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

- *Software agent (surfBot : software robot)* :-

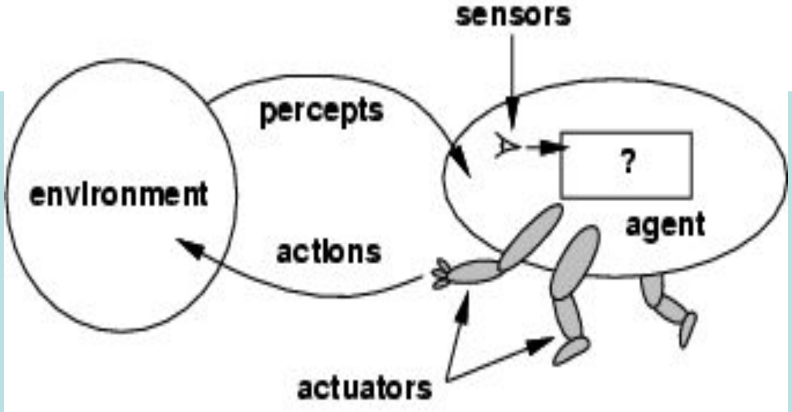
  receives keystrokes, file contents, network packets etc as sensory input.

  acts on the environment by displaying on the screen, writing files, sending message packets.

Agent perception and action

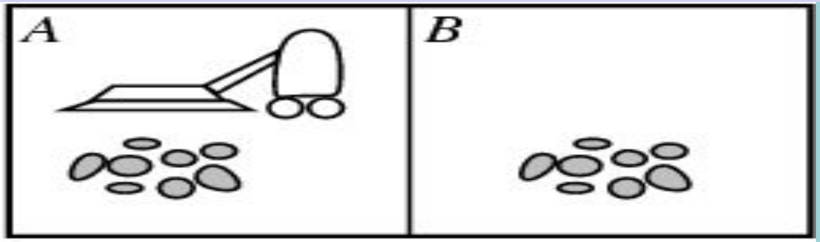
- **Percept** :- refers to an agents perceptual input at any given instant.
- **Percept Sequence** :- refers to the complete history of what an agent has perceived so far.
- **Action**:- an agent’s choice of action at any given instant can depend on the entire percept sequence observed till that time.
- **Agent function**:- an agent’s behavior is described by the agent function that maps any percept sequence to an action.

Agents and Environments



- The agent function maps from percept histories to actions: [f: $\mathcal{P}^{\star}$ □ $\mathcal{A}$]
- The agent function is implemented by an agent program.
- 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*

Percept Sequence : **Actions** :

[A, Clean]	right
[A, Dirty]	suck
[B, Clean]	left
[B, dirty]	suck
[A, Dirty],[A, Clean]	right

……………

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:
  - the amount of dirt cleaned up,
  - the amount of time taken,
  - the amount of electricity consumed,
  - the amount of noise generated, etc .

RATIONALITY

- **Rationality** at any given time depends on :
  - Agent's prior knowledge of the environment.
  - Actions that the agent can perform.
  - Agent's percept sequence till that moment.
  - The performance measure that defines the criterion of success.

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.

Requirements for a Rational Agent

- ***Rationality*** ; which is distinct from omniscience (all-knowing with infinite knowledge)
- ***Information Gathering Capability*** ; Agents can perform actions in order to modify future percepts so as to obtain useful information. Learning ability.
- Ability to **adapt**.
- An agent is **autonomous** if its behavior is determined by its own experience (with ability to learn and adapt)

Building Intelligent Agents

- An agent is completely specified by the **agent function** which maps percept sequences to action.
- An agent has some internal data structures that is updated as and when new percepts arrive.
- The data structures are operated on by the agent’s decision making procedures to generate an action choice, which is then passed to the architecture to get executed.

Building Intelligent Agents cont..

- An agent consists of an architecture plus a program that runs on that architecture.
- In designing intelligent systems , there are 4 main factors to consider:-
  - **Percepts**:- the inputs to our system.
  - **Actions**:- the outputs of our system.
  - **Goals**:- what the agent is expected to achieve.
  - **Environment**:- what the agent is interacting with.

