CHAPTER : TWO
Intelligent Agent

Content

- Intelligent Agent
- Agent Schematic
- Types of Agent
- Rational Agents
- Specifying Task Environments
- Types of Environments

Intelligent Agent

- I want to build a robot that will
  - Clean my house
  - Information filtering agents
  - Cook when I don’t want to
  - Wash my clothes
  - Cut my hair
  - Fix my car (or take it to be fixed)
  - Take a note when I am in a meeting
  - Handle my emails

i.e. do the things that I don’t feel like doing…

- AI is the science of building machines (agents) that act rationally with respect to a goal.

Category

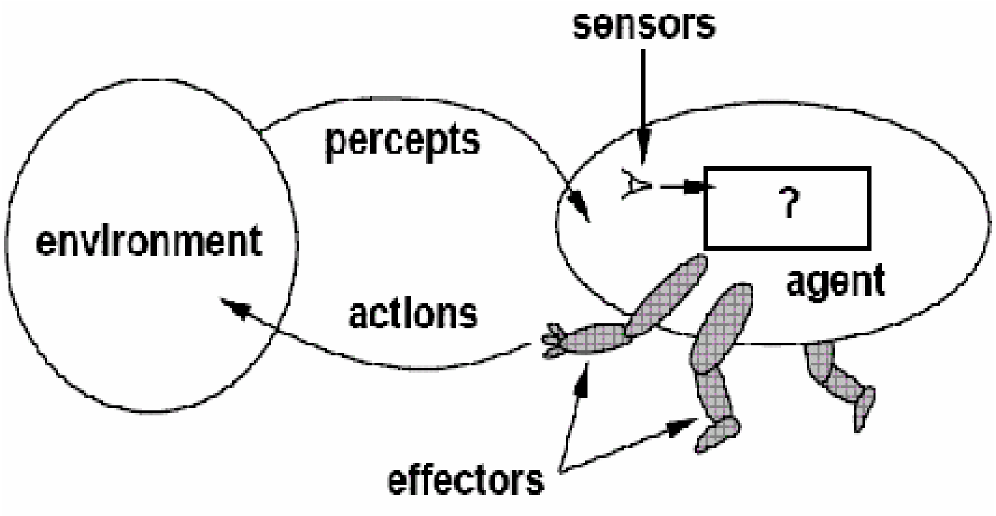
Types of Intelligent Agents

- Software agents:
  - Also called a softbot (software robot)
  - It is an agent that interacts with a software environment by issuing commands and interpreting the environments feedback.
  - E.g. mail handling agent, information filtering agent

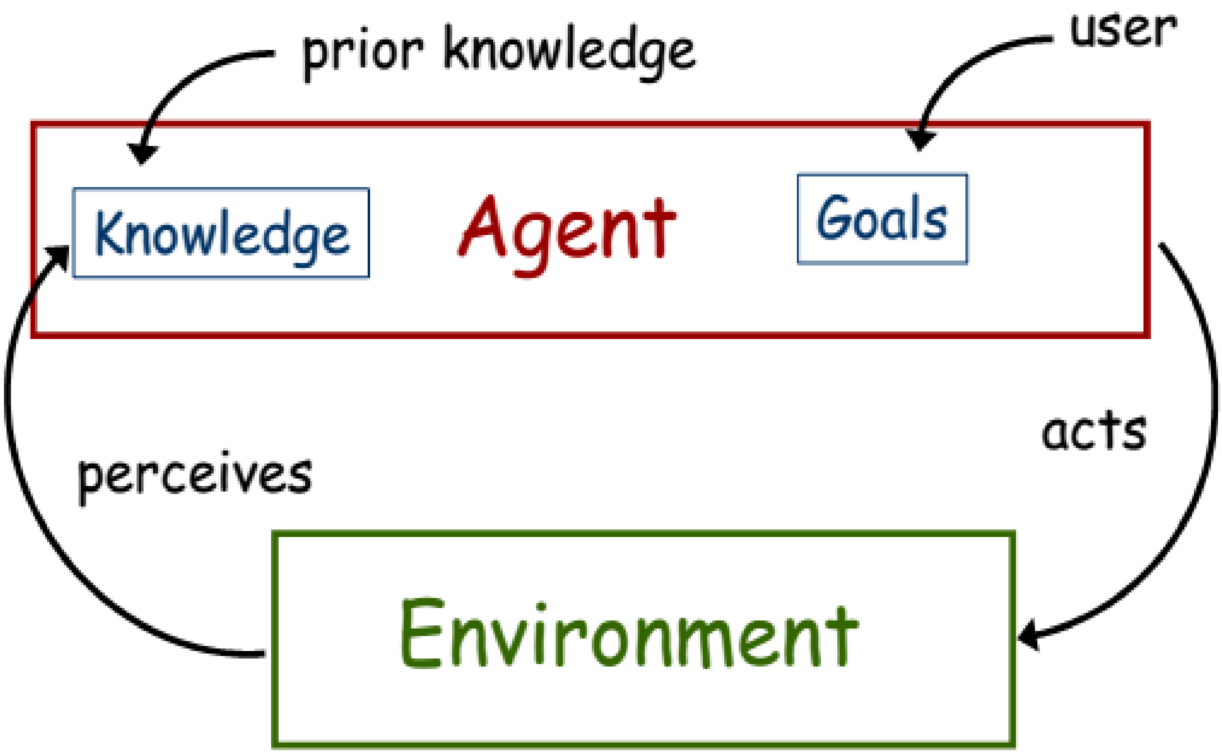
- Physical agents
  - are robots that operates in the physical world and can perceive and manipulate objects in that world

