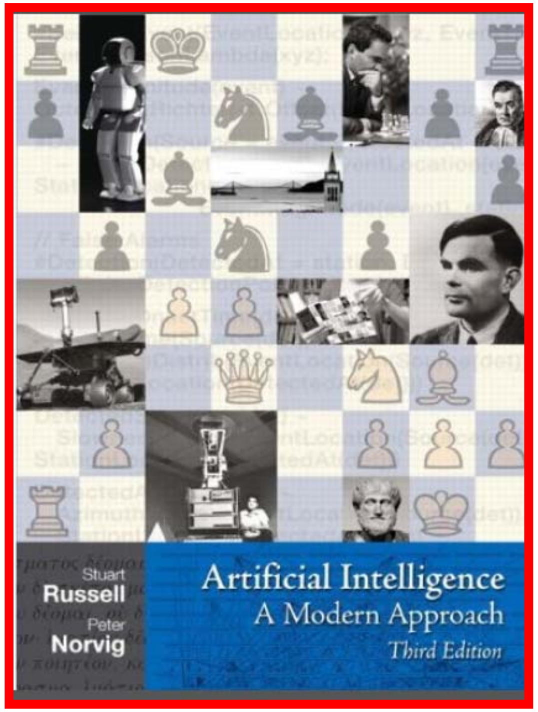


(Chapter-2)
Intelligent Agents

Yanmei Zheng

CHAPTER OUTLINE

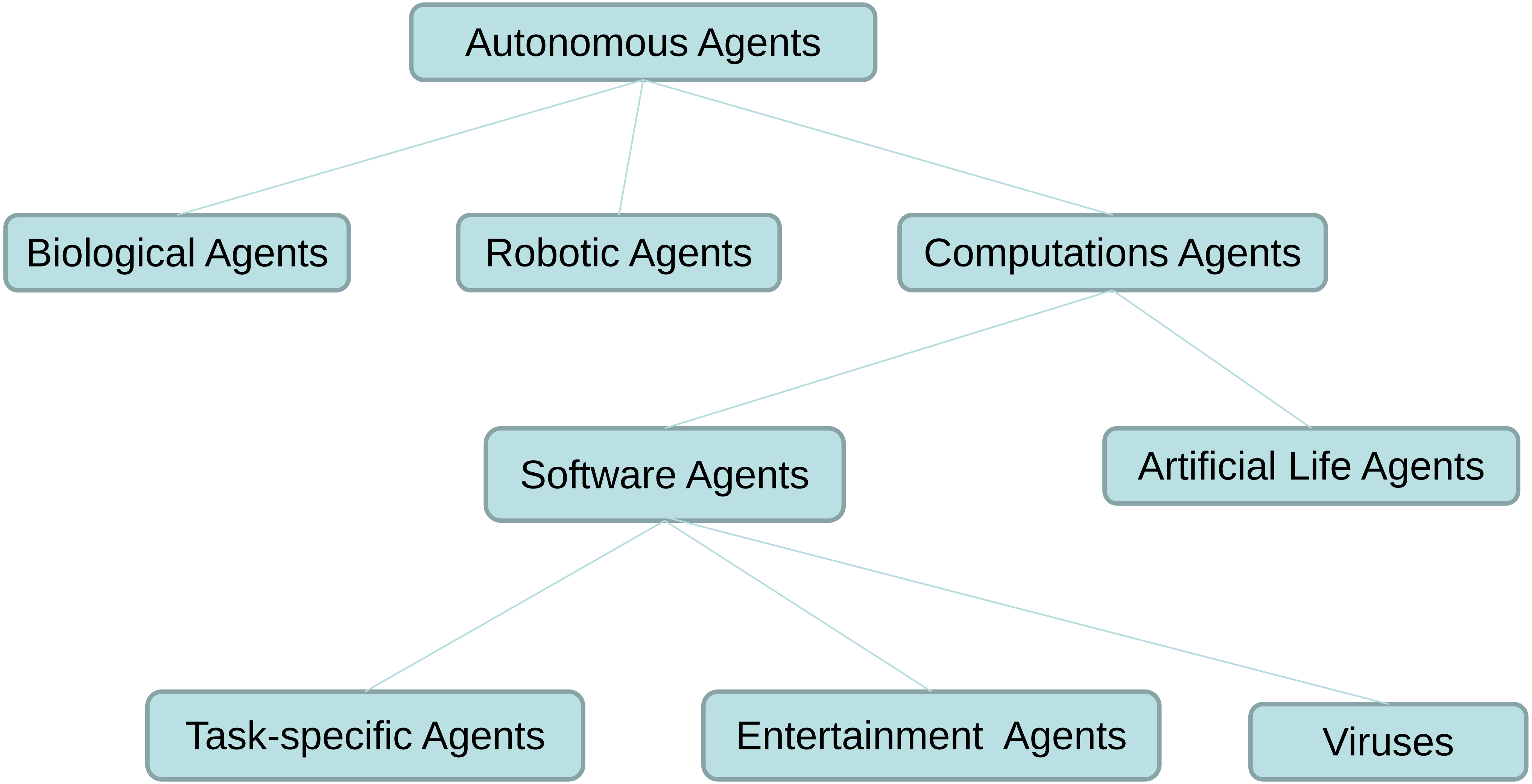
- ✓ What is an Agent?
- ✓ What is a rational agent?
- ✓ PEAS (Performance measure, Environment, Actuators, Sensors)
- ✓ Agent Task environments
- ✓ Different classes of agents

What is an Agent?

Intelligent Agent

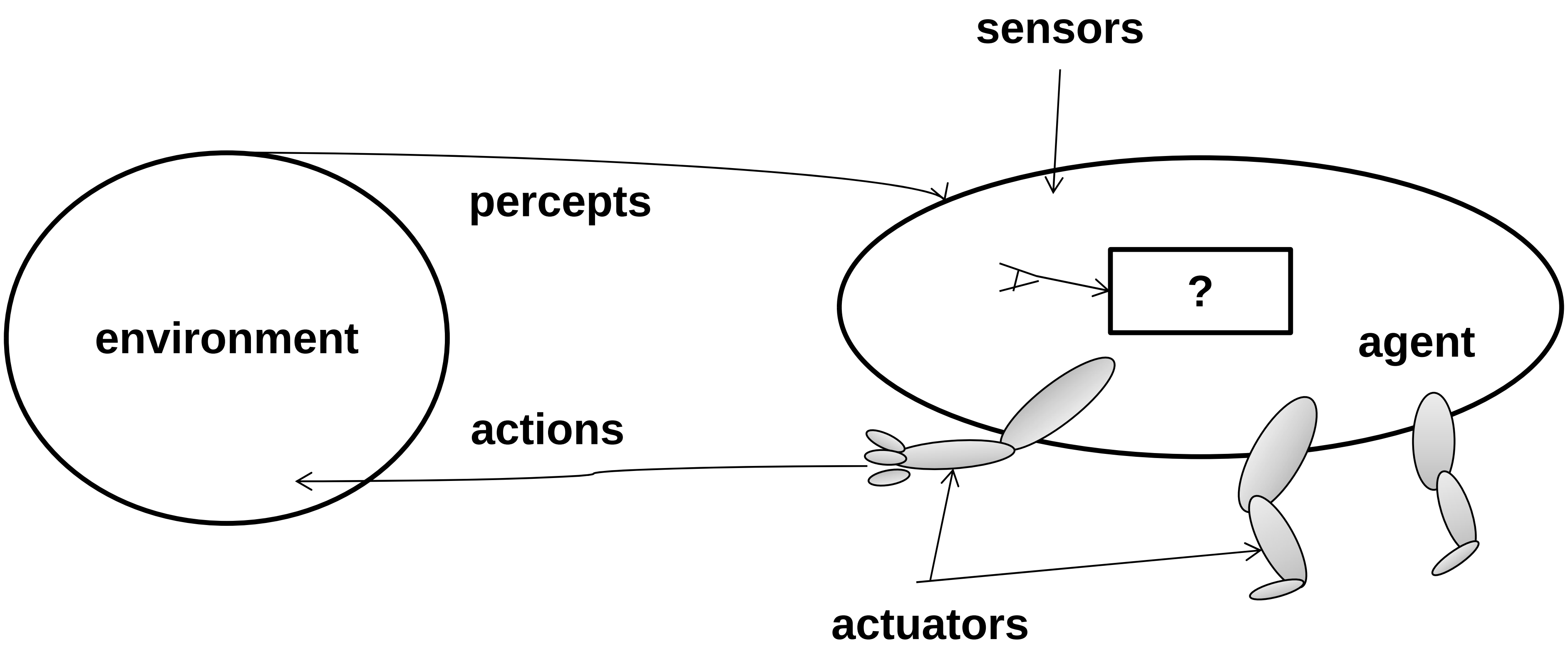
Intelligent Agent--important in this lecture.

