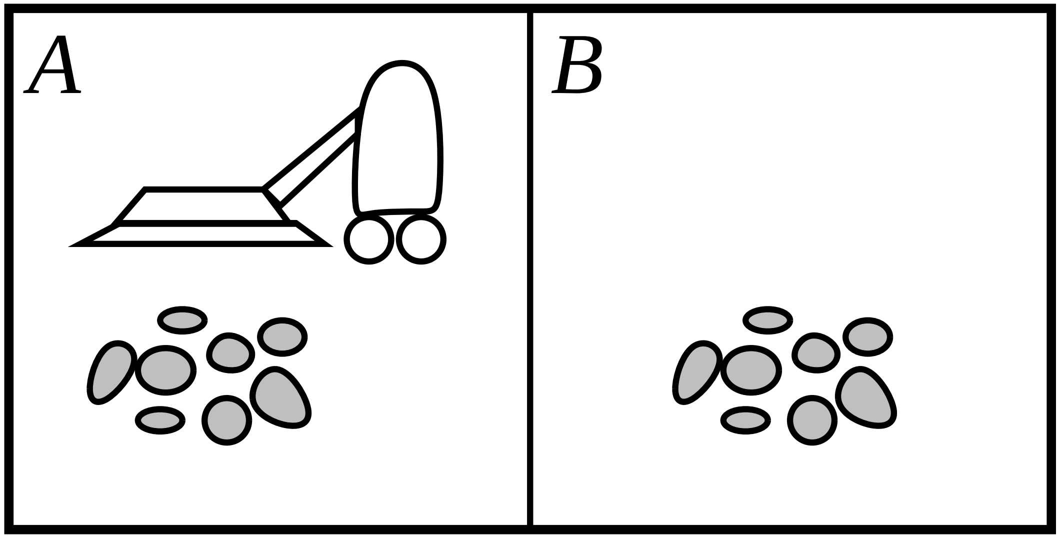
Agents include humans, robots, softbots, thermostats, etc.

The agent function maps from percept histories to actions:

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

The agent program runs on the physical architecture to produce f

Vacuum-cleaner world



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

Actions: $Left$, $Right$, $Suck$, $NoOp$

A vacuum-cleaner agent

A simple agent function is:

> If the current square is dirty, then suck;
> otherwise, move to the other square.

. . . or as pseudo-code:

```
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
```

How do we know if this is a **good** agent function?
What is the **best** function?
Is there one?

Who decides this?

Rationality

Fixed performance measure evaluates the environment sequence

– one point per square cleaned up in time T?
– one point per clean square per time step, minus one per move?
– penalize for $> k$ dirty squares?

A rational agent chooses any action that

– maximizes the expected value of the performance measure
– given the percept sequence to date

Rational $\neq$ omniscient

– percepts may not supply all relevant information

Rational $\neq$ clairvoyant

– action outcomes may not be as expected

Hence, rational $\neq$ successful

Rational $\Rightarrow$ exploration, learning, autonomy

PEAS

To design a rational agent, we must specify the task environment, which consists of the following four things:

Performance measure??

Environment??

Actuators??

Sensors??

Examples of agents:

- Automated taxi,
- Internet shopping agent,
- Boardgames

Automated taxi

The task environment for an automated taxi:

Performance measure?? Safety, destination, profits, legality, comfort, . . .

Environment?? Streets, traffic, pedestrians, weather, . . .

Actuators?? Steering, accelerator, brake, horn, speaker/display, . . .

Sensors?? Video, accelerometers, gauges, engine, keyboard, GPS, . . .

Internet shopping agent

The task environment for an internet shopping agent:

Performance measure?? Price, quality, appropriateness, efficiency

Environment?? Current and future WWW sites, vendors, shippers

Actuators?? Display to user, follow URL, fill in form

Sensors?? HTML pages (text, graphics, scripts)

Question-answering system

The task environment for a question-answering system:

Performance measure?? User satisfaction? Known questions?

Environment?? Wikipedia, Wolfram alpha, ontologies, encyclopedia, . . .

Actuators?? Spoken/written language

Sensors?? Written/spoken input

Question-answering system: Jeopardy!

The task environment for a question-answering system:

Performance measure?? $ $ $

Environment?? Wikipedia, Wolfram alpha, ontologies, encyclopedia, . . .

