Intelligent Agents

Chapter 2

Outline

◇ PAGE (Percepts, Actions, Goals, Environment)

◇ Environment types

◇ Agent functions and programs

◇ Agent types

◇ Vacuum world

PAGE

Must first specify the setting for intelligent agent design

Consider, e.g., the task of designing an automated taxi:

Percepts??

Actions??

Goals??

Environment??

PAGE

Must first specify the setting for intelligent agent design

Consider, e.g., the task of designing an automated taxi:

Percepts?? video, accelerometers, gauges, engine sensors, keyboard, GPS, ...

Actions?? steer, accelerate, brake, horn, speak/display, ...

Goals?? safety, reach destination, maximize profits, obey laws, passenger comfort, ...

Environment?? US urban streets, freeways, traffic, pedestrians, weather, customers, ...

Internet shopping agent

Percepts??

Actions??

Goals??

Environment??

Rational agents

Without loss of generality, "goals" specifiable by performance measure defining a numerical value for any environment history

Rational action: whichever action maximizes the expected value of the performance measure given the percept sequence to date

Rational $\neq$ omniscient
Rational $\neq$ clairvoyant
Rational $\neq$ successful

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Accessible??				
Deterministic??				
Episodic??				
Static??				
Discrete??				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Accessible??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Semi	Semi	No
Discrete??	Yes	Yes	Yes	No

The environment type largely determines the agent design

The real world is (of course) inaccessible, stochastic, sequential, dynamic, continuous

Agent functions and programs

An agent is completely specified by the agent function
mapping percept sequences to actions

(In principle, one can supply each possible sequence to see what it does.
Obviously, a lookup table would usually be immense.)

One agent function (or a small equivalence class) is rational

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

An agent program takes a single percept as input, keeps internal state:

```
function SKELETON-AGENT(percept) returns action
    static: memory, the agent's memory of the world

    memory ← UPDATE-MEMORY(memory, percept)
    action ← CHOOSE-BEST-ACTION(memory)
    memory ← UPDATE-MEMORY(memory, action)
    return action
```

AIMA code

The code for each topic is divided into four directories:

- `agents`: code defining agent types and programs
- `algorithms`: code for the methods used by the agent programs
- `environments`: code defining environment types, simulations
- `domains`: problem types and instances for input to algorithms

(Often run algorithms on domains rather than agents in environments.)

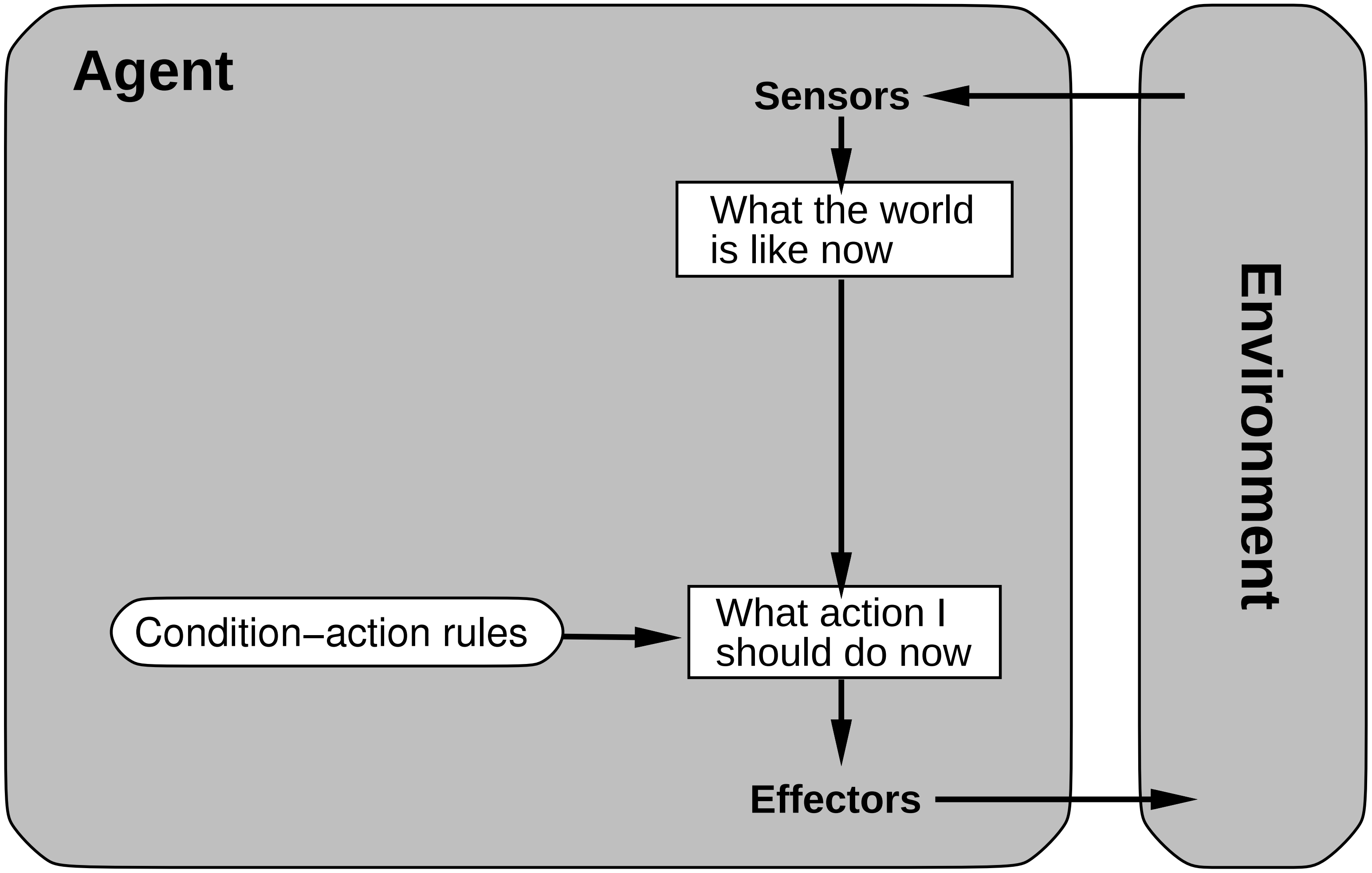
```
(setq joe (make-agent :name 'joe :body (make-agent-body)
                      :program (make-dumb-agent-program)))

(defun make-dumb-agent-program ()
  (let ((memory nil))
    #'(lambda (percept)
        (push percept memory)
        'no-op)))
```

Agent types

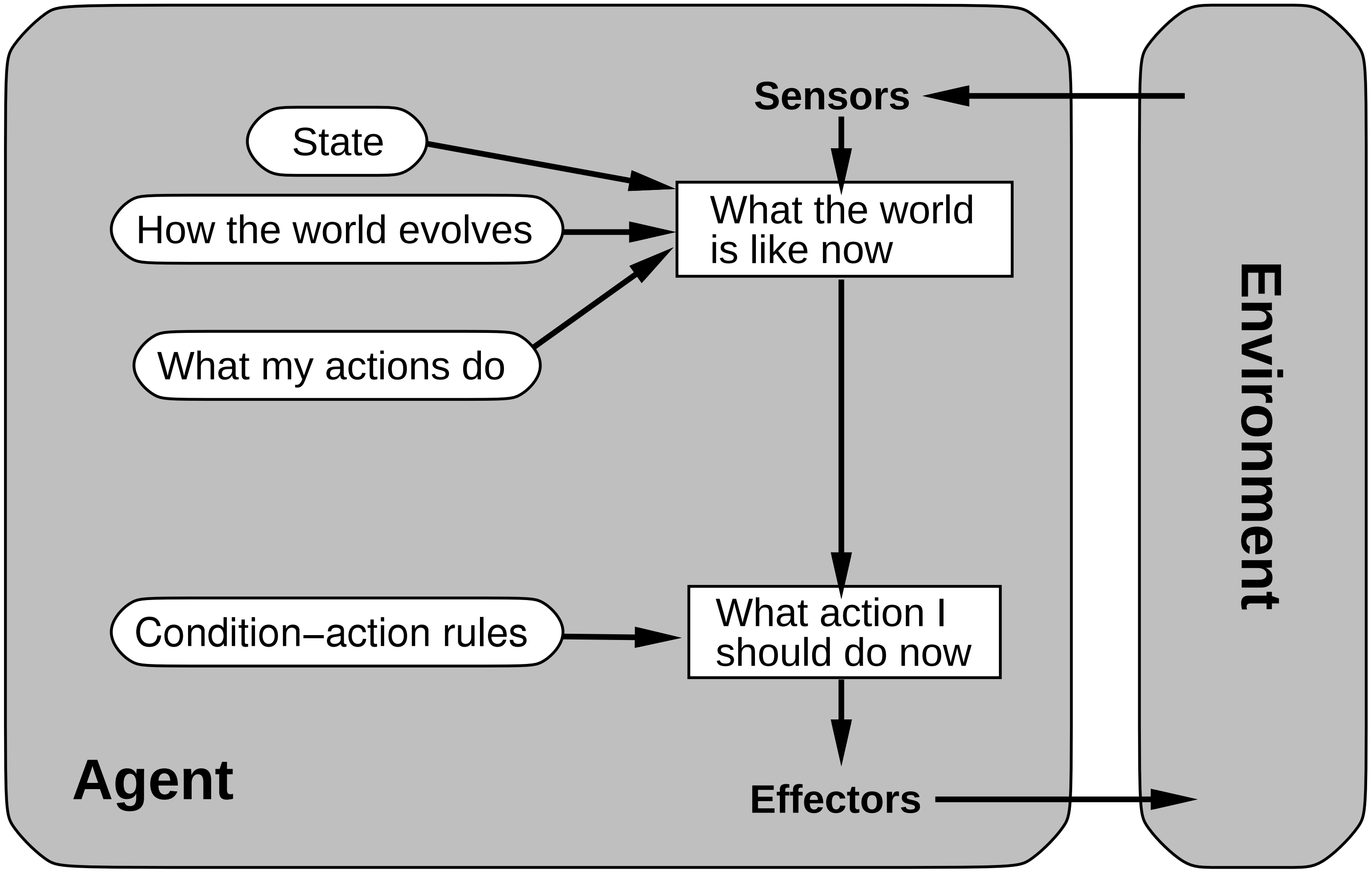
Four basic types in order of increasing generality:

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

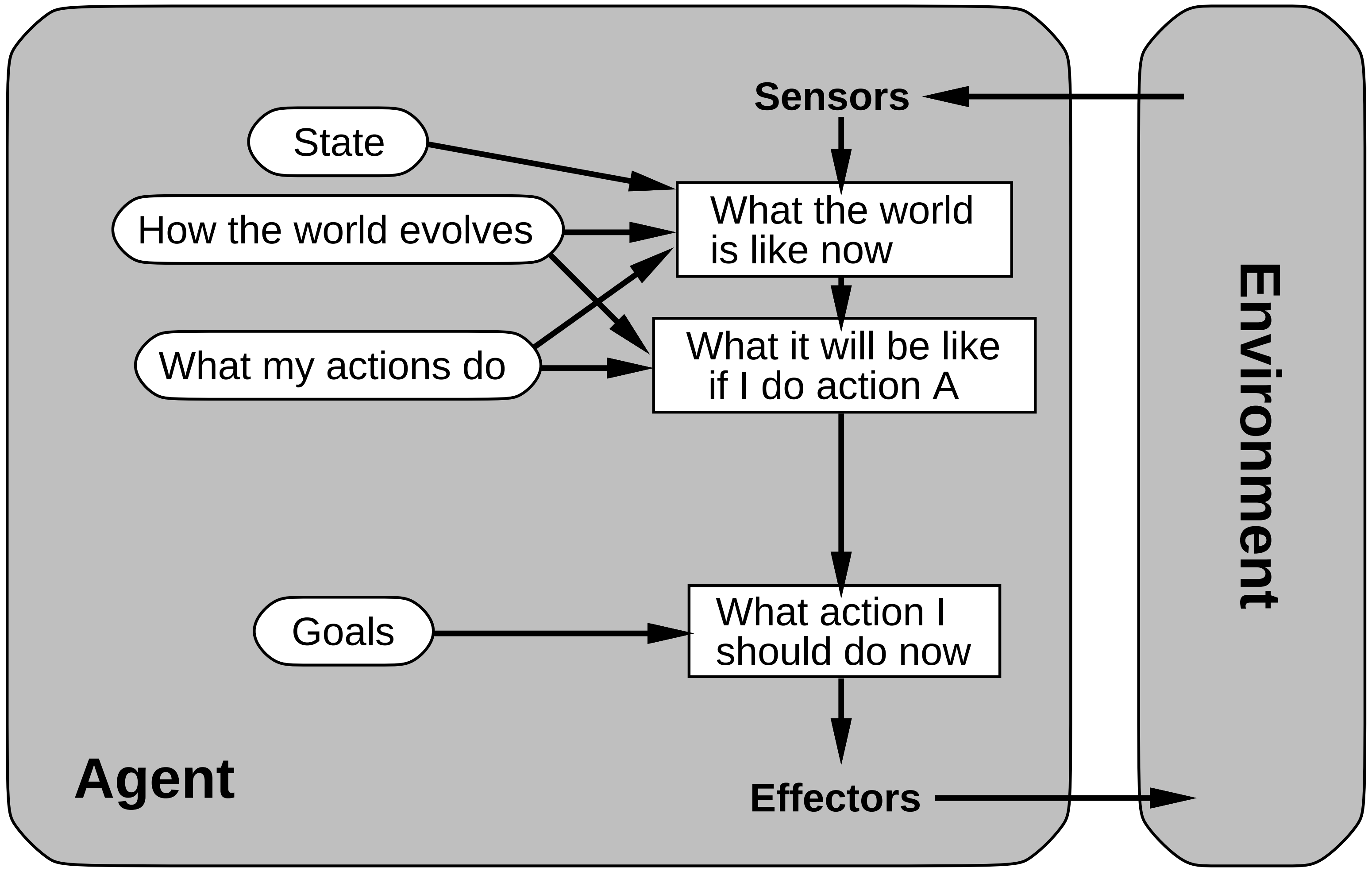
Simple reflex agents



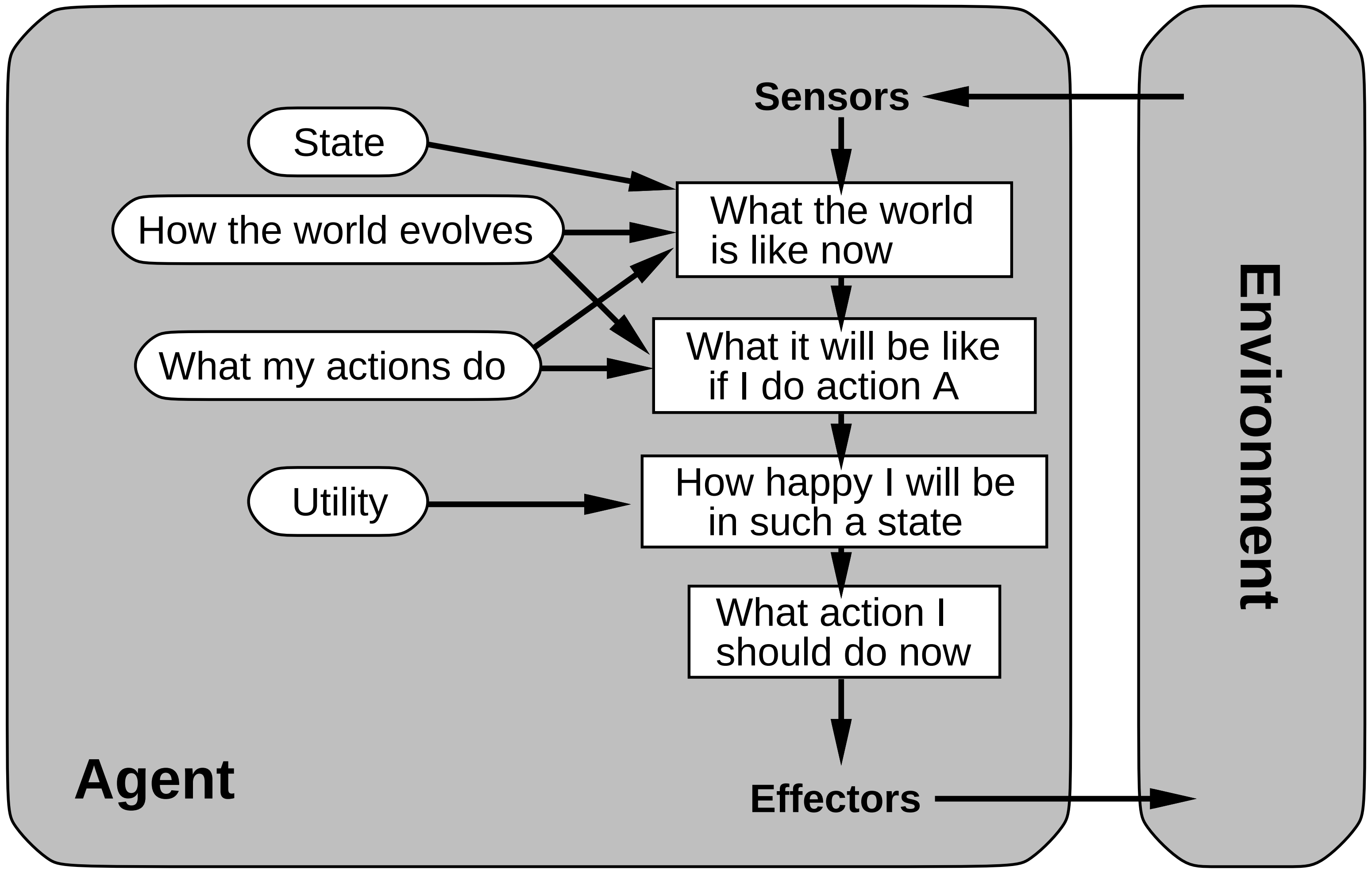
Reflex agents with state



Goal-based agents

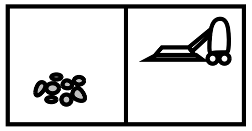

Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Effectors
Environment

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
Effectors
Environment

The vacuum world

`code/agents/environments/vacuum.lisp`



Percepts (`<bump> <dirt> <home>`)

Actions `shutoff forward suck (turn left) (turn right)`

Goals (performance measure on environment history)
- +100 for each piece of dirt cleaned up
- -1 for each action
- -1000 for shutting off away from home

Environment
- grid, walls/obstacles, dirt distribution and creation, agent body
- movement actions work unless bump into wall
- suck actions put dirt into agent body (or not)

Accessible? Deterministic? Episodic? Static? Discrete?