Agent

- Agent is something that perceives/observes its environment through **SENSORS** and acts upon that environment through **EFFECTORS**.
- The agent is assumed to exist in an environment in which it perceives and acts
- An agent is rational/sensible since it does the right thing to achieve the specified goal.



- **Agent = architecture + program**



- Require more flexible interaction with the environment, the ability to modify one's goals, knowledge that be applied flexibly to different situations

Agent

	Human beings	Agents
Sensors	Eyes, Ears, Nose	Cameras, Scanners, Mic, infrared range finders
Effectors	Hands, Legs, Mouth	Various Motors (artificial hand, artificial leg), Speakers, Radio

Examples of agents in different types of applications

Agent type	Percepts	Actions	Goals	Environment
Medical diagnosis system	Symptoms, patient's answers	Questions, tests, treatments	Healthy patients, minimize costs	Patient, hospital
Interactive English tutor	Typed words, questions, suggestions	Write exercises, suggestions, corrections	Maximize student's score on exams	Set of students, materials
Part-picking robot	Pixels of varying intensity	Pick up parts and sort into bins	Place parts in correct bins	Conveyor belts with parts
Satellite image analysis system	Pixels intensity, color	Print a categorization of scene	Correct categorization	Images from orbiting satellite
Refinery controller	Temperature, pressure readings	Open, close valves; adjust temperature	Maximize purity, yield, safety	Refinery

Rationality/Reasonableness vs. Omniscience/Awareness/knowledge

- Rational agent acts so as to achieve one's goals, given one's beliefs (one that does the right thing).
  - What does right thing mean? one that will cause the agent to be most successful and is expected to maximize goal achievement, given the available information
- An Omniscient/all-knowing agent knows the actual outcome of its actions, and can act accordingly, but in reality omniscience is impossible.
- Rational agents take action with expected success, where as omniscient agent take action with 100% sure of its success
- Is human beings Omniscient or Rational agent?

Example

- You are walking along the road to **shewaber**; You see an old friend across the street. There is no traffic.
- So, being rational, you start to cross the street.
- Mean while a big banner falls off from above and before you finish crossing the road, you are flattened.

Were you irrational to cross the street?

- This points out that rationality is concerned with expected success, given what has been perceived.
  - Crossing the street was rational, because most of the time, the crossing would be successful, and there was no way you could have foreseen the falling banner.
  - The EXAMPLE shows that we can not blame an agent for failing to take into account something it could not perceive. Or for failing to take an action that it is incapable of taking.

Rational agent

- In summary what is rational at any given point depends on four things.
  - **Perception/sensitivity**: Everything that the agent has perceived so far concerning the current scenario in the environment
  - **Knowledge**: What an agent already knows about the environment
  - **Action**: The actions that the agent can perform back to the environment
  - **Performance measure**: The performance measure that defines degrees of success of the agent.
- Generally, rationality refers to "doing the right thing".
  - ✓ So that one key property of Intelligent Agents is being a rational agent, because rational agents means doing the right thing.