PEAS

- Task environments are problems to which rational agents are the solutions. The task environment is specified by **PEAS**.
- PEAS stands for :- Performance measure, Environment, Actuators, Sensors.
- PEAS specifies the settings for an intelligent agent design.

PEAS

- 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.

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.
- Fully observable vs. Partially observable
  - If an agent's sensors give it access to the complete state of the environment at each point in time,
  - then the environment is <u>effectively and **fully observable**</u> if the sensors detect all aspects, that are relevant to the choice of action.

Environment Types contd..

- **Partially observable**

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.

Example:

- A local dirt sensor of the cleaner cannot tell Whether other squares are clean or not

Environment Types

- Deterministic vs. stochastic
  - next state of the environment Completely determined by the current state and the actions executed by the agent, then the environment is deterministic, otherwise, it is Stochastic.
  - Strategic environment: deterministic except for actions of other agents

-Cleaner and taxi driver are:

- Stochastic because of some unobservable aspects noise or unknown

Environment Types

- **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 contd..

- **Static** (vs. Dynamic): The environment is unchanged while an agent is deliberating. (The environment is **semi dynamic** 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 contd..

	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 **agent function** mapping percept sequences to actions.
- One agent function (or a small equivalence class) is rational.
- Aim: find a way to implement the Rational Agent function concisely.

Agent Types contd..

- Four basic **Types of Agents** in order of increasing sophistication:
  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents

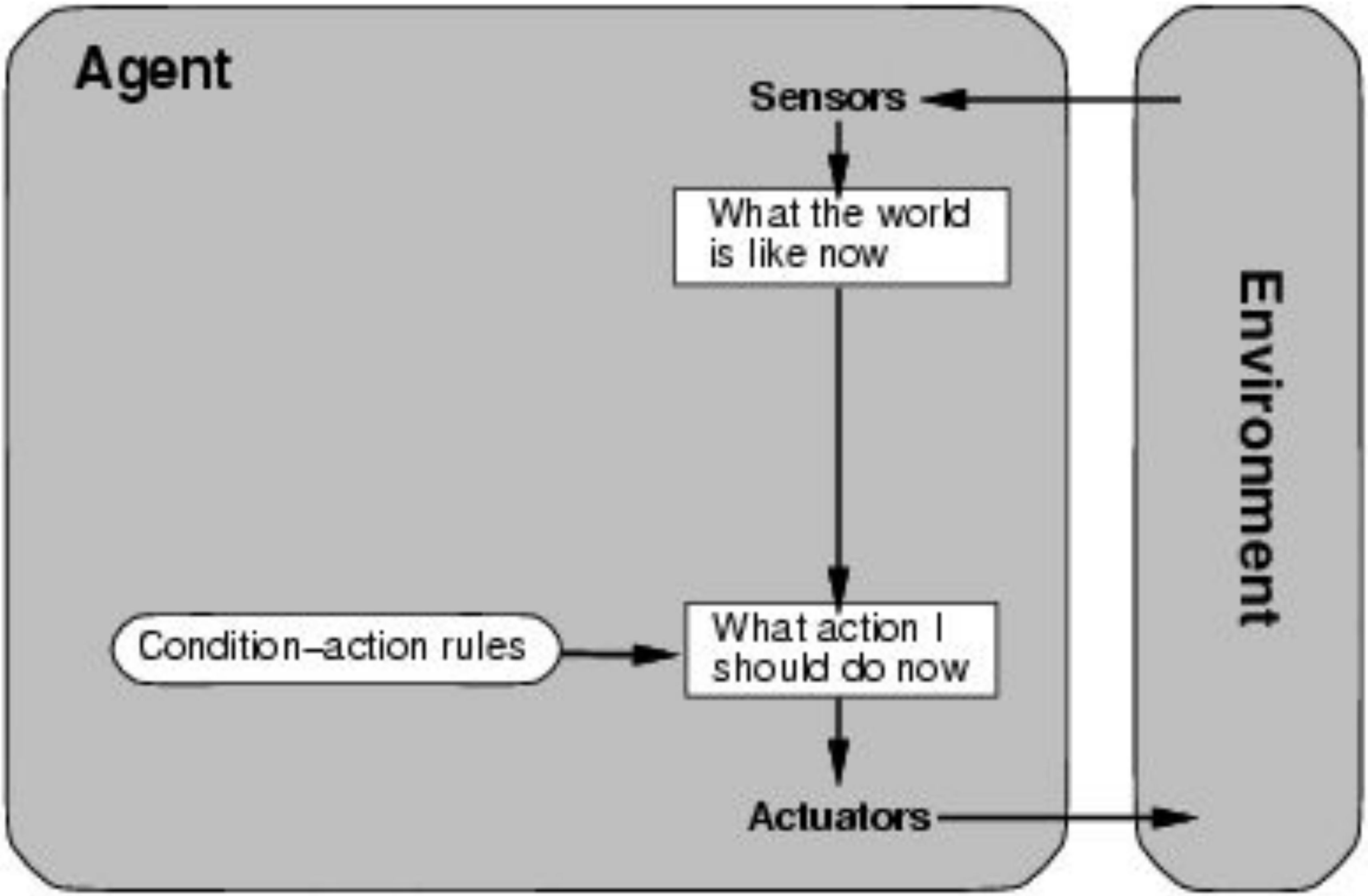
Simple Reflex Agents

- It works by finding a **rule** whose condition matches the current situation (as defined by the percept) and then doing the action associated with that rule.
- A set of condition ~ action rules / situation ~ action rules / productions / if~ then rules are defined.

  e.g if the car – in - front is braking

  then initiate – braking
- This agent function only succeeds when the environment is fully observable.