Actuators?? Spoken/written language, answer button

Sensors?? Written/spoken input

Why did IBM choose Jeopardy as its goal for a QA system?
Because of the peformance measure!

Environment types

	Solitaire	Backgammon	Web shopping	Taxi
Observable??	Fully	Fully	Partly	Partly
Deterministic??	Deterministic	Stochastic	Partly	Stochastic
Episodic??	Sequential	Sequential	Sequential	Sequential
Static??	Static	Static	Semi	Dynamic
Discrete??	Discrete	Discrete	Discrete	Continuous
Single-agent??	Single	Multi	Single*	Multi

*except auctions

The environment type largely determines the agent design

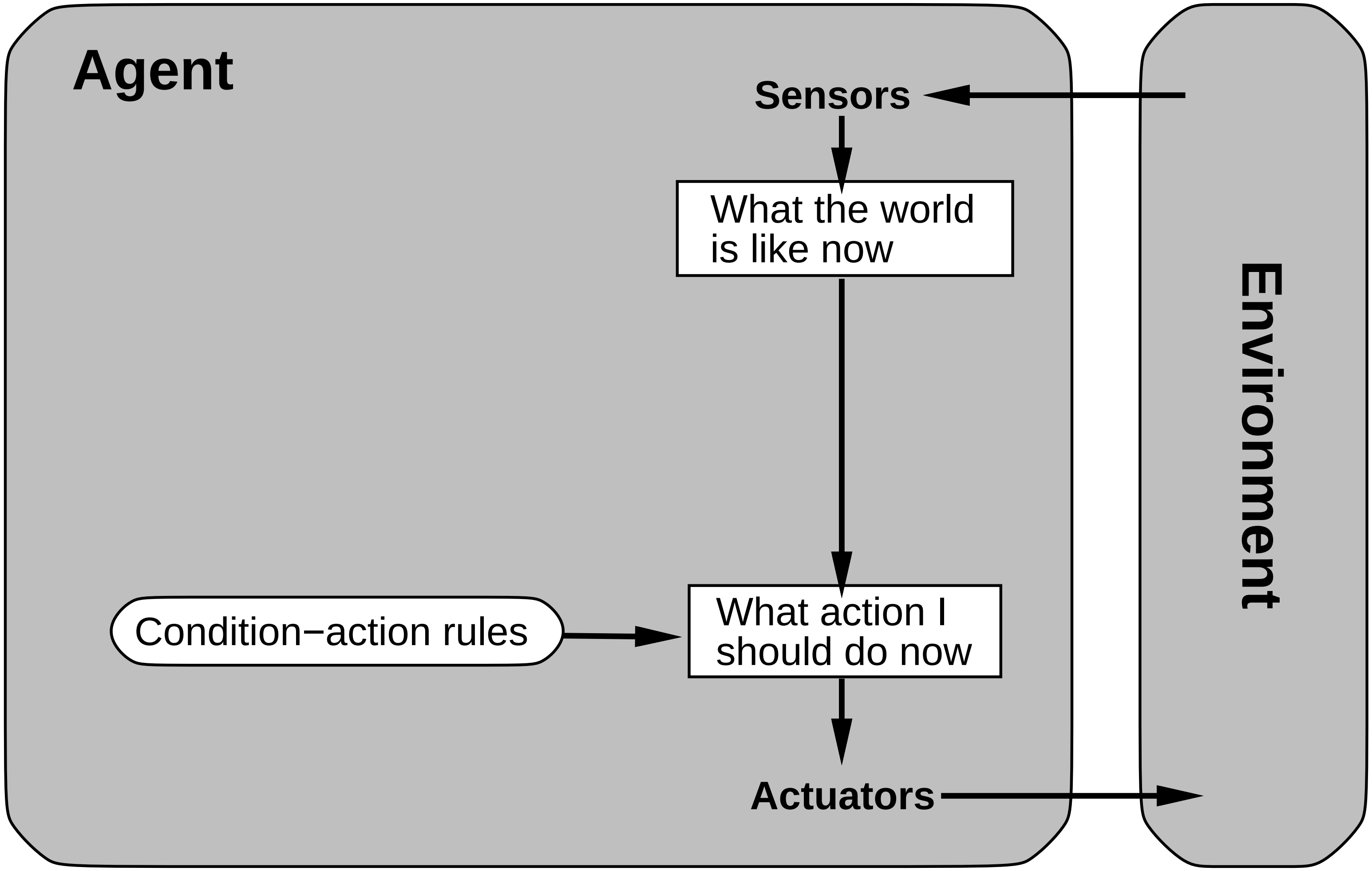
The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Agent types

