Intelligent Agent

CHAPTER 2

Outline

- Agents and environments
- Rationality
- PEAS

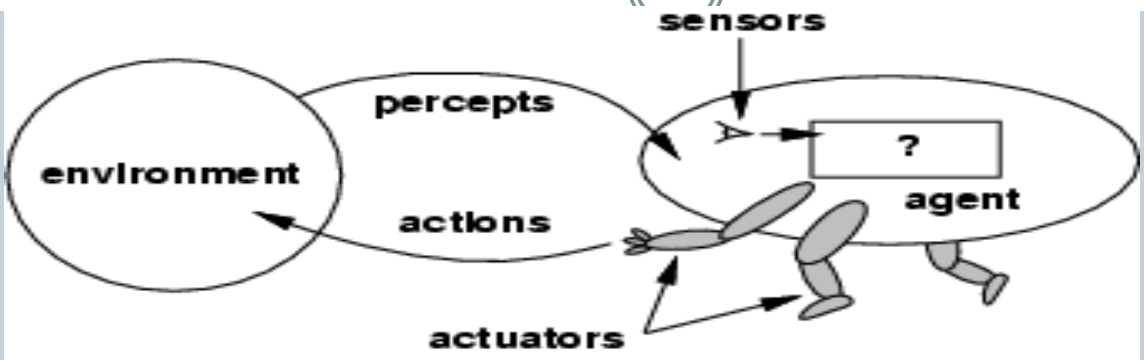
(Performance measure, Environment, Actuators, Sensors)

- 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 effectors.
- Human agent: eyes, ears, and other organs for sensors; hands, legs, mouth, and other body parts for effectors.
- Robotic agent: cameras and infrared range finders for sensors; various motors for effectors.

Agents and environments



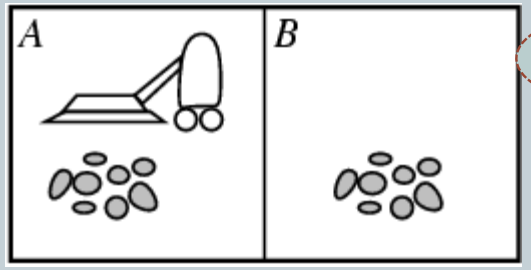
- The agent function maps from percept histories to actions:

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

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

agent = architecture + program

Vacuum-cleaner world



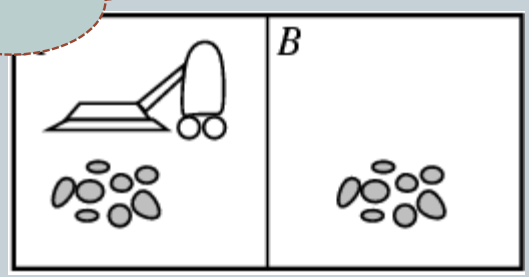
Function: if the current square is dirty, then suck the dirt, otherwise, move to the other square.

- consider the case of an agent that is supposed to vacuum a dirty floor.
- Percepts: location and contents, e.g., [A, Dirty]
- Actions: *Left*, *Right*, *Suck*, *NoOp*

A vacuum-cleaner agent

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

The table is an external characterization of the agent.




```
function Reflex-Vacuum-Agent( [location,statu
if status = Dirty then return Suck
else if location = A then return Right
else if location = B then return Left
```

Internally, the agent function for an artificial agent will be implemented by an **agent program**

?

What makes an agent good or bad, Intelligent or stupid.

Good Behavior: 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.
- 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. If the sequence is desirable, then the agent has performed well.

Rational agents

- Performance measure: An objective criterion for success of an agent's behavior. Obviously, there is not one fixed measure suitable for all agents.

  Example: Vacuum-cleaner.

- performance measure of a vacuum-cleaner agent could be mount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.
- As general rule: it is better to design performance measures according to what one actually wants in the environment, rather than according to how one thinks the agent should behave.