Different Agent



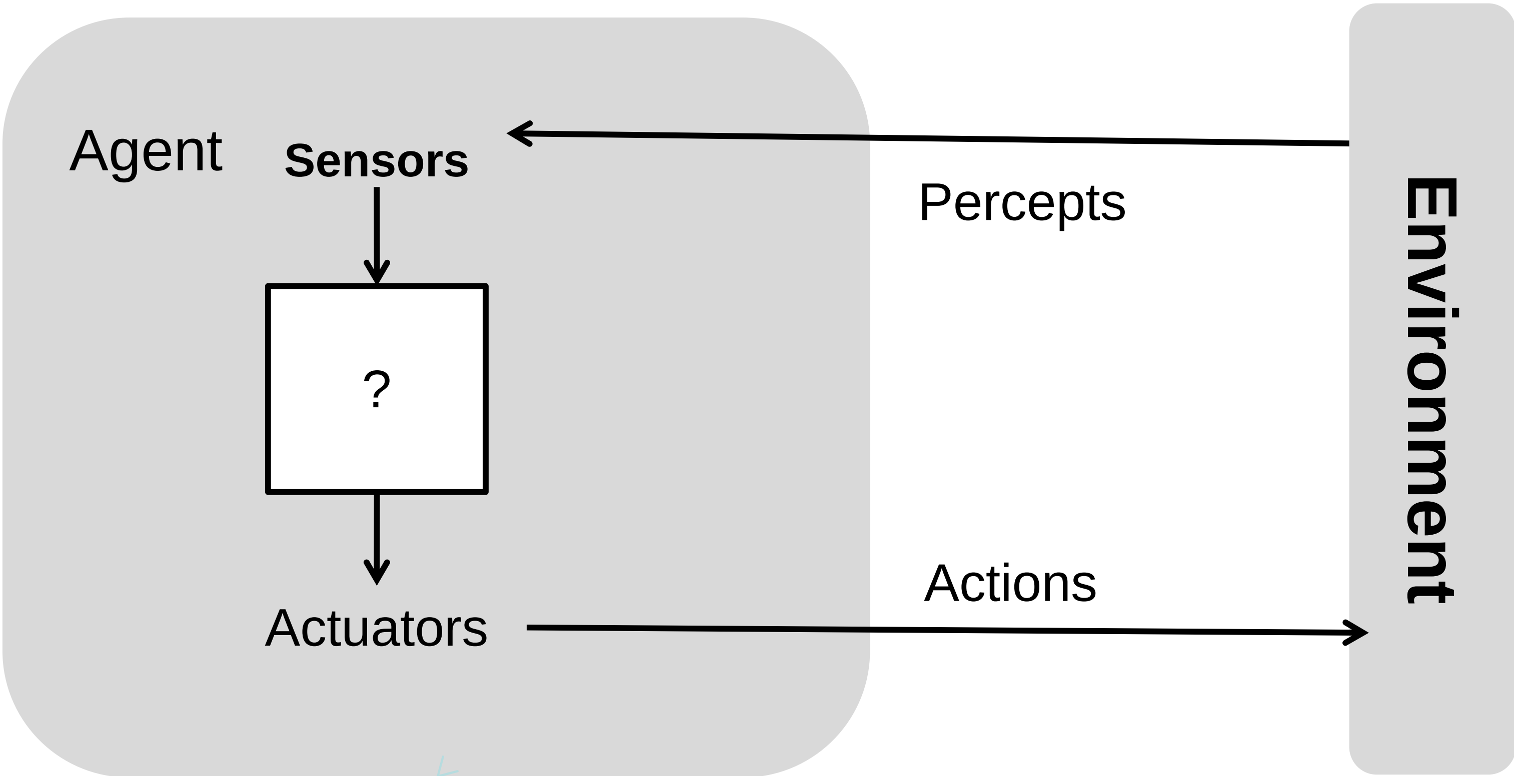
What is an Agent?

- Perceive the environment through sensors (→Percepts)
- Act upon the environment through actuators (→Actions)