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

Simple reflex agents

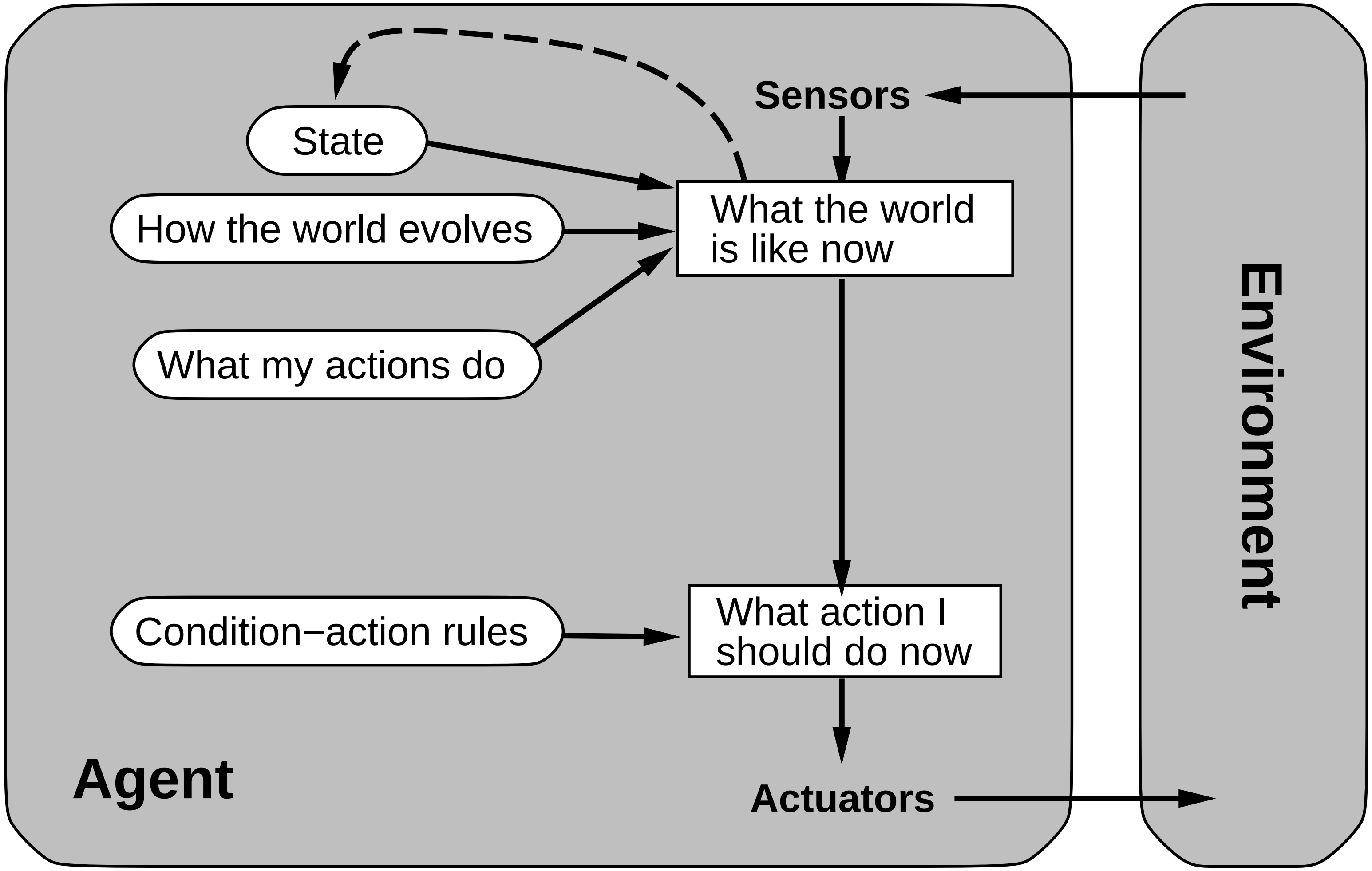


Example

```
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
```

```
def reflex_vacuum_agent(sensors):
    if sensors[’status’] == ’Dirty’:
        return ’Suck’
    elif sensors[’location’] == ’A’:
        return ’Right’
    elif sensors[’location’] == ’B’:
        return ’Left’
```