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.*

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

- The **performance measure** that defines degree of success.
- Everything that the agent has perceived so far. We will call this complete perceptual history the **percept sequence.**
- What the **agent knows** about the environment.
- The **actions** that the agent can perform**.**

Rationality

- percepts may not supply all relevant information.
- 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)

Characteristics of agents

- They are **autonomous** … can work on their own.
- They are **persistent** over a prolonged time period.
- They are **adaptive**….adjust to change.
- They are **mobile** ….can be transported over network.
- They have ability to **learn**.

Applications of Intelligent Agents (IA)

- IA's are used to access and navigate information using different search engines.
- IA 's help in decision-making by the knowledge workers.
- IA like voice-activated inference agent reduces the user's task of explicitly commanding the computer.
- IAs perform the time consuming and cumbersome tasks of searching database, doing retrievals and filtering of information and sending the result-sets back to the user, in distributed environments.
- IA can be used to assist mangers to do their job. Some management-oriented tasks that an agent can do are- advising, alter, browse, distribute, explain, filter, guide, match, monitor, navigate, organize, query, report, search.....

Specifying the task environment

To design a rational agent, we must specify the task environment Consider.

- 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

The task of designing an automated taxi:

Performance measure

Safe, fast, legal, comfortable trip, maximize profits.

Environment

Roads (streets/freeways) , traffic, pedestrians, weather, customers.

Actuators

speaker/display, steering wheel, accelerator, brake, signal, horn.

Sensors

Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard.

Internet shopping agent

Performance measure

price, quality, appropriateness, efficiency

Environment

current and future WWW sites, vendors, shippers

Actuators

display to user, follow URL,

Sensors

HTML pages (text, graphics, scripts)

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: if the agent's sensors give it access to the complete state of the environment is fully observable.
- Single agent vs. Multiagent: An agent operating by itself in an environment. (competitive or cooperative multiagent). The agent-design problems in multiagent environment are: *communication*, *randomized behavior*.
- Deterministic vs Stochastic: if the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise, it is stochastic.

Environment types

- Episodic vs. Sequential: In an episodic task environment, the agent's experience is divided into atomic episodes. In each episodes the agent receives a percept and then performs a single action. Crucially, in the next episode does not depend on the actions taken in previous episodes. In sequential environments, the current decision could affect all future decisions.
- Static vs. Dynamic: The environment is changed while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise it is static. Static environments are easy to deal with because the agent need not keep looking at the world while it is deciding on an action, nor need to worry about the passage of time. Dynamic environments, are continuously asking the agent about it wants to do.

(The environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does)

Environment types

- Discrete vs. Continuous: The discrete/continuous distinction applies to the state of the environment, to the way time is handled, and to the percepts and the actions of the agent.
- Known vs. unknown: it is not refers to the environment itself but to the agent's (or designer's) state of knowledge about the "laws of physics" of the environment.

Environment types

- We can expected that, the hardest case is:

  partially observable, multiagent, stochastic, sequential, dynamic, continuous, and unknown.

- Taxi driving is hard in all these senses, expect that for the most part the driver's environment is known.

The Structure of Agent

- The job of AI is to design an Agent Program that implements the agent function (the mapping from percepts to actions).
- This program is assumed to run on some sort of computing device with physical sensors and actuators
- We call this the architecture

Agent = Architecture + Program.

Agent Program

- The agent program implements the agent function.
- The appropriate design of the agent program depends on the nature of the environment.

```
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
```

Agent Programs

- **Why not just look up the answers?**

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 percepts
Action LOOKUP(percept, table)
Return action

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

Failure of lockup table strategy

- No physical agent in this universe will have the space to store the table.
- The designer would not have time to create the table.
- No agent could ever learn all the right table entries from its experience.
- Even if the environment is simple enough to yield a feasible table size, the designer still has no guidance about how to fill in the table entries.
- If the environment changed in some way, the agent would be lost

Agent types

Four basic types in order of increasing generality:

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents.
- Learning agents.