Simple Reflex Agents

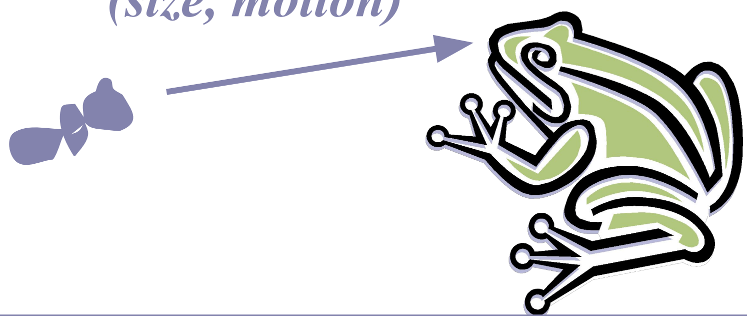


```
function SIMPLE-REFLEX-AGENT(percept) returns action
   static: rules, a set of condition-action rules

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

A Simple Reflex Agent in Nature

percepts
(size, motion)



<u>RULES:</u>

(1) If small moving object,
then activate SNAP

(2) If large moving object,
then activate AVOID and inhibit SNAP

ELSE (not moving) then NOOP

needed for completeness

Action: SNAP or AVOID or NOOP

Model-Based Reflex Agents

- **Model-based agents** can handle partially observable environments.
- Its current state is stored inside the agent maintaining some kind of structure which describes the part of the world which can't be seen.
- This behavior requires information on how the world behaves and works. This additional information completes the "world view" model.

Model-Based Reflex Agents

- A **model–based reflex agent** keeps track of the current state of the world using an internal model. It then chooses an action in the same way as the reflex agent.
- The state changes whenever an action is performed or something is perceived in the environment.

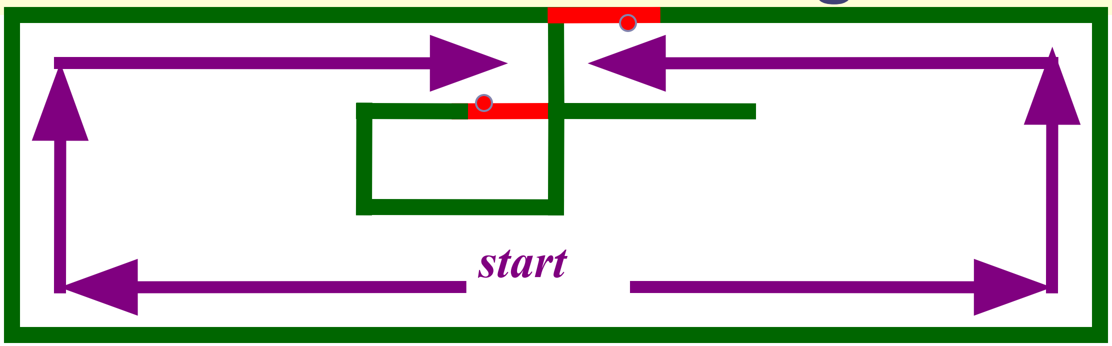
Model-Based Reflex Agents

- For the world that is partially observable
  - the agent has to keep track of an internal state
    - That depends on the percept history
    - Reflecting some of the unobserved aspects
    - E.g., driving a car and changing lane
- Requiring two types of knowledge
  - How the world evolves independently of the agent
  - How the agent's actions affect the world

Example Table Agent With Internal State

IF	THEN