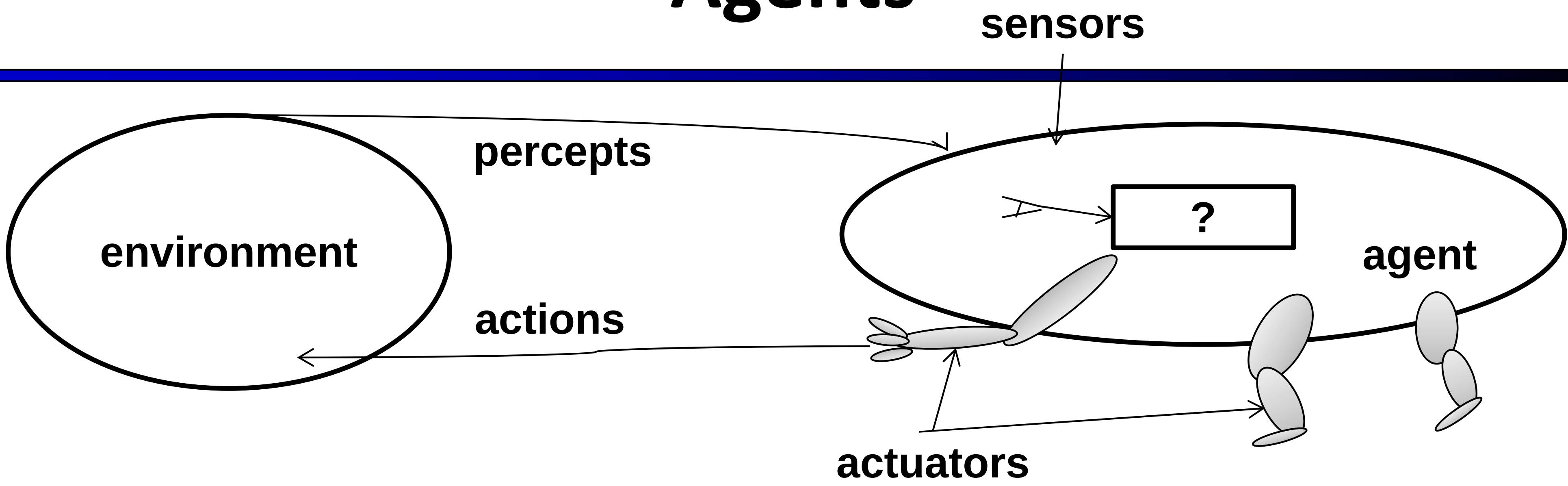


Agents

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators



Agents

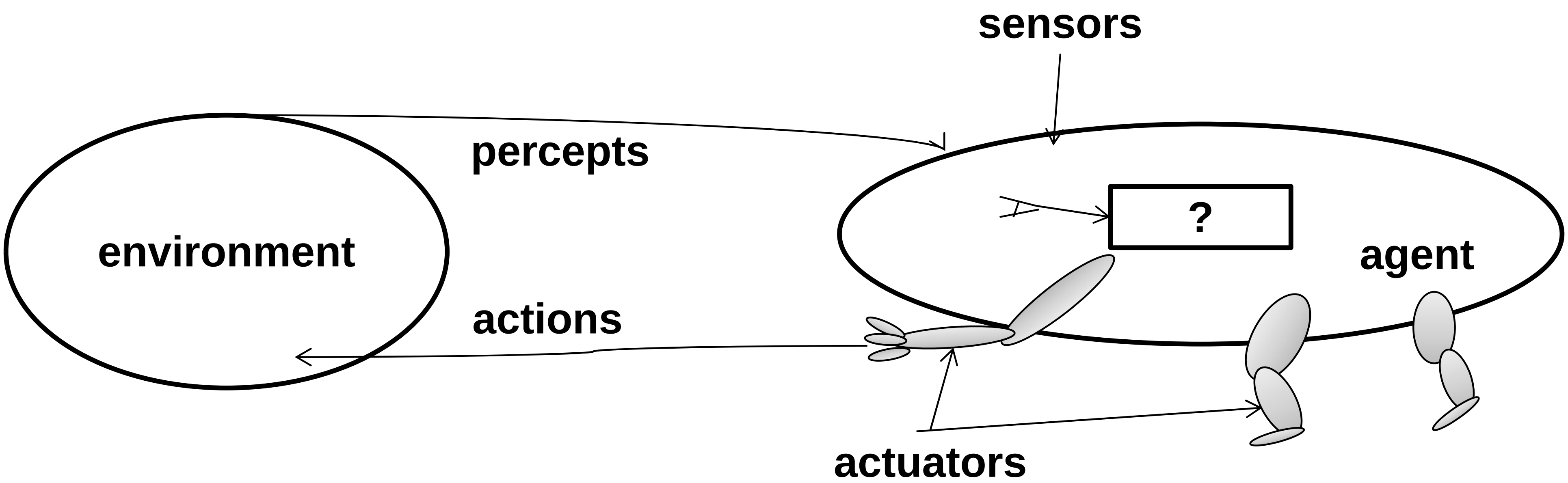


- **Human agent:**
  - sensors = eyes, ears and other organs
  - actuators= hands, legs, mouth and other body parts

Robotic agent:

- sensors = cameras and infrared range finders
- actuators= various motors for actuators

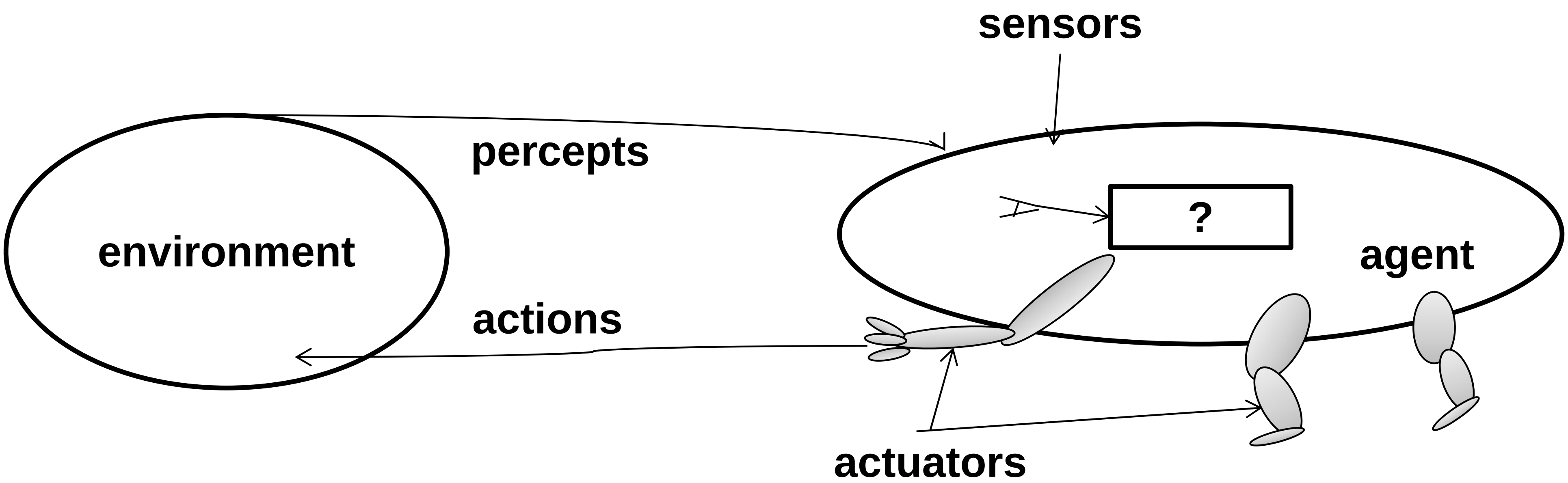
Agents



- We use the term <u>percept</u> to refer to the agent 's perceptual inputs at any given instant

- An agent <u>percept sequence</u> is the complete history of every thing the agent has ever perceived.

Agents and environments



- The **agent function** maps from percept histories to actions:

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

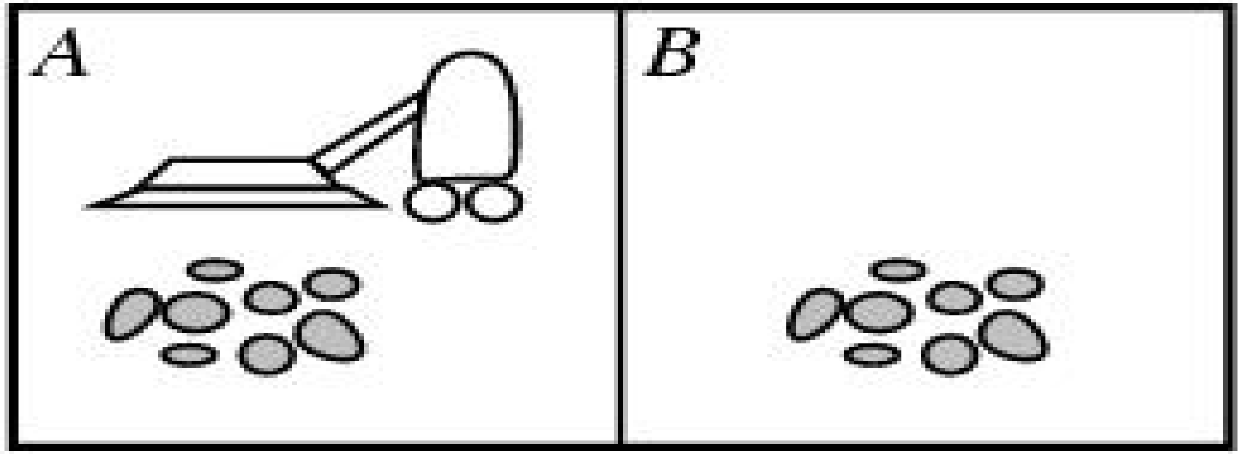
- The **agent program** runs on the physical **architecture** to produce *f*