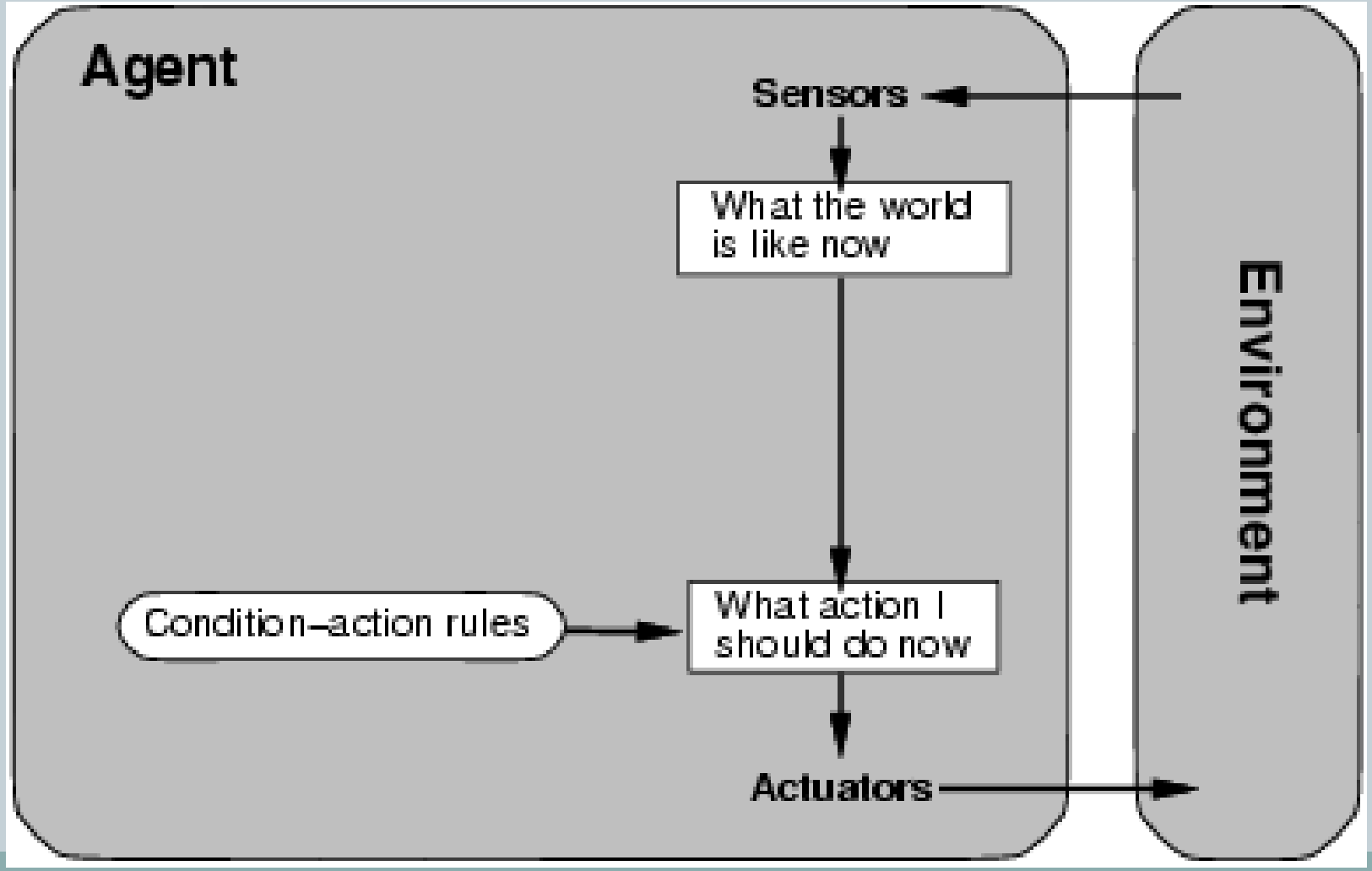
All these can be turned into learning agents

Simple Reflex Agents

- These agents select actions on the basis of the *current* percept, ignoring the rest of the percept history.
- For example, the vacuum agent is a simple reflex agent, because its decision is based only on the current location and on whether that location contains dirt.
- Simple reflex behaviors occur even in more complex environments.
- We can use condition-action rule,

if *car-in-front-is-braking* **then** *initiate-braking*.

