# Intelligent Agents

Chapter 2

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

- Human agent:
  - eyes, ears, and other organs for sensors;
  - hands, legs, mouth, and other body parts for actuators

- Robotic agent:
  - cameras and infrared range finders for sensors; various motors for actuators
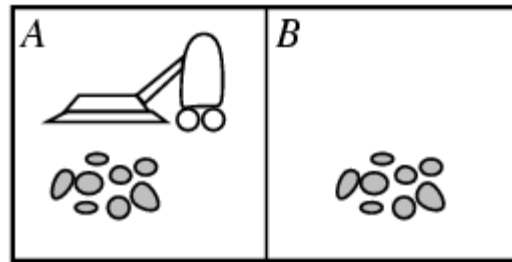
Agents and environments
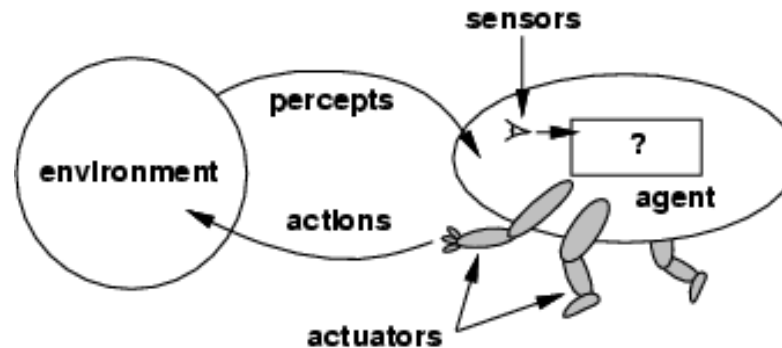
- The agent function maps from percept histories to actions:

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

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

- agent = architecture + program

Vacuum-cleaner world

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

A vacuum-cleaner agent (Tabulation of Agent Function)

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

.

.

.

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.

- Rationality is not same as perfection.

- Rationality maximizes expected performance, while perfection maximizes actual performance.

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)

PEAS

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

PEAS

- Must first specify the setting for intelligent agent design
- Consider, e.g., 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, speedometer, GPS, odometer, engine sensors

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

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.

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

Agent functions and programs

- An agent is completely specified by the <u>agent function</u> mapping from percept sequences to actions
- Aim: find a way to implement the rational agent function concisely

Table-lookup agent

function TABLE-Driven-Agent(percept) returns
an action
static: 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

Table-lookup agent

- 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-VACCUM-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

Agent types

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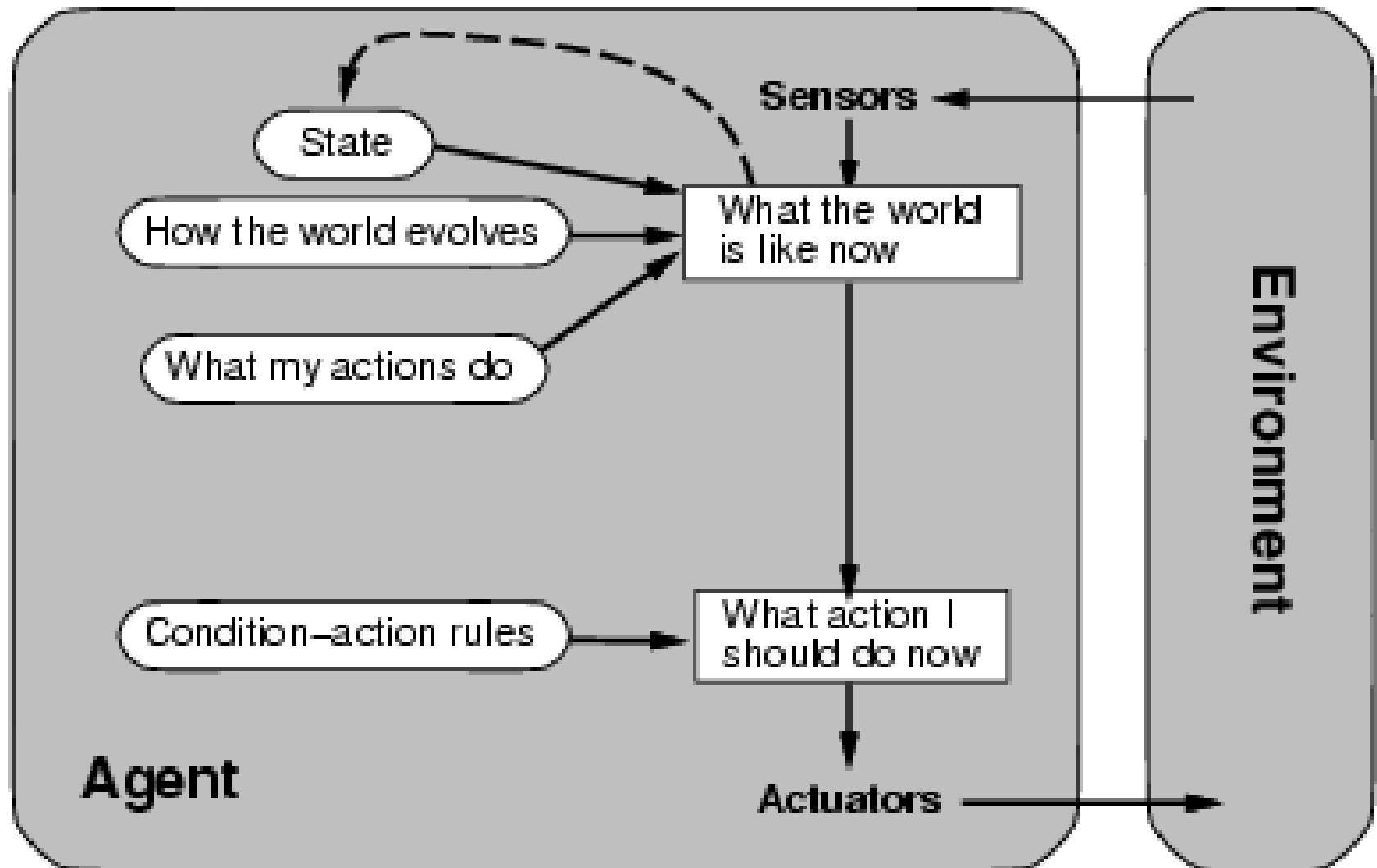   - Goal-based agents
    - Utility-based agents