- Therefore in designing an intelligent agent, one has to remember PEAS (Performance, Environment, Actuators, Sensors) framework.

Action

- **Example 1**: PEAS description for **Automated Taxi Driver.**
  a) Performance Measure
     - ✓ How much it is safe, fast, legal, comfortable trip, maximize profits, etc.

  b) Environment
     - ✓ The roads, pedestrians, customers, other traffic.

  c) Actuators
     - ✓ The steering wheel, accelerator, brake, signal, horn, etc.

  d) Sensors
     - ✓ The cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard, etc.

- **Example 2**: PEAS description of Part-Sorting Robot
    a) Performance Measure
        - ✓ Percentage of parts in correct bins
    b) Environment
        - ✓ Conveyor belt with parts, bins
    c) Actuators
        - ✓ Robotic arm
    d) Sensors
        - ✓ Camera, joint angle sensors

- **Example 3**: PEAS description of Spam Filter
  a) Performance Measure
     - ✓ Minimizing false positives, false negatives
  b) Environment
     - ✓ A user's email account
  c) Actuators
     - ✓ Mark as spam, delete, etc.
  d) Sensors
     - ✓ Incoming messages, other information about user's account

Performance measure

- How do we decide whether an agent is successful or not?
  - Establish a standard of what it means to be successful in an environment and use it to measure the performance.
  - A rational agent should do whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

- What is the performance measure for “**crossing the road**”?
- What about “**Chess Playing**”?

Assignment

- Consider the need to design a “taxi driver agent” that serves in Debre markos city;
  - Identify what to perceive, actions to take, the environment it interacts with?
  - Identify **sensors, effectors, goals, environment** and **performance measure** that should be integrated for the agent to be successful in its operation?

Designing an agent/Agent Function

- An **agent function** maps the percept histories to actions.
- An **agent program** is the one that runs on the physical architecture to produce the agent function.
- **Agent = architecture + program**
- Architecture
  - Runs the programs
  - Makes the percept from the sensors available to the programs
  - Feeds the program's action choices to the effectors
- Programs
  - Accepts percept from an environment and generates actions
    - Before designing an agent program, we need to know the possible percept and actions
  - By enabling a learning mechanism, the agent could have a degree of autonomy/independence, such that it can reason and take decision

2

Types of Environments

1. **Fully Observable (vs. Partially Observable)**
   - The agent's sensors give it access to the complete state of the environment at each point in time.
2. **Deterministic (vs. Stochastic)**
   - The next state of the environment is completely determined by the current state and the agent's action.
3. **Episodic (vs. Sequential)**
   - The agent's experience is divided into atomic "episodes," and the choice of action in each episode depends only on the episode itself.

4. **Static (vs. Dynamic)**
   - The environment is unchanged while an agent is deliberating.
   - **Semi dynamic**:- The environment does not change with the passage of time, but the agent's performance score does.
5. **Discrete (vs. Continuous)**
   - The environment provides a fixed number of distinct percepts, actions, and environment states.
   - Time can also evolve in a discrete or continuous fashion.

6. **Single Agent (vs. Multi-agent)**
    - An agent operating by itself in an environment

7. **Known (vs. Unknown)**
    - The agent knows the rules of the environment

Case Examples for Task Environment Types

- Example 1: Task environment types of three agents: *Chess*, *Taxi with Clock* and *Taxi without Clock*

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

- **Example 2**: Task Environments types of four agents: ***Chess***, ***Poker***, ***Image Analysis*** and ***Butler Robot***

Task Env.	Observable	Agents	Episodic	Static	Discrete
Chess	Fully	Multi	Sequential	static	discrete
Poker	Partially	Multi	Sequential	static	discrete
Image analysis	Fully	Single	Episodic	static	continuous
Butler Robot	Partially	Single	Sequential	dynamic	continuous

Environment Types

Below are lists of properties of a number of familiar environments

Problems	Observable	Deterministic	Episodic	Static	Discrete
Crossword Puzzle	Yes	Yes	No	Yes	Yes
Part-picking/cutting robot	No	No	Yes	No	No
Web shopping program	No	No	No	No	Yes
Tutor	No	No	No	Yes	Yes
Medical Diagnosis	No	No	No	No	No
Taxi driving	No	No	No	No	No

•Hardest case: a environment that is inaccessible, sequential, non-deterministic, dynamic, continuous.