Simple reflex agents



Simple reflex agents

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.*

function REFLEX-AGENT-WITH-STATE(*percept*) **returns** *action*
  static: *state*, a description of the current world state
    rules, a set of condition-action rules

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

Problems with simple reflex agents

- The agent **work only if** the correct decision can be made on the basis of only the current percept-that is, **only if** the environment is fully observable.
- Infinite loops **are often unavoidable** for simple reflex agents operating in partially observable environment.
- Escape from infinite loops is possible if the agent can randomize its action.

Model-based reflex agents

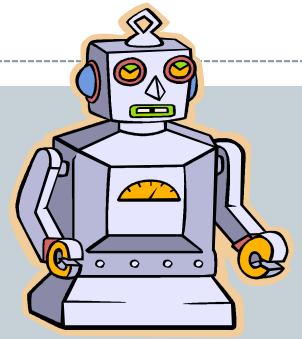
- The most effective way to handle partial observability is for the agent to keep track of the part of the world it can't see now.
- The agent should maintain some sort of **internal state** that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state.
- Example: *for the driving tasks of changing paths, the agent needs to keep track of where the other cars are if it can't see them all at once.*

Model-based reflex agents

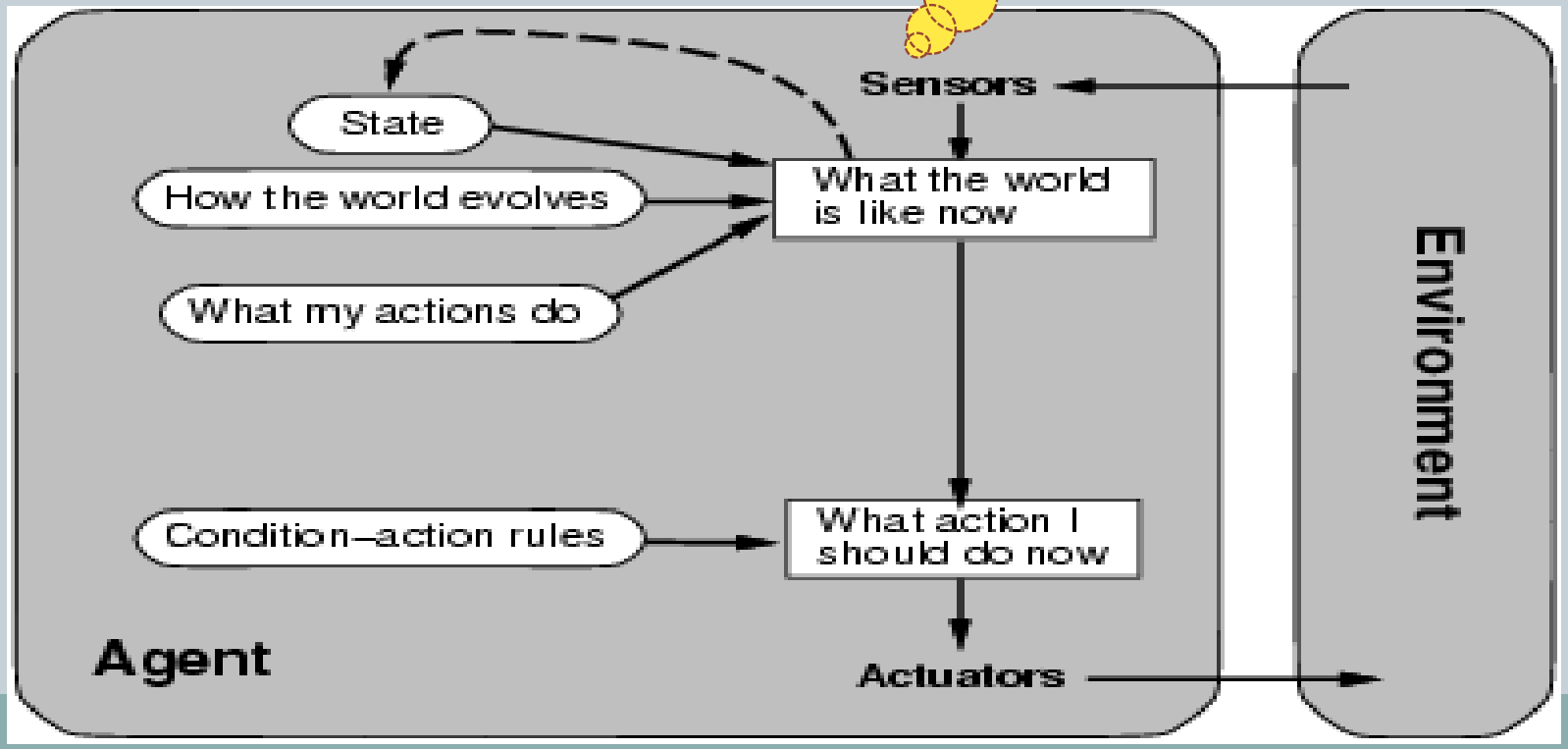
- Updating this internal state information requires two kinds of knowledge to be encoded in the agent program.
- First: we need some information about how the world evolves independently of the agent.
- Second: we need some information about how the agent's own actions affect the world.