Simple reflex agents
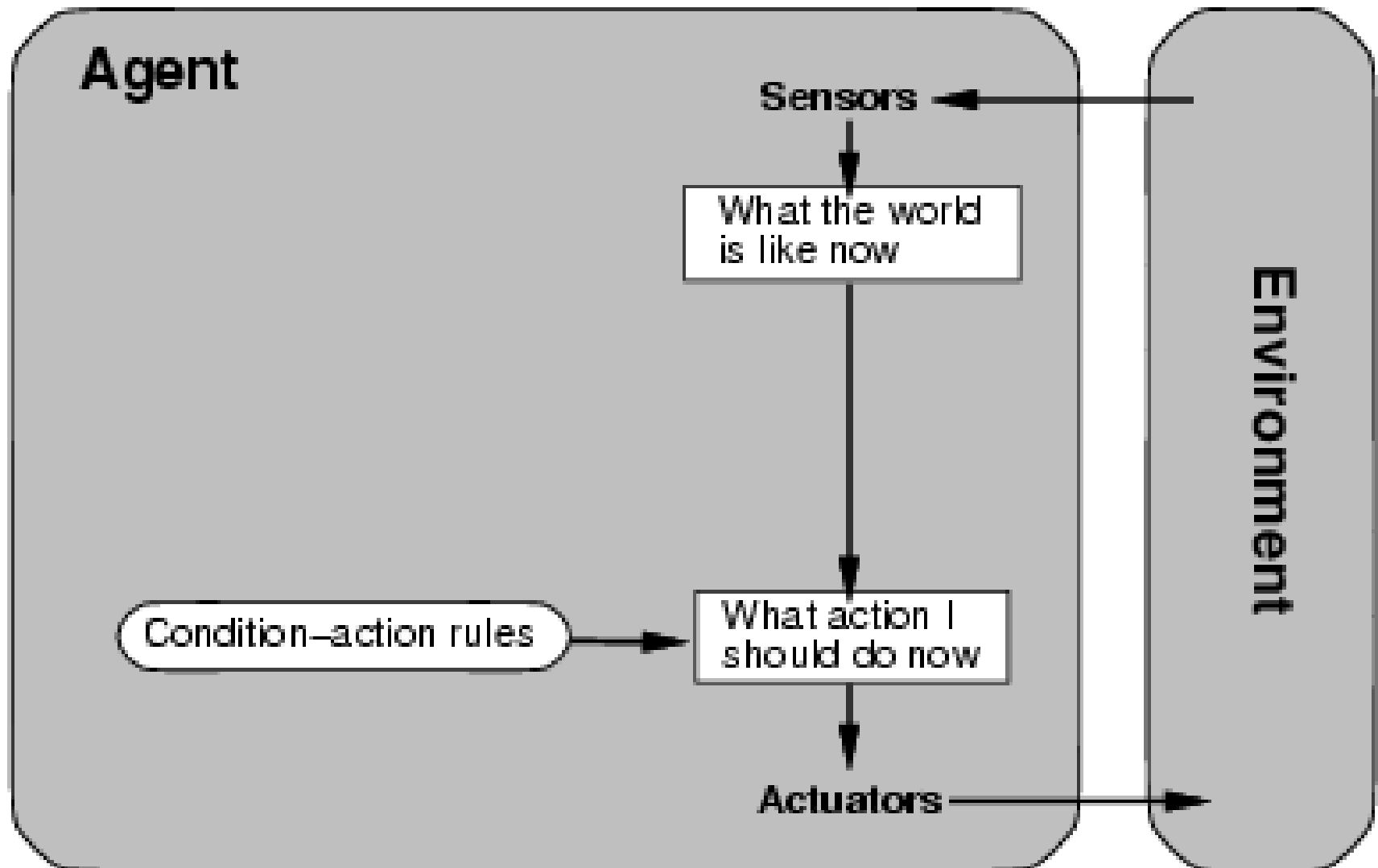
Simple reflex agents