Saw an object ahead, and turned right, and it's now clear ahead	Go straight
Saw an object Ahead, turned right, and object ahead again	Halt
See no objects ahead	Go straight
See an object ahead	Turn randomly

Example Reflex Agent With Internal State: **Wall-Following**

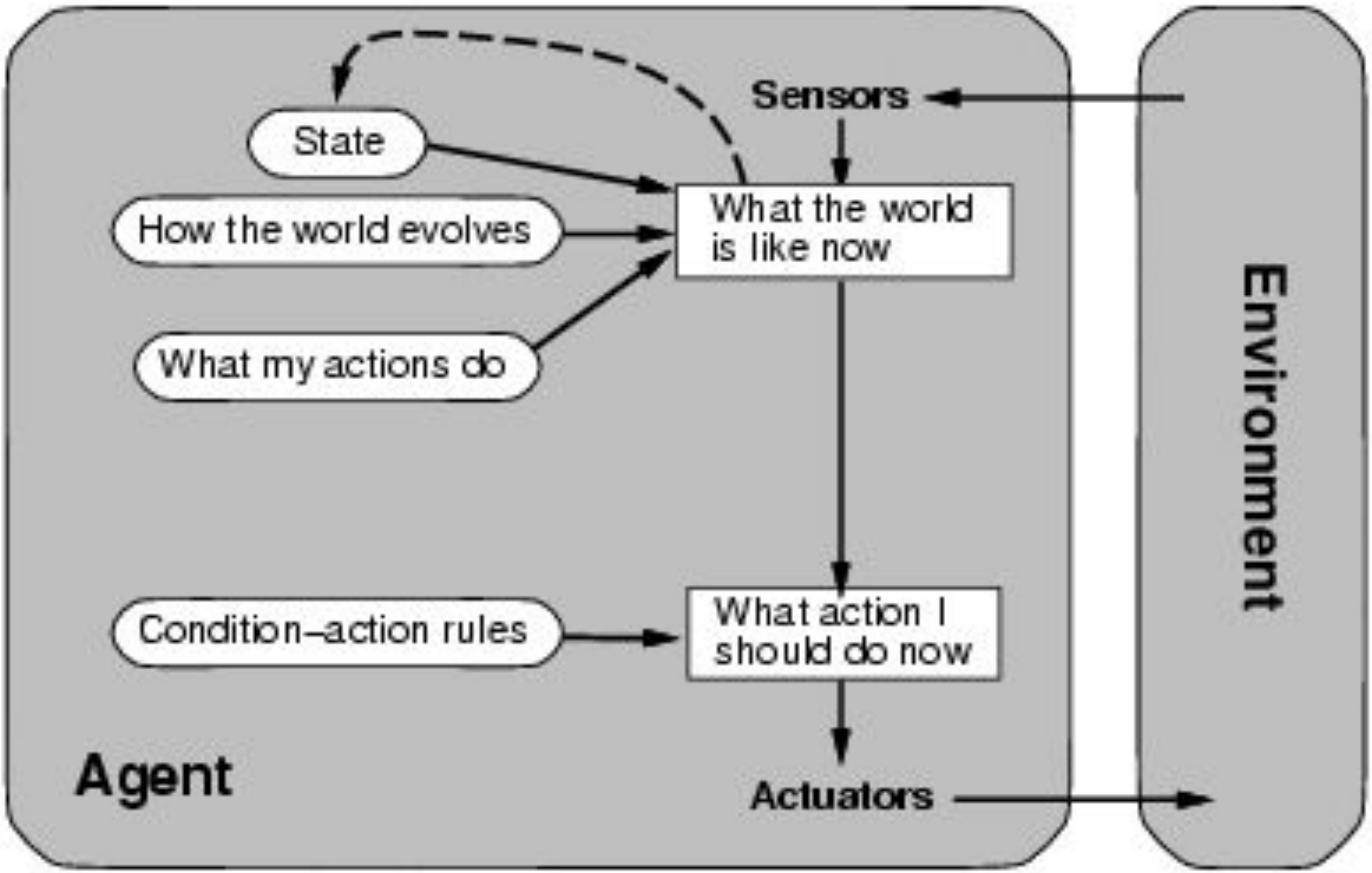


Actions: left, right, straight, open-door

Rules:

1. If open(left) & open(right) and open(straight) then choose randomly between right and left
2. If wall(left) and open(right) and open(straight) then straight
3. If wall(right) and open(left) and open(straight) then straight
4. If wall(right) and open(left) and wall(straight) then left
5. If wall(left) and open(right) and wall(straight) then right
6. If wall(left) and door(right) and wall(straight) then open-door
7. If wall(right) and wall(left) and open(straight) then straight.
8. (Default) Move randomly

Model-Based Reflex Agents



Model-Based Reflex Agents

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*

The agent is with memory

Goal Based Agents

- Knowing about the current state of the environment is not always enough to decide what to do;
- Additionally sort of goal information which describes situations that are desirable is also required.
- This allows the agent a way to choose among multiple possibilities , selecting the one which reaches a goal state.

Goal-Based Agents

- Current state of the environment is always not enough
- The goal is another issue to achieve
  - Judgment of rationality / correctness
- Actions chosen goals, based on
  - the current state
  - the current percept

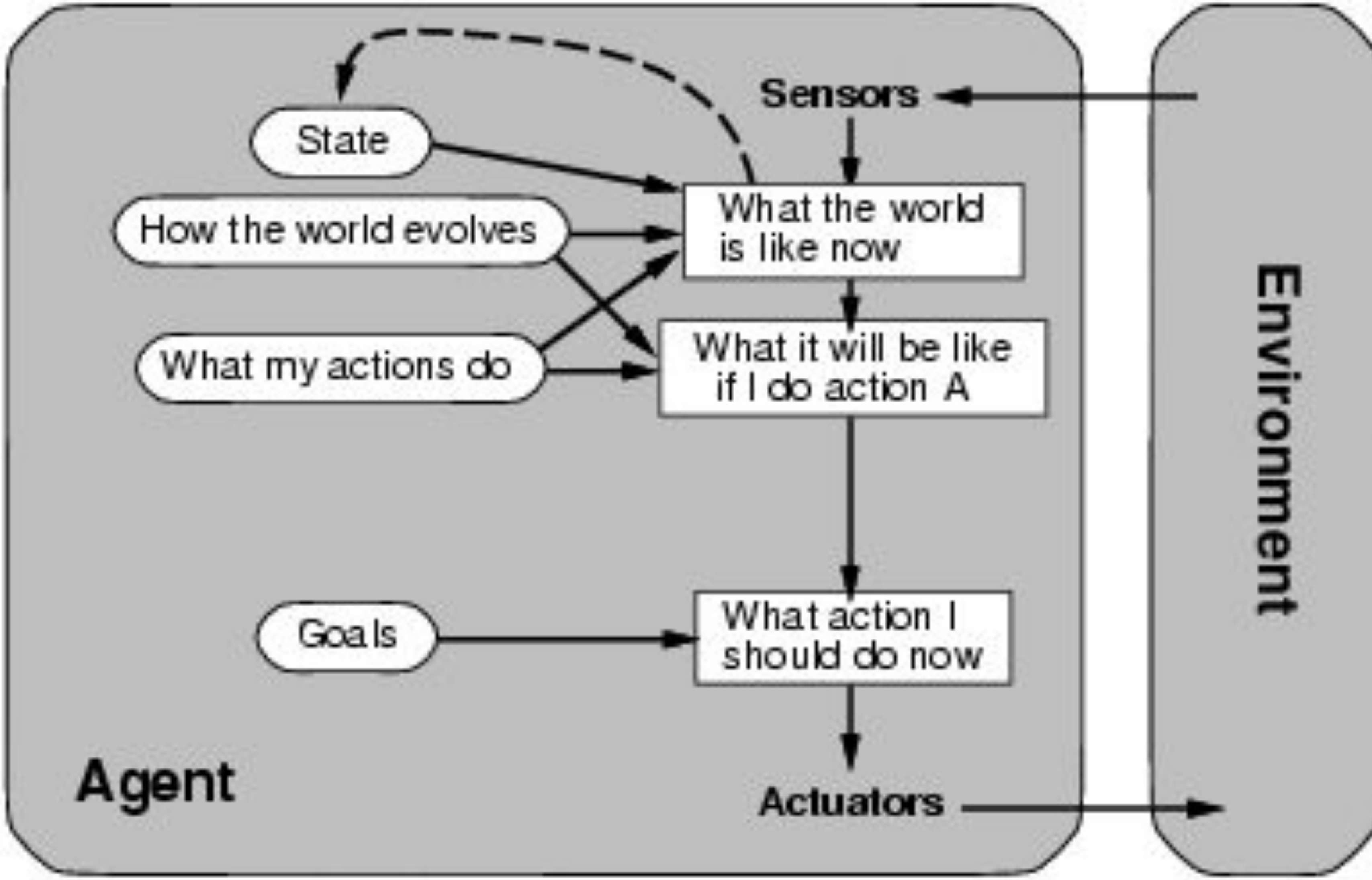
Goal-Based Agents