An agent that uses a model is called a model-based agent.

Model-based reflex agents



It shows how the current percept is combined with the old internal state to generate the updated description of the current state, based on the agent's model of how the world work.



Model-based reflex agents

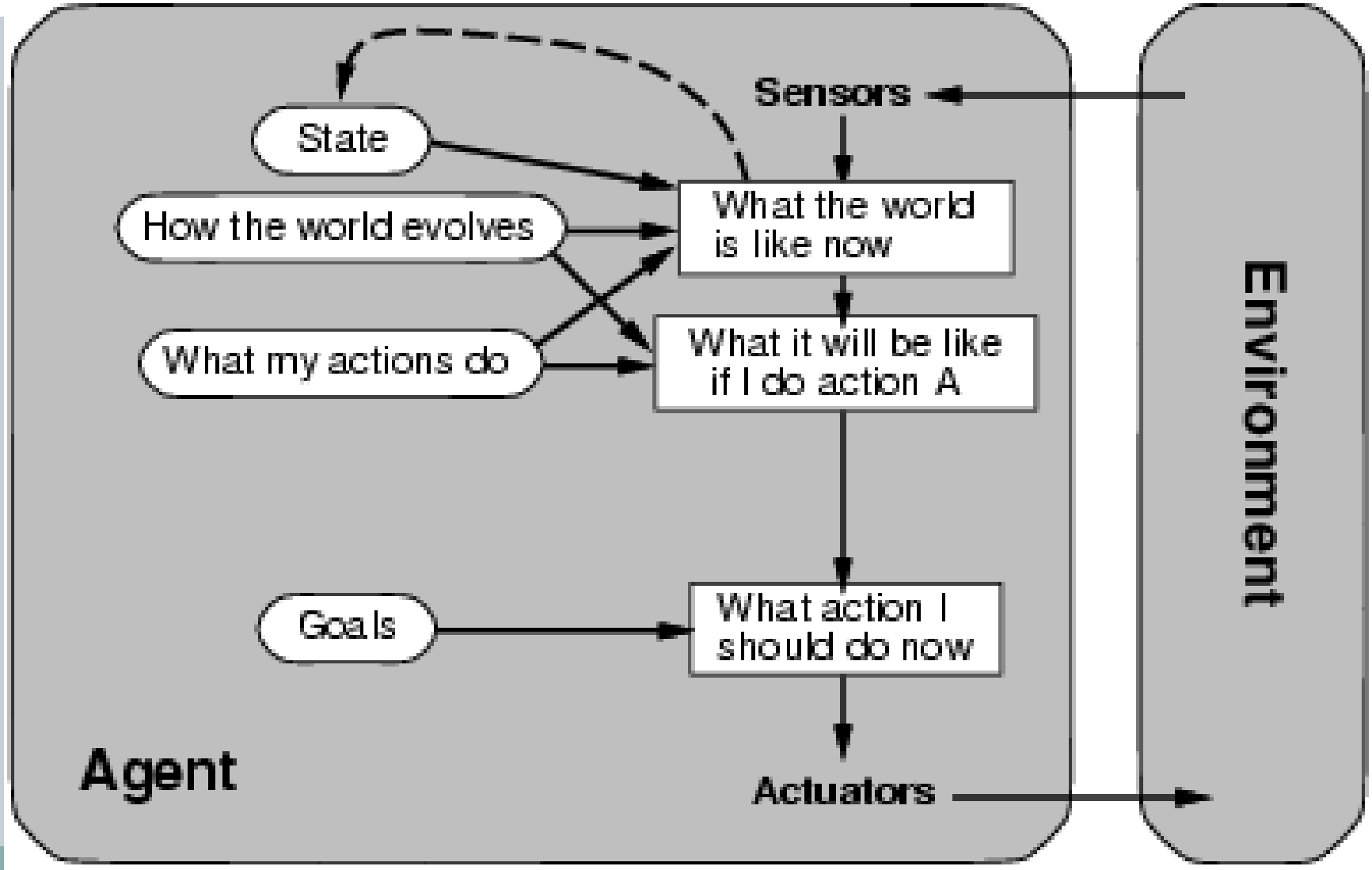
Function : MODEL-BASED-REFLEX-AGENT (percept) returns an action
Persistent: state, the agent's current conception of the world state
model, a description of how the next state depends on current state and action
rules, a set of condition-action rules
action, the most recent action, initially none