Agent = architecture + program

AUTONOMOUS AGENT

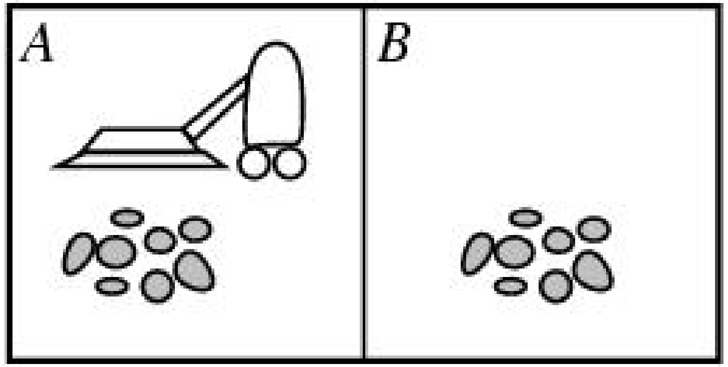
- One whose actions are based on both built-in knowledge and own experience
- Initial knowledge provides an ability to learn
- A truly autonomous agent can adapt to a wide variety of environments

The vacuum-cleaner world (1)



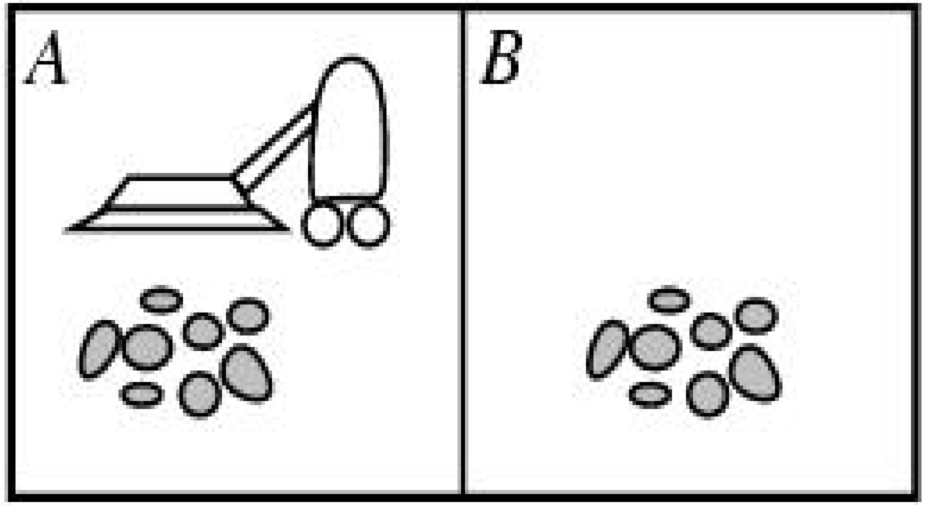
- **Environment**: square A and B
- **Percepts**: [location and content] e.g. *[A, Dirty]*
- **Actions**: left, right, suck, and no-op
- **Agent's function** → look-up table

The vacuum-cleaner world (2)



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

The vacuum-cleaner world (3)



function REFLEX-VACUUM-AGENT ([*location, status*])
return an action
if *status == Dirty* then return *Suck*
else if *location == A* then return *Right*
else if *location == B* then return *Left*

Rational agent?

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.**
- ***how and when to evaluate the agent's success.??***
- **In order to evaluate their performance, we have to define a performance measure.**

Rational agents

example : Autonomous vacuum cleaner

- **E.g., performance measure**
  - **Amount of dirt cleaned up**
  - **Amount of time taken**
  - **Amount of electricity consumed(Energy usage)**
  - **Amount of noise generated(Noise level)**
  - **Level of cleanliness**
  - **Safety (behavior towards hamsters/small children)**

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.**

Rational behavior is dependent on

1. **Performance measures (goals)**
2. **Percept sequences up to date**
3. **The agent prior Knowledge of the environment**
4. **Possible actions**

Rational agents

- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt)

Rational agents

Optimal behavior is often <u>unattainable</u> (not totally achieved)

- **Not all relevant information is perceivable**
- **Complexity of the problem is too high**

- **<u>Active</u> perception is necessary to avoid trivialization.**
- **The ideal rational agent acts according to the function**

Percept Sequence X World Knowledge → Action

Structure of Rational Agents

- **Agent program**, executed on an
- **Architecture** which also provides an interface to the environment (percepts, actions)

Agent = architecture + program

Rationality vs. Omniscience

- An omniscient agent knows the actual effects of its actions and it is impossible in real world

- In comparison, a rational agent behaves according to its percepts and knowledge and attempts to maximize the expected performance

- Example: If I look both ways before crossing the street, and then as I cross I am hit by a car, I can hardly be accused of lacking rationality.

Omniscient =have all the knowledge

Agent Task environments

Task Environment

- **Before we design an intelligent agent, we must specify its "task environment":**

PEAS:

Performance measure

Environment

Actuators

Sensors

Task Environment (cont..)

- 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, sonar, speedometer, GPS, odometer, engine sensors, keyboard

Task Environment (cont..)

- **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, patient's answers, examination reports)

Task Environment (cont..)

- **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

Task Environment (cont..)

- Agent: Interactive English tutor
  - Performance measure: Maximize student's score on test
  - Environment: Set of students
  - Actuators: Screen display, Speaker (exercises, suggestions, corrections)
  - Sensors: Keyboard

Examples of Rational Agents

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	healthy patient, costs, lawsuits(court cases)	patient, hospital, stuff	display questions, tests, diagnoses, treatments,	keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	correct image categorization	downlink from orbiting satellite	display categorization of scene	color pixel arrays
Part-picking robot	percentage of parts in correct bins(books)	conveyor belt with parts, bins	jointed arm and hand	camera, joint angle sensors

Examples of Rational Agents

Agent Type	Performance Measure	Environment	Actuators	Sensors
Refinery controller	purity, safety	Refinery, operators	valves pumps, heaters displays	temperature, pressure, chemical sensors
Interactive English tutor	student's score on test	set of students, testing agency	display exercises, suggestions, corrections	keyboard entry
web crawling agent	did you get only pages you wanted	User, internet	Display related info	keyboard entry

Properties of Task Environment or Types

- **Fully observable vs. partially observable**
- **Deterministic vs. stochastic**
- **Episodic vs. sequential**
- **Static vs. dynamic**
- **Discrete vs. continuous**
- **Single agent vs. multi agent**

Fully observable vs. partially observable

- **fully observable :** if an agent's sensors give it access to the complete state of the environment at each point in time.
- agent need not maintain any internal state to keep track of the world.
- An environment might be **partially observable** because of
  - noisy and inaccurate sensors
  - or because parts of the state are simply missing from the sensor data
  - Examples: vacuum cleaner with local dirt sensor.

Deterministic vs. stochastic

- deterministic Environment : if the next state of the environment is completely determined by the current state and the action executed by the agent.

- EX: Vacuum cleaner is Deterministic why?
- Ex: Taxi driving agent (robot driving agent) is stochastic, why?
  - He doesn’t know about traffic, can never predict traffic situation

Stochastic= connected to random events

Episodic vs. sequential:

- Episodic An agent's action is divided into atomic episodes. Each episodic perceive then take action(this action depend on this episodes) and next episodic does not rely on previous one it taking the right action.