Reflex agents with state

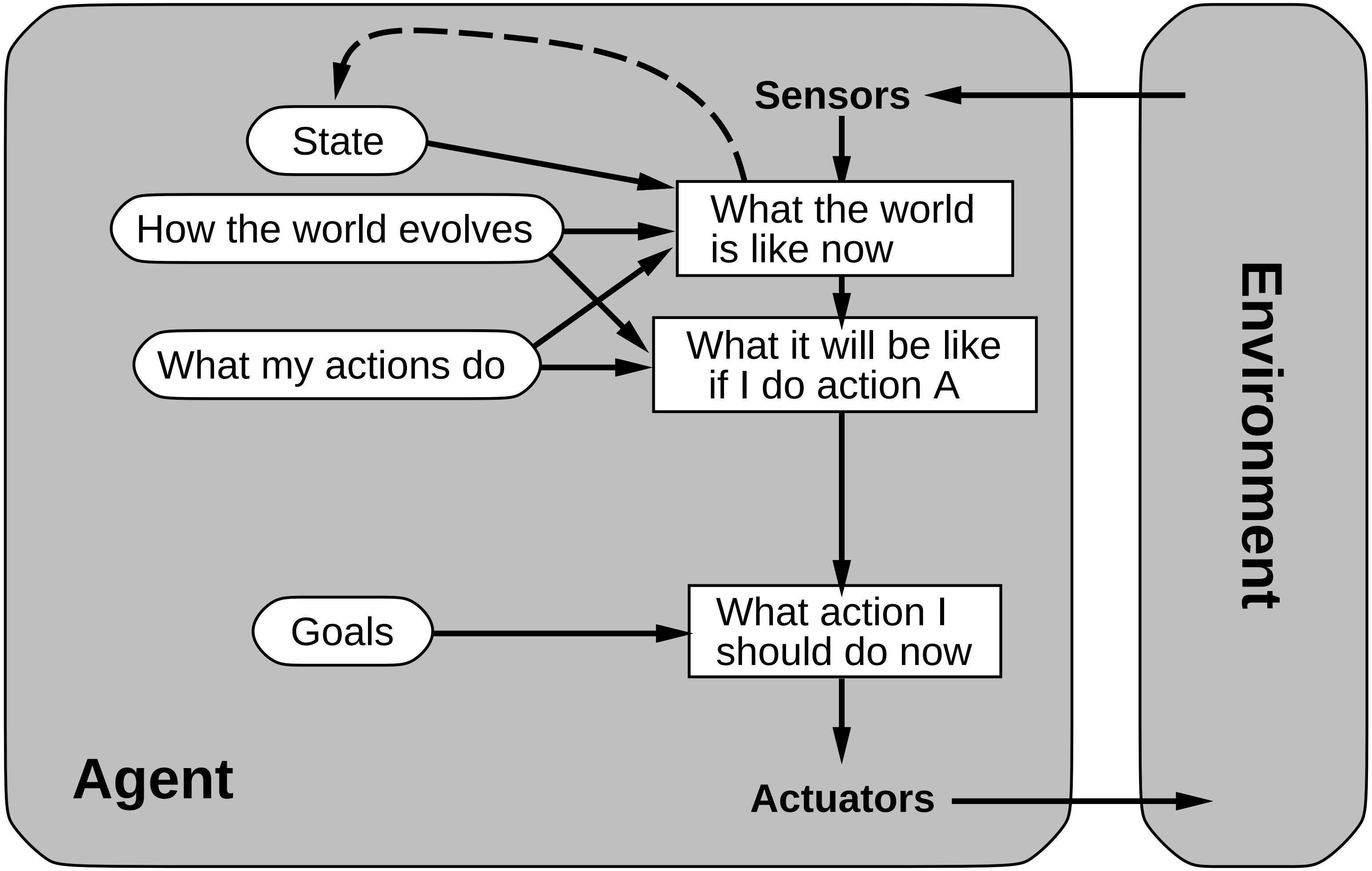


Example

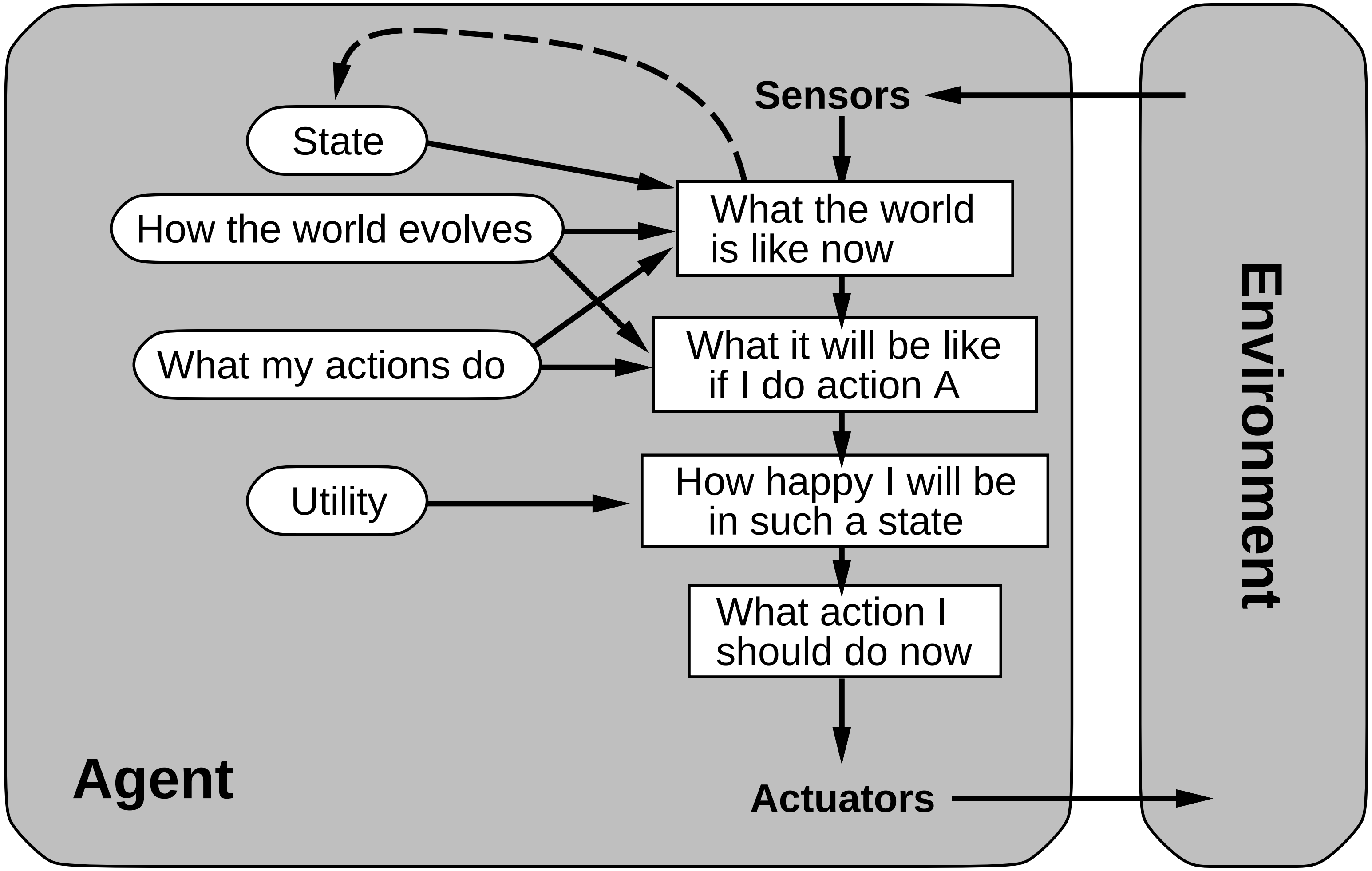
```
function REFLEX-VACUUM-AGENT( [location,status]) returns an action
static: last_A, last_B, numbers, initially ∞

    if status = Dirty then . . .
```

```
initial_state = {’last-A’: maxint, ’last-B’: maxint}
def reflex_vacuum_agent_state(sensors, state=initial_state):
    if sensors[’status’] == ’Dirty’:
        if sensors[’location’] == ’A’:
            state[’last-A’] = 0
        else:
            state[’last-B’] = 0
        return ’Suck’
    elif sensors[’location’] == ’A’ and state[’last-B’] > 3:
        return ’Right’
    elif sensors[’location’] == ’B’ and state[’last-A’] > 3:
        return ’Left’
    else:
        return ’NoOp’
```