State ⟵ UPDATE-STATE(state, action, percept, model)
Rule ⟵ RULE-MATCH(state, rules)
Action⟵ rule.ACTION
return action

Goal-based agents

- As well as the current description, the agent needs some sort of goal information that describes situations that are desirable.
- Example: the taxi can turn left, turn right or go straight on, the correct decision depends on where the taxi is trying to get to.
- The goal-based action selection could be a straightforward (when the goal satisfaction results immediately from a single action) or a tricky (when the agent has to consider long sequence action in order to find a way to achieve the goal) .

Goal-based agents



Goal-based agents

- The decision making in goal-based agent is different from the condition action rules.

Example: the reflex agent brakes when it sees brake lights. A goal-based agent, it will slow down. Thus the only action that will achieve the goal of not hitting other cars is to brake.

Problem of goal-based agent

- Although the goal agent appears less efficient, it is more flexible because the knowledge that supports its decisions is represented explicitly and can be modified.
- Example: *if it starts to rain, the agent can update its knowledge of how effectively its brakes will operate; this will automatically cause all of the relevant behaviors to be altered to suit the new conditions*. For the reflex agent, we would have to rewrite many condition-action rules.

Utility-based agents

- A more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent. Because “happy” does not sound very scientific, economists and computer scientists use the term **utility** instead.
- example: *many action sequences will get the taxi to its destination but some are quicker, safer, reliable, or cheaper than others.*
- An agent’s utility function is essentially an internalization of the performance measure. If the internal utility function and external performance measure are in agreement, then an agent that choose actions to maximize its utility will be rational according to the external performance measure.