  EX: classification tasks,

- Sequential: the current decision could affect all future decisions

  EX: chess and taxi driver

Static vs. dynamic:

- **Static environment** is unchanged while an agent is deliberating
  - it is easy to deal with because the agent need not keep looking at the world while it is deciding on the action or need it worry about the passage of time
  - EX :crossword puzzles are static
- **Dynamic environments:** continuously ask the agent what it wants to do

  Ex: taxi driving is dynamic

Discrete vs. Continuous:

- **Discrete : A limited number of distinct, clearly defined states, percepts and actions.**
  - **Ex: Chess has finite number of discrete states, and has discrete set of percepts and actions.**
- **Continuous : not limited**
  - **Taxi driving has continuous states, and actions**

Single agent vs. multi-agent:

- **An agent operating by itself in an environment is single agent**
  - **EX: Crossword is a single agent**
  - **Ex: chess is a competitive multi-agent environment**

Environment types

- The simplest environment is
  - Fully observable, deterministic, episodic, static, discrete and single-agent.
- Most real situations are:
  - Partially observable, stochastic, sequential, dynamic, continuous and multi-agent.

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

Agent functions and programs

- An agent is completely specified by the agent function mapping percept sequences to actions
- One agent function (or a small equivalence class) should be rational
- Aim: find a way to implement the rational agent function concisely

Different classes of agents

1. Simple Reflex agents
2. Model based Reflex agents
3. Goal Based agents
4. Utility Based agents

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 $\leftarrow$ LookUp(percepts,table)

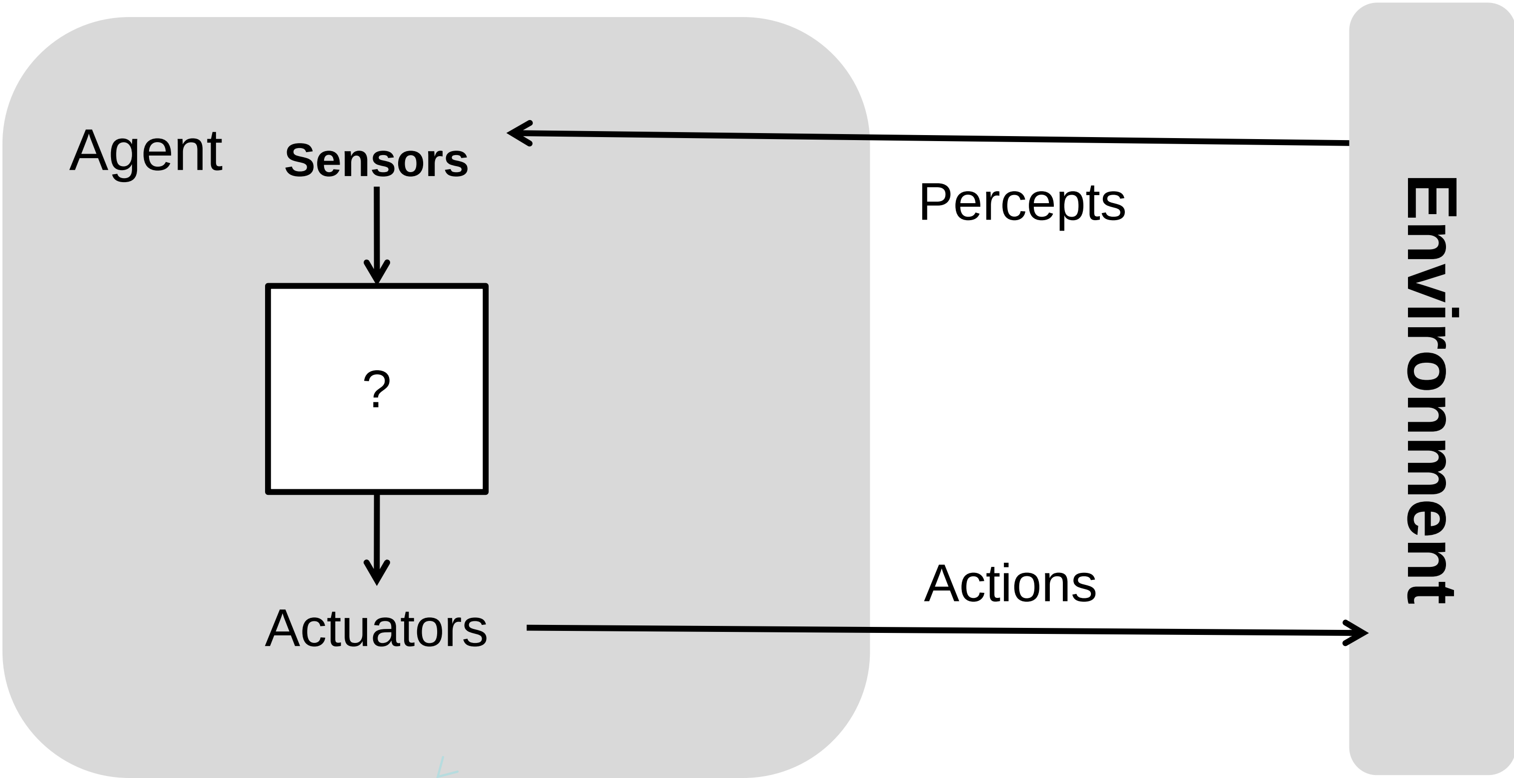
return action

Table-lookup agent

- \input{algorithms/table-agent-algorithm}

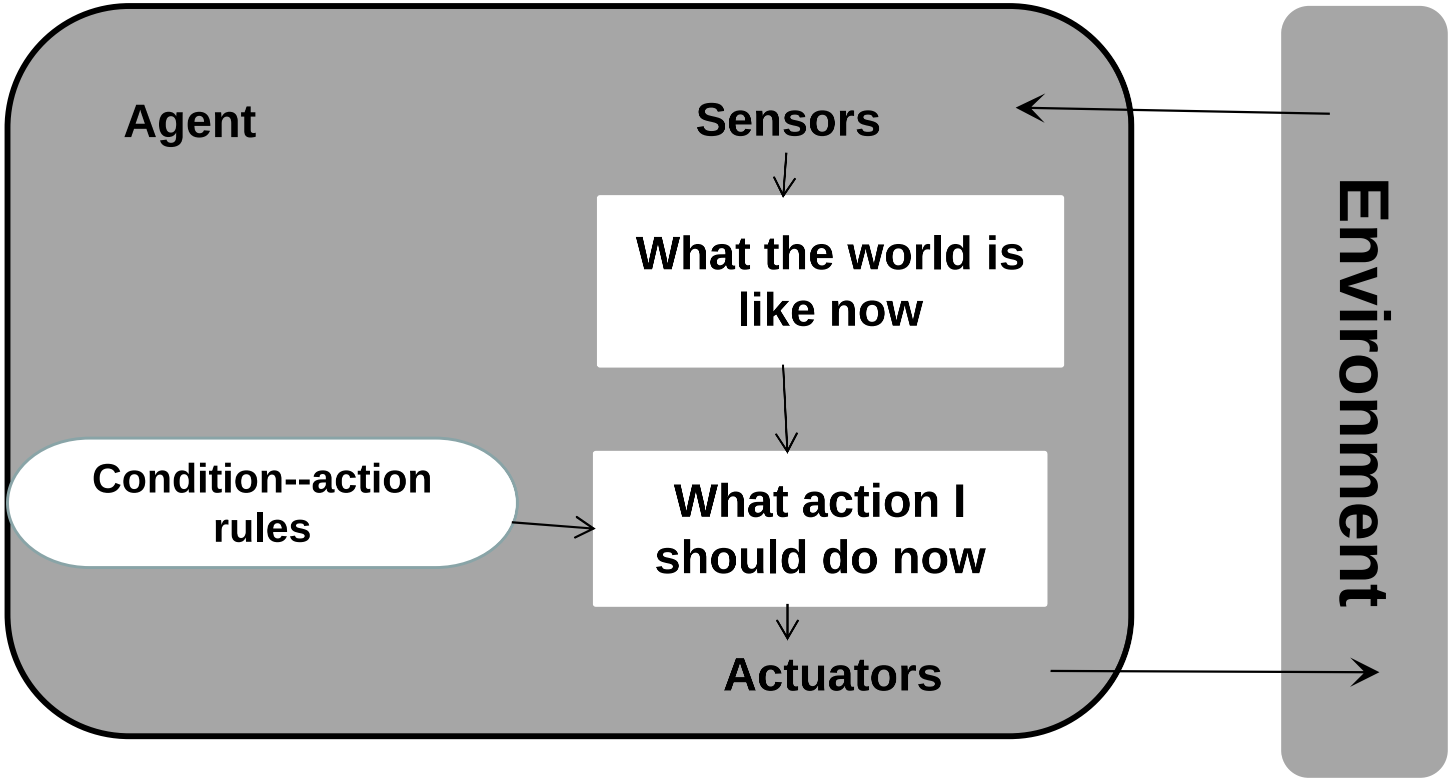
- Drawbacks:
  - Huge table
  - Take a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries

Structure of agents



Simple reflex agents

Rule of condition—action： **if** status = Dirty **then return** Suck



$f: P \rightarrow A$ $\quad\quad$ f : IF-THEN

Simple Reflex Agent

Two-state vacuum environment Agent

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

- ## A simple Agent

function SIMPLE-REFLEX-AGENT(percept) **returns** an action

static: rules,a set of condiction-action rules

state ← INTETPRET-INPUT(percept)

rule ← RULE-MATCH(state,rules)

action ← RULE-ACTION [rule]

return action

Simple Reflex Agent

- Reflex agents respond **immediately** to percepts.

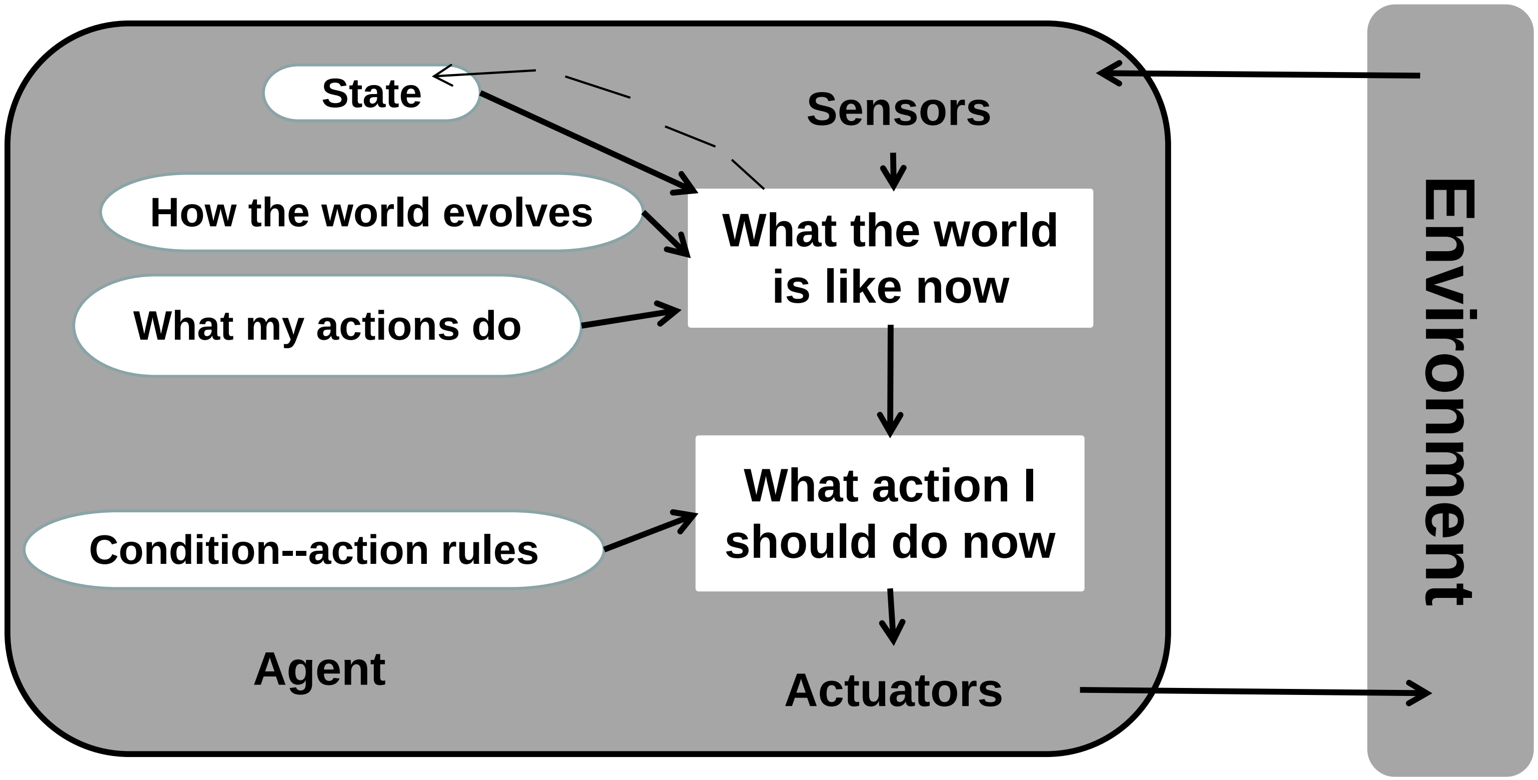
- **Select actions on the basis of the current percept, ignoring the rest of the percept history**

- **Ex vacuum cleaner , why?**

  Because its decision based only on the current location and whether it contain dirt or not.

Model-based reflex agents

A Model of the world=How the world works + My action



f : P+M→A **f : IF⁺-THEN**

Model-based reflex agents

function REFLEX-AGENT-WITH-STATE(percept) **returns** an action

static: state, a description of the current world state

rules,a set of condiction-action rules

action,the most recent action,initially none

state←UPDATE-STATE(state,action,percept)

rule←RULE-MATCH(state,rules)