- Conclusion
  - Goal-based agents are less efficient
  - but more flexible
    - Agent □ Different goals □ different tasks
  - Search and planning
    - two other sub-fields in AI
    - to find out the action sequences to achieve its goal

A model based - Goal Based Agent

- It keeps track of the world state and a set of goals it is trying to achieve and picks an action that will eventually lead to the achievement of its goal.

Goal-Based Agents



Utility Based Agents

- Goal based agents only distinguish between goal states and non-goal states.
- It is possible to define a measure of how desirable a particular state is. This measure can be obtained through the use of a **utility function** which maps a state to a measure of the utility of the state.
- So, utility function maps a state onto a real number, which describes the associated degree of happiness.
- A complete specification of the utility function allows rational decisions.

Utility-Based Agents

- Goals alone are not enough
  - to generate **high-quality** behavior
  - E.g. meals in Canteen, good or not ?
- Many action sequences □ the goals
  - some are better and some worse
  - If goal means success,
  - then **utility** means the degree of success (how successful it is)

Utility-Based Agents

- it is said state A has higher utility
  - If state A is more preferred than others
- Utility is therefore a function
  - that maps a state onto a real number
  - the degree of success

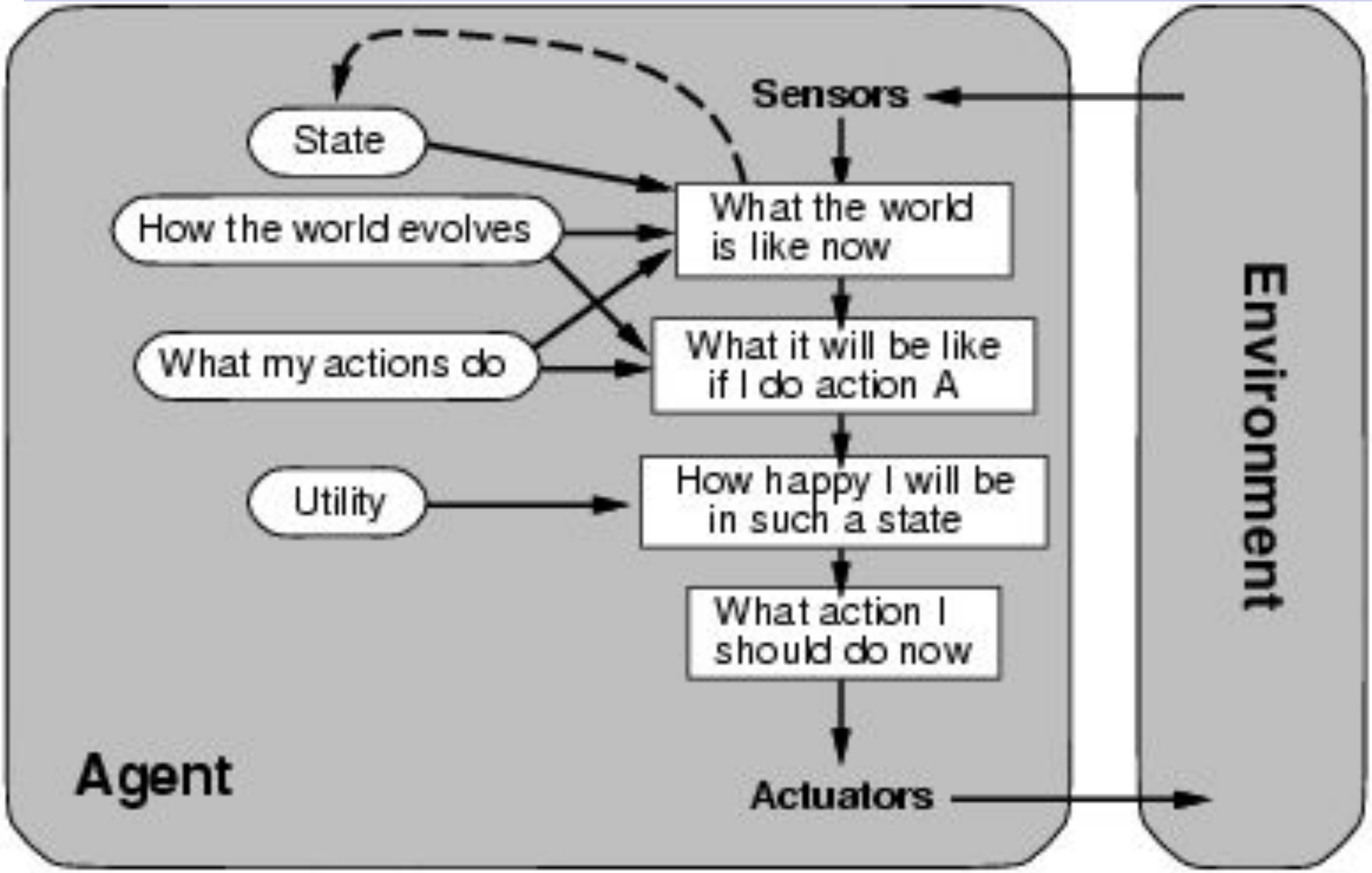
Utility-Based Agents

- Utility has several advantages:
  - When there are conflicting goals,
    - Only some of the goals but not all can be achieved