Agents acting in an environment

3 Type

Hierarchy of Agent Types

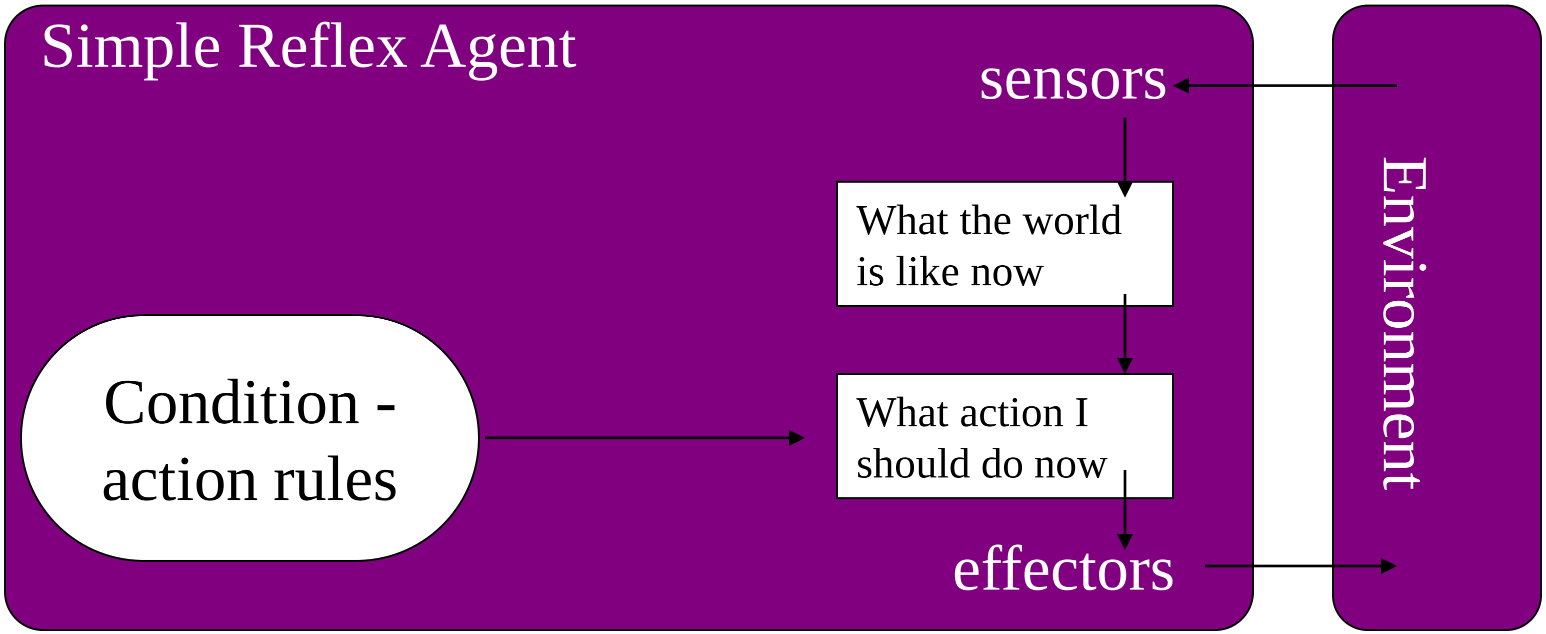
- Let's have a closer look on the following five agent categories:-
  - I. **Simple Reflex Agents**
  - II. **Model-Based Reflex Agents**
  - III. **Goal-Based Agents**
  - IV. **Utility-Based Agents**
  - V. **Learning Agents**

I. Simple Reflex Agents

- Select action on the basis of current percept, ignoring all past percepts
- It uses just ***condition-action rules***
  - ✓ The rules are like the form “if … then …”
  - ✓ Efficient but have narrow range of applicability
  - ✓ Because knowledge sometimes cannot be stated explicitly
  - ✓ Work only
    - If the environment is fully observable

➢ If car-in-front-is breaking then initiate-braking.
➢ Blinking when something approaches the eye.