Utility-based agents

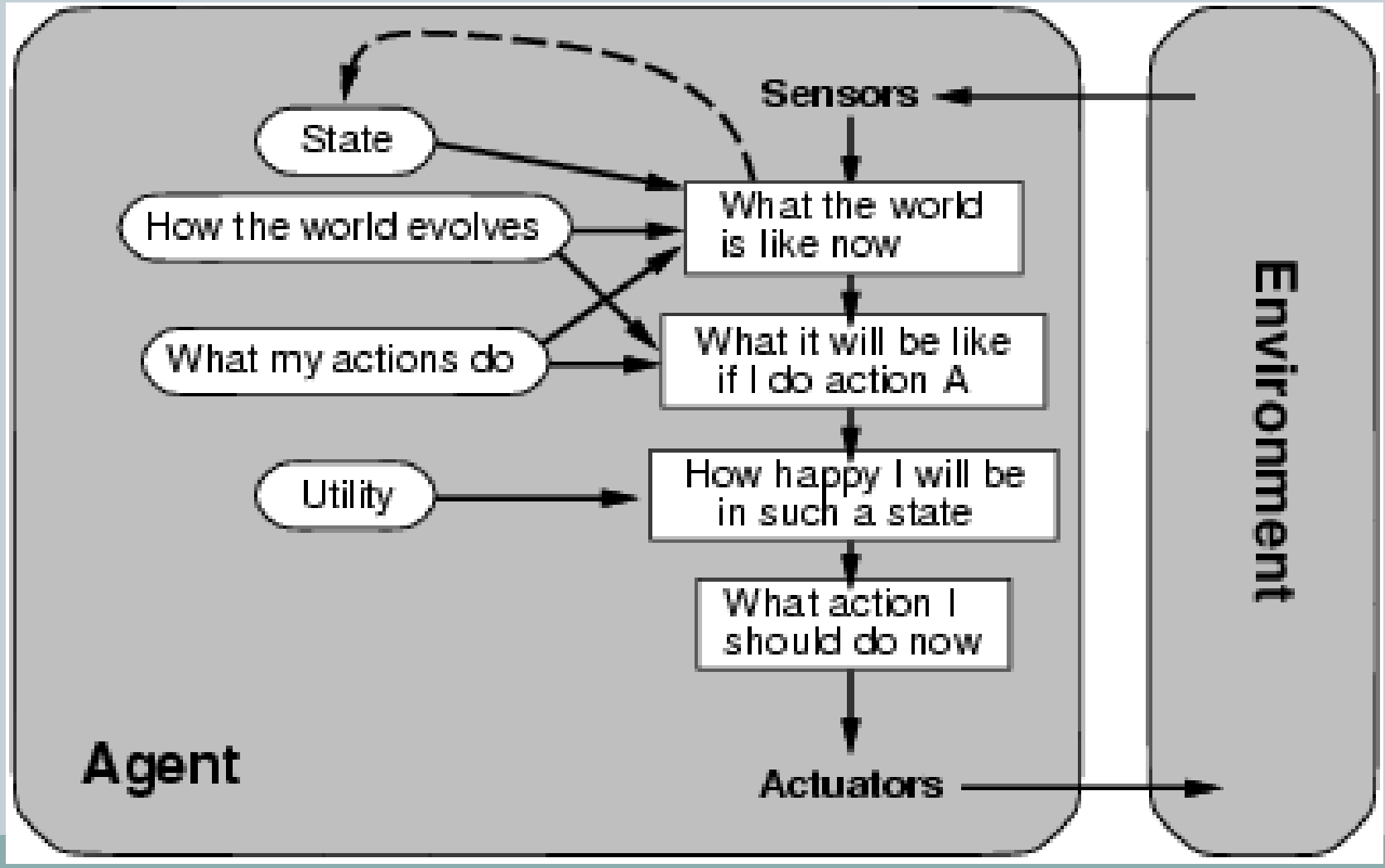
- A utility-based agent can make rational decisions.

first: when there are conflicting goals, only some of which can be achieved, the utility function specifies the appropriate tradeoff.

second: when there are several goals that the agent can aim for, none of which can be achieved with certainty, utility provides a way in which the likelihood of success can be weighed against the importance of the goals.

The utility-based agent has to model and keep track of its environment, tasks that have involved a great deal of research on perception, representation, reasoning, and learning.

Utility-based agents



Learning agents

- All agents can improve their performance through learning.
- Learning allows the agent to operate in initially unknown environments and to become more competent than its initial knowledge alone might allow.

Learning agents

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.