    - utility describes the appropriate trade-off
  - When there are several goals
    - None of them are achieved **certainly**
    - utility provides a way for the decision-making

A model based-Utility Based Agent

- It uses a model of the world along with a utility function that measures its preferences among states of the world. Then it chooses an action that leads to the best expected utility.

Utility-based agents



Learning Agents

- After an agent is programmed, can it work immediately?
  - No, it still need teaching
- In AI,
  - Once an agent is done
  - We teach it by giving it a set of examples
  - Test it by using another set of examples
- We then say the agent learns
  - A learning agent

Learning Agents

- Four conceptual components
  - Learning element
    - Making improvement
  - Performance element
    - Selecting external actions
  - Critic
    - Tells the Learning element how well the agent is doing with respect to fixed performance standard.

      (Feedback from user or examples, good or not?)
  - Problem generator
    - Suggest actions that will lead to new and informative experiences.

Learning Agents contd..

- Learning has an advantage that it allows the agents to initially operate in unknown environments and to become more competent than its initial knowledge alone might allow.

Learning Agents contd..

- Components:-
  - Learning element:- responsible for making improvements.
  - Performance elements:- responsible for selec ting actions.
  - Critics:- provides feedback
  - Problem generator:- responsible for suggesting actions that will lead to new experiences.

Learning Agents contd..