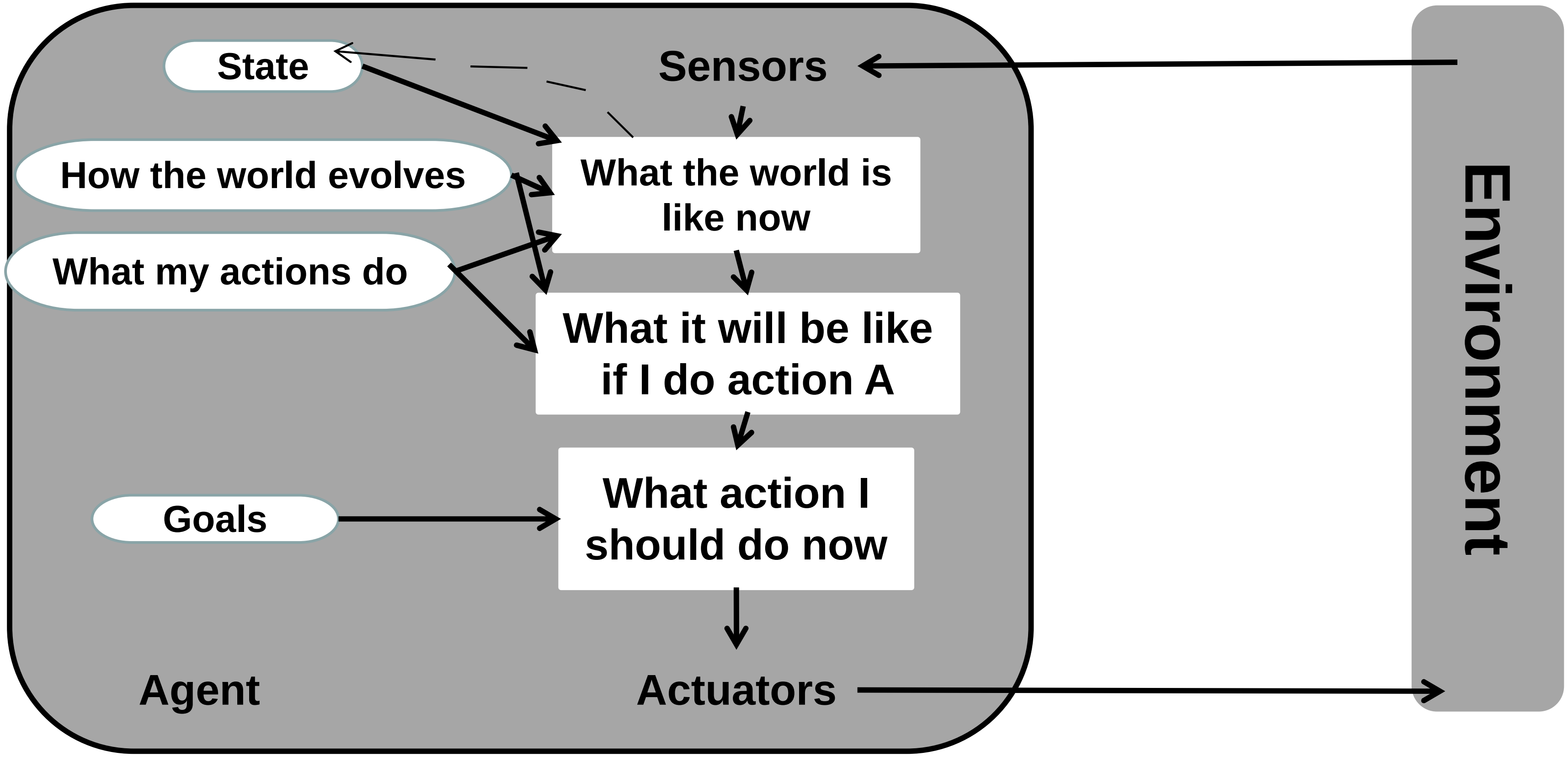
action←RULE-ACTION[rule]

return action

- The most effective way to handle **partial observably**
- In case the **agent's history in perception** in addition to the actual percept is required to **decide on the next action**, it must be represented in a suitable form.

Goal-based agents

+ A set of goals it is trying to achieve



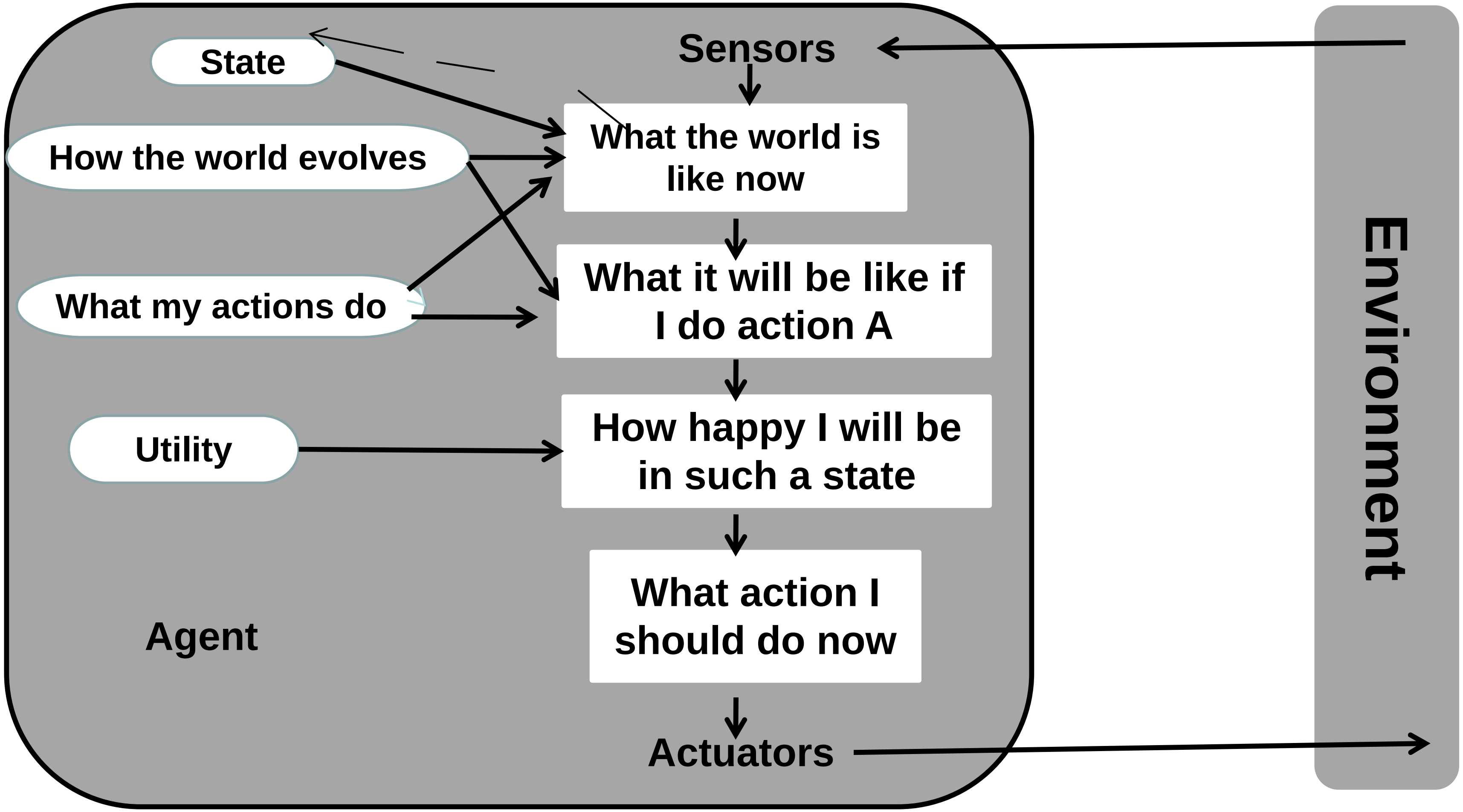
$f : P{+}M{+}Try \rightarrow A$ $\qquad$ f : Target-Try

Goal-based agents

- **Goal-based agents** work towards goals.
- **Often, percepts alone are insufficient to decide what to do.**
- **This is because the correct action depends on the given explicit goals (e.g., go towards X).**
- **The goal-based agents use an explicit representation of goals and consider them for the choice of actions.**
- **Ex : taxi driving destination , vacuum cleaner**

Utility-based agents

+utility function， choose actions to maximize its utility



f* : *P+M+Try+Utility→A　　　　***f* : Utility**

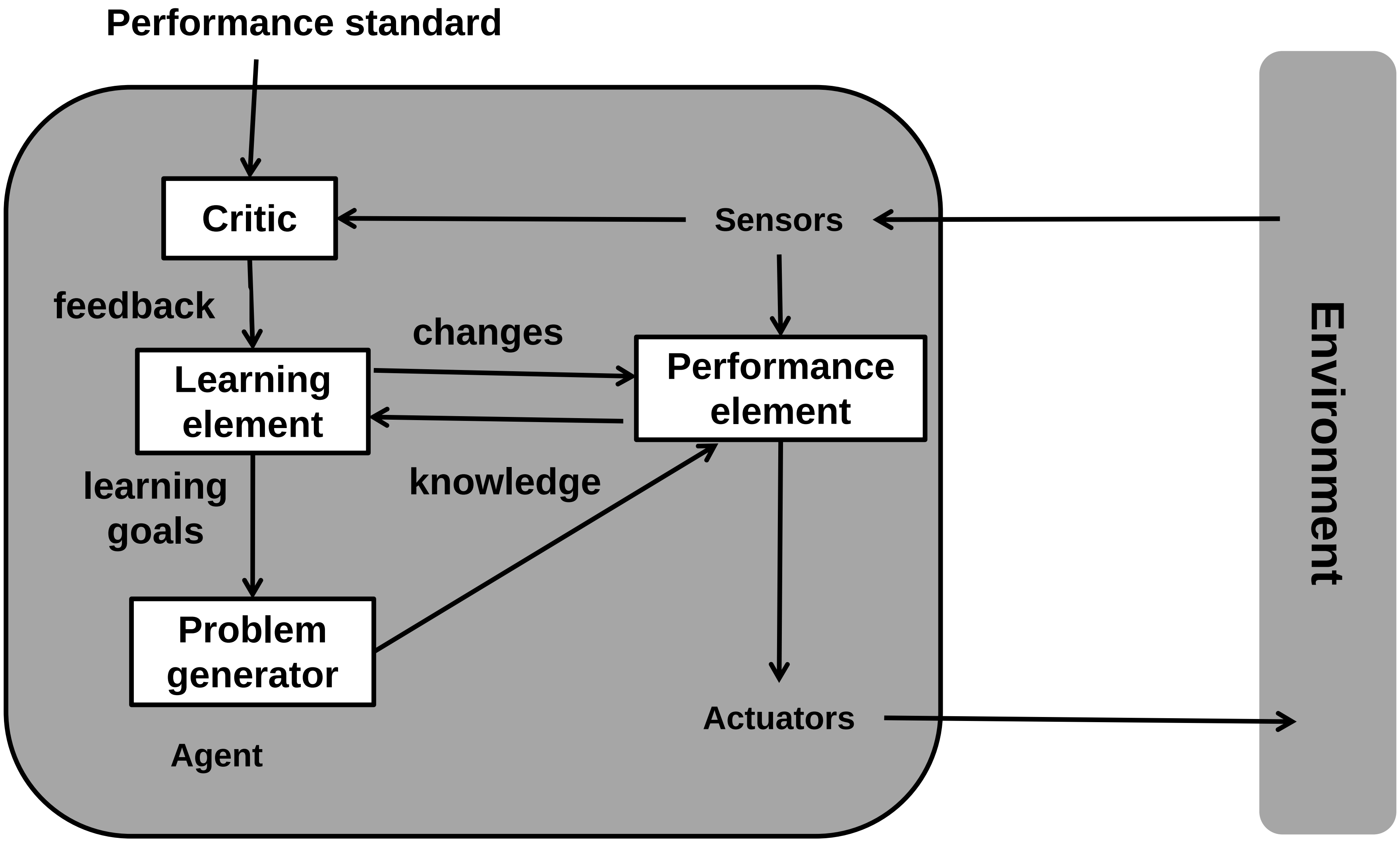
Utility-based agents

- Usually, there are several possible actions that can be taken in a given situation.
- Utility-based agents take action that maximize their reward.
- A utility function maps a state (or a sequence of states) onto a real number. The agent can also use these numbers to weigh the importance of competing goals.
- Ex taxi driving , may be many paths lead to goal but some are quicker, cheaper, safer

Learning Agents

- **Learning agents** improve their behavior over time
- **Learning agents can become more competent over time.**
- **They can start with an initially empty knowledge base.**
- **They can operate in initially unknown environments.**

Learning agents

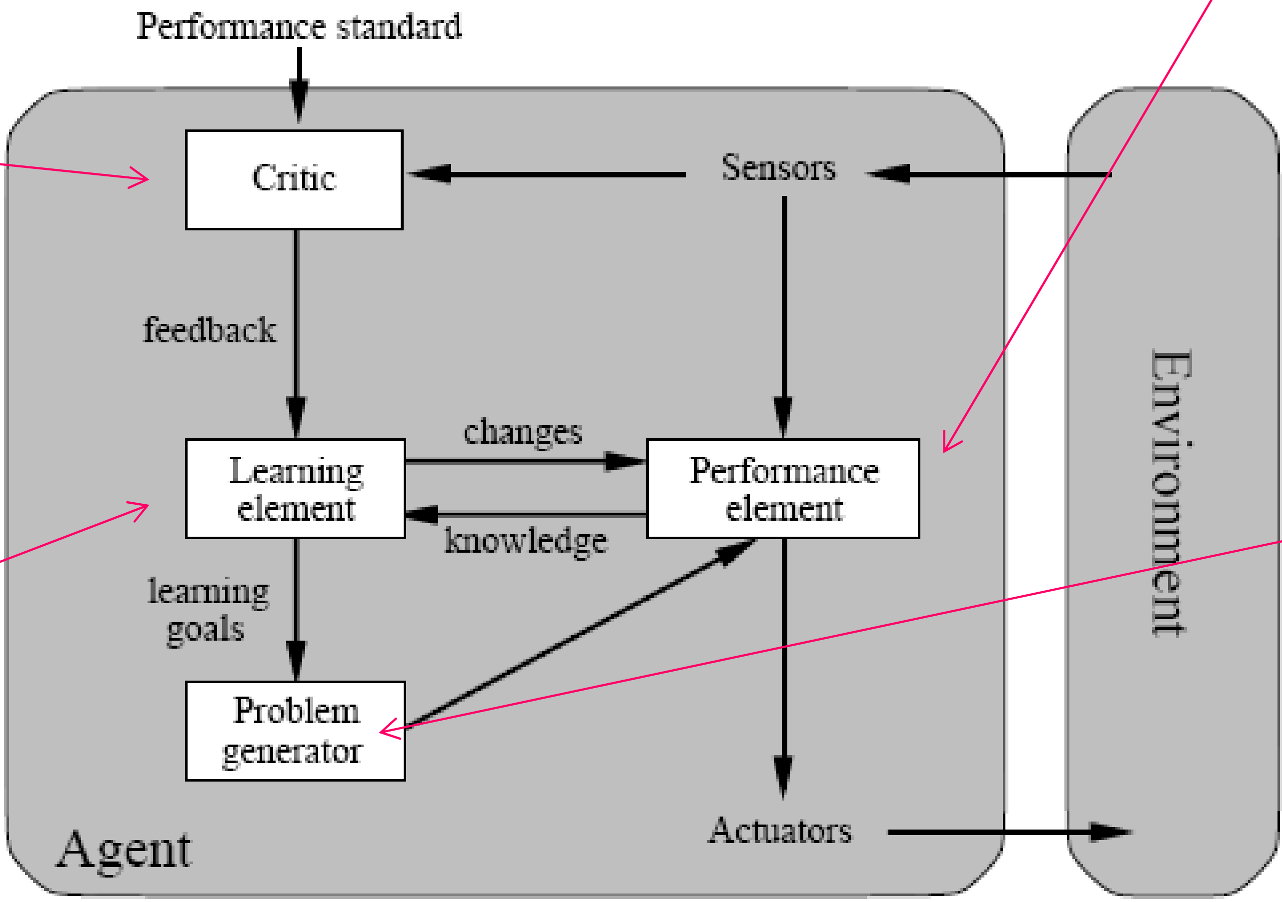


Learning Agents

determines the performance of the agent:

percept only doesn't provide how much is the agent is successful

responsible for making improvements



take percept and decide an action

suggests exploratory actions that will lead to new informative experiences

Thank you

End of
Chapter 2