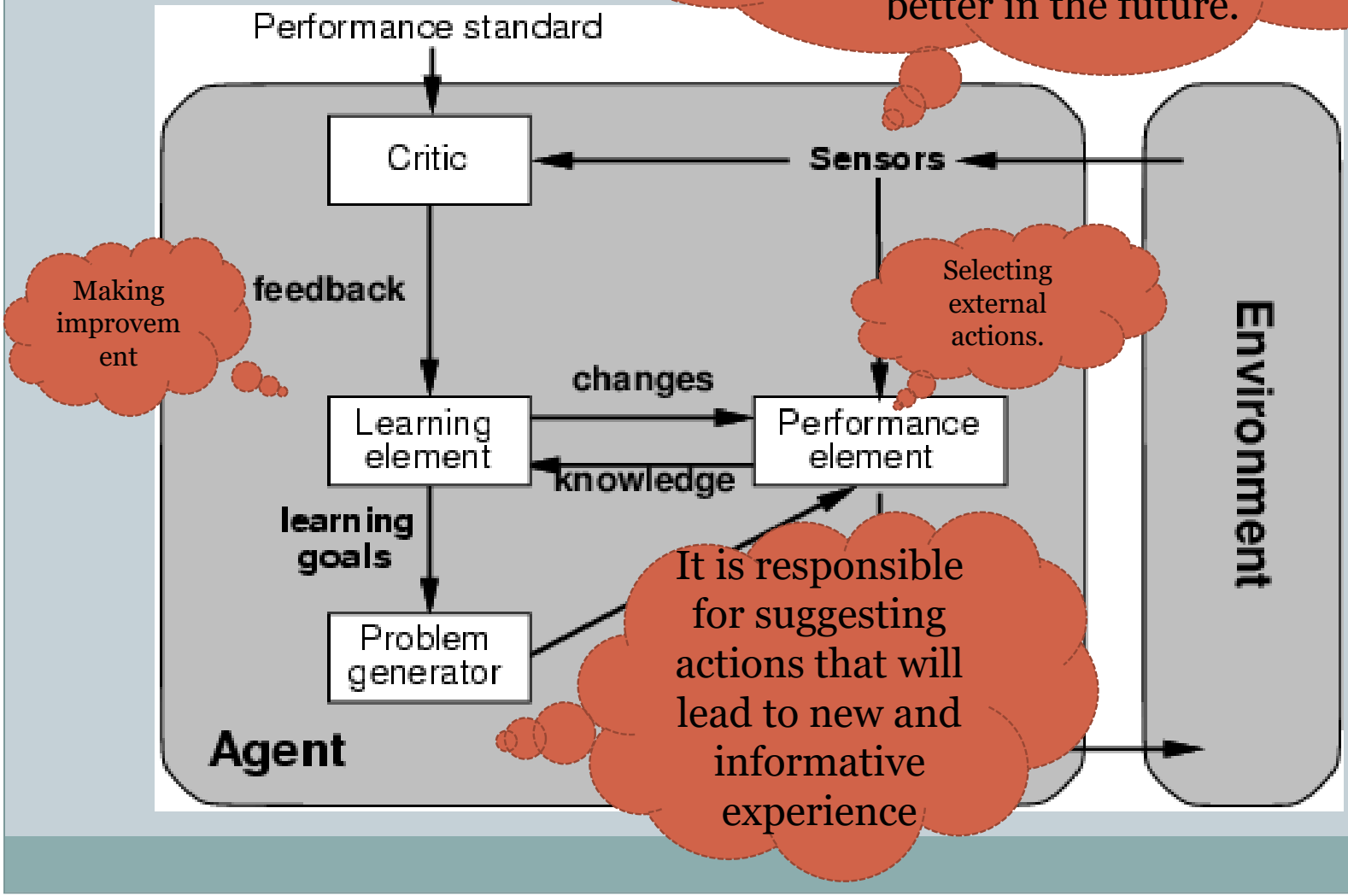


EXHIBIT D.1 Software Agents vs. Traditional Software Programs

Characteristics	Regular Software	Agents
Nature	Static	Dynamic
Manipulation	Direct: User initiates every action	Indirect: Autonomous
Interactivity	Noninteractive	Dialogues are fully interactive. Actions may be initiated by either the user or the agent system. Interacts with user and with other agents.
Flexibility	Never changes, unless changed by a human or an error in the program	Adapts, learns
Temporal continuity	Runs one time, then stops to be run again when called	Persistent: Continues to run over time
Response	Predictable: Does what you tell it to, even if you didn't mean what you said	Interprets what you mean, not what you say. In the best of circumstances, actions are based on rules, but they may change over time or in reaction to different circumstances.
Autonomy, independence	Follows instructions	May initiate actions, as well as respond to instructions
Mobility	Stays in one place	May be mobile, traveling to other servers
Concurrency	Generates process in one dedicated server with limited processing power	Dispatches simultaneously to accomplish various parts of a task in parallel
Local interaction	Accesses data across network using client-server architecture	Can travel and interact with local entities, such as databases, file servers and stationary agent, through message passing

Source: Based on Feldman and Yu (1999).

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 definie task environnements

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.