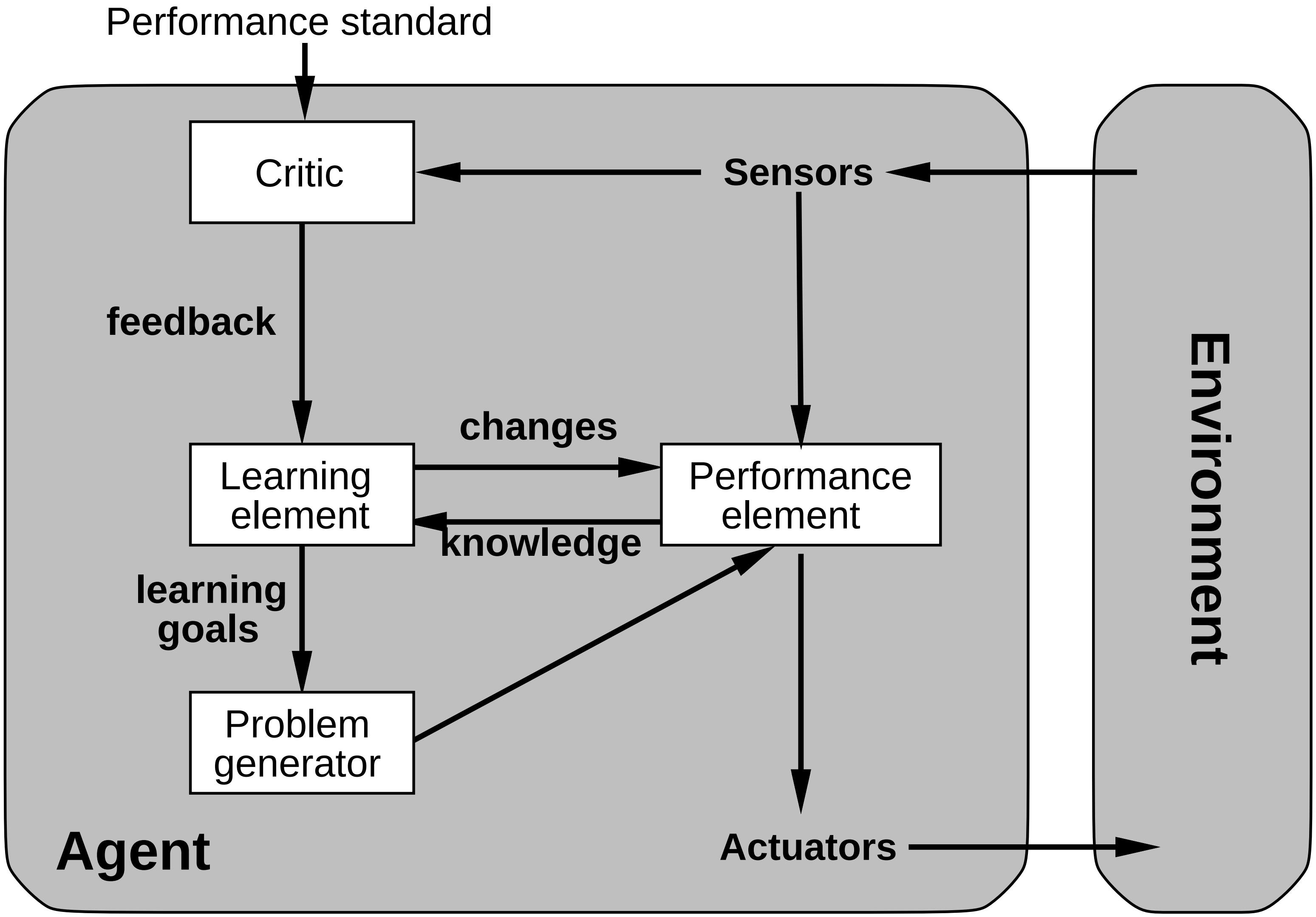
Goal-based agents


Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

Utility-based agents


Sensors
State
How the world evolves
What my actions do
Utility
What the world is like now
What it will be like if I do action A
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

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