function SIMPLE-REFLEX-AGENT
                    (percepts) returns an action
static: rules, a set of condition-action-rules

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

Model-based reflex agents

# Model-based reflex agents

function REFLEX-AGENT-WITH-STATE
                               (percepts) returns an action
static: state, a description of the current world state
       rules, a set of condition-action-rules
       action, the most recent action, initially none

state ← UPDATE-STATE (state,action,percept)
rule← RULE-MATCH (state, rules)
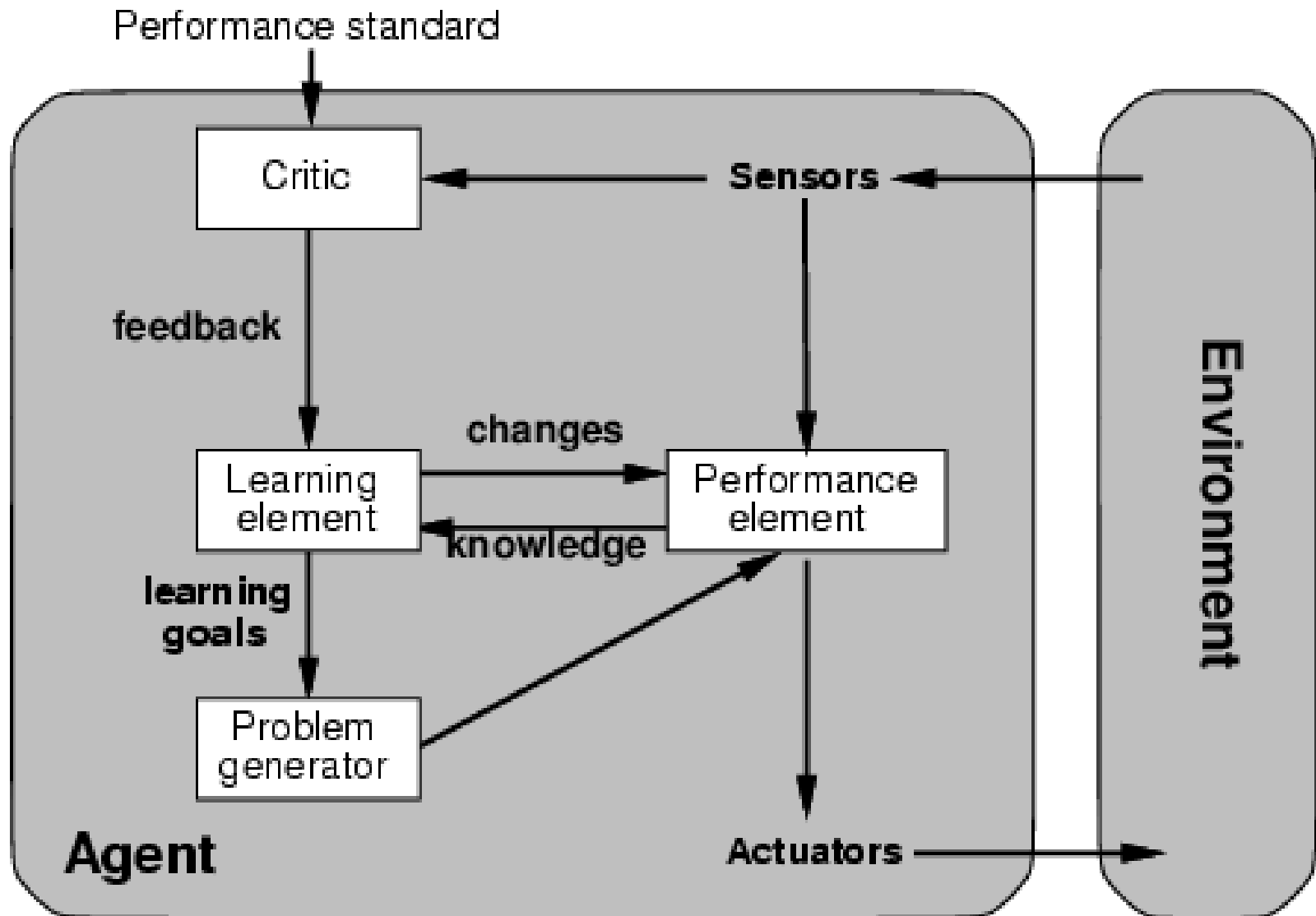action ← RULE-ACTION [rule]
return action

Goal-based agents
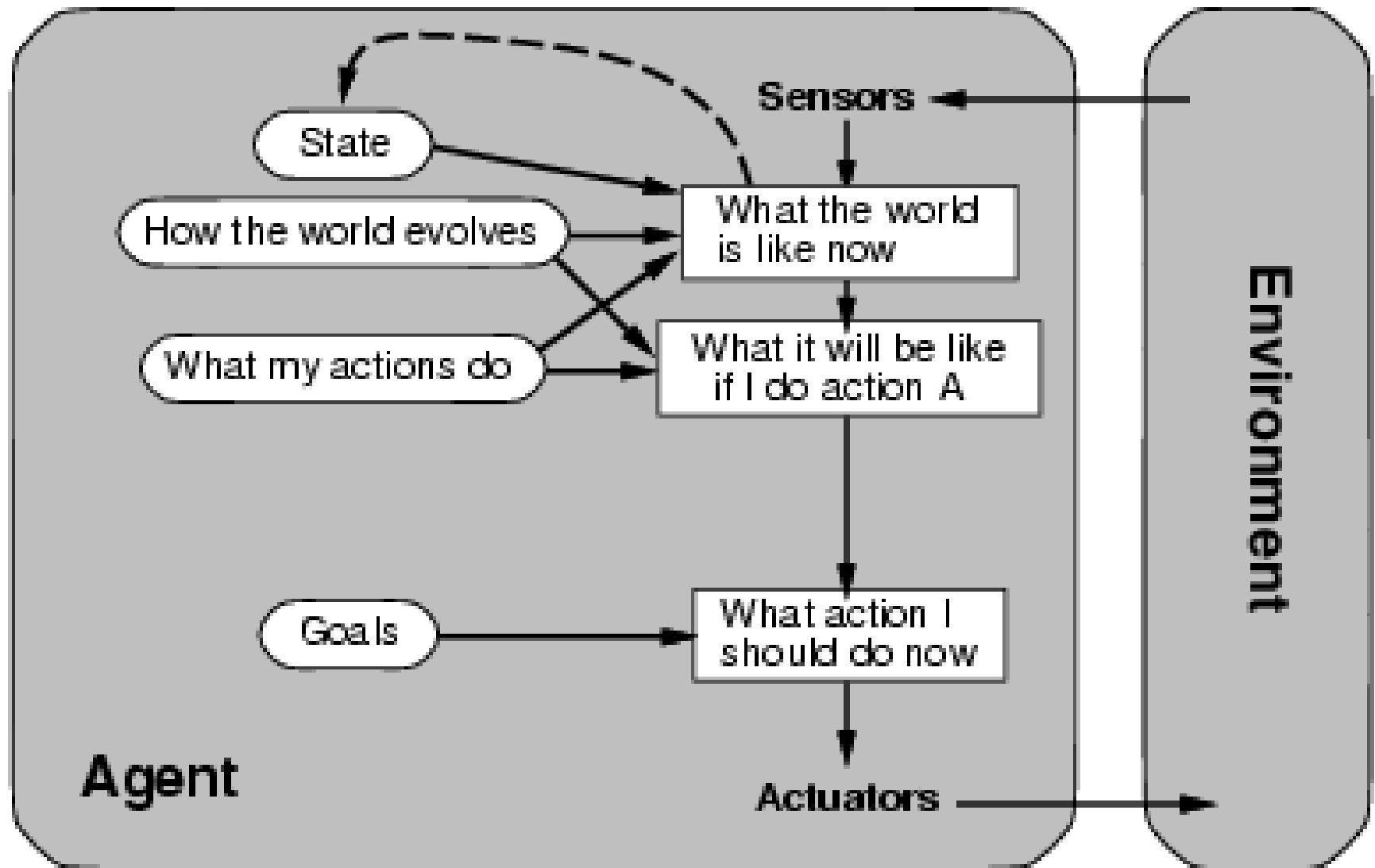
Utility-based agents
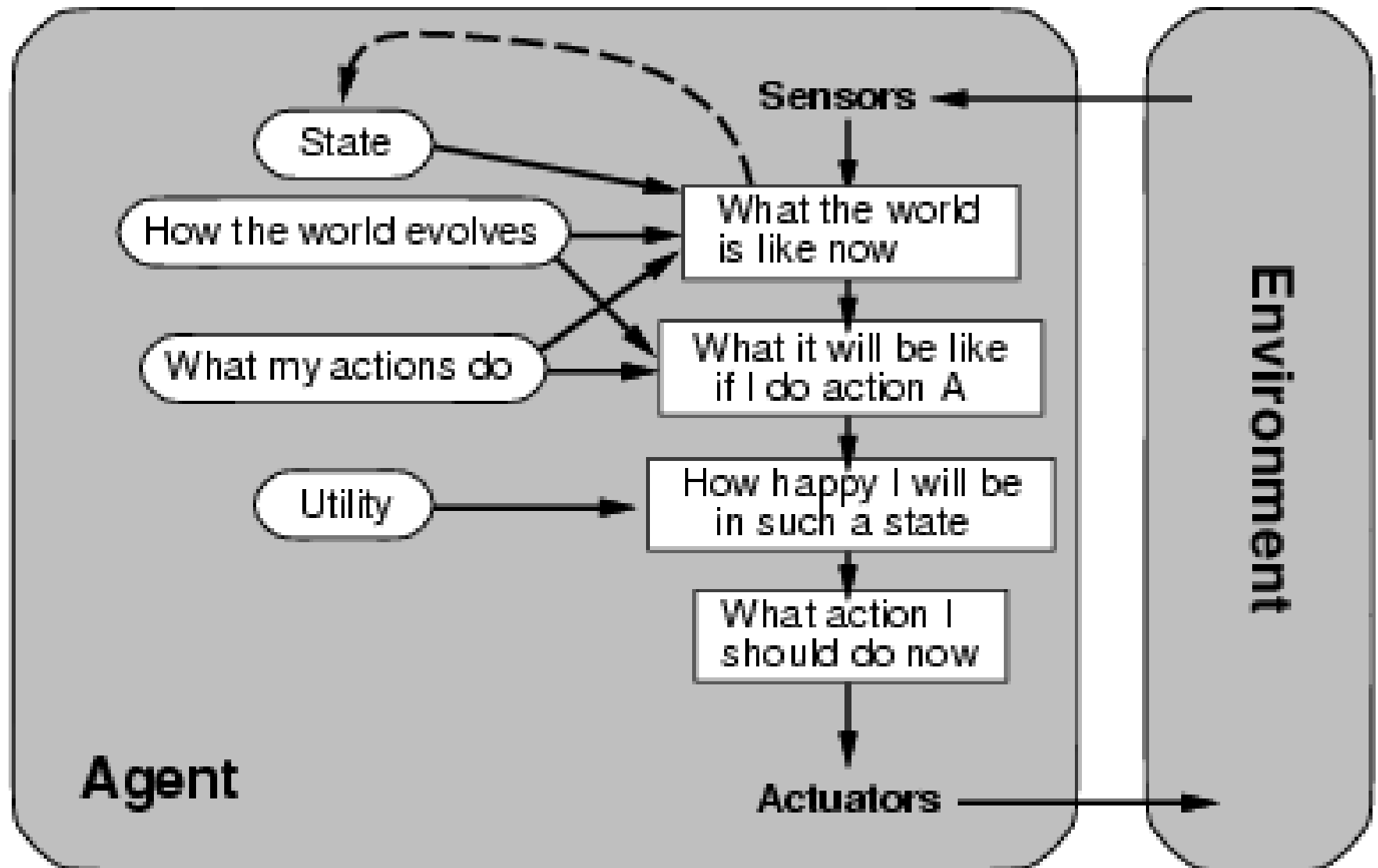
Learning agents

Learning agents

- Four components of a Learning Agent

  - Learning element
    - responsible for making improvements
  - Performance element
    - Responsible for selecting external actions (previously considered to be entire agent)

Learning agents

– Critic

  - The learning element uses feedback from the critic on how the agent is doing and determines how the performance element should be modified to do better in the future

– Problem Generator

  - Responsible for suggesting actions that will lead to new and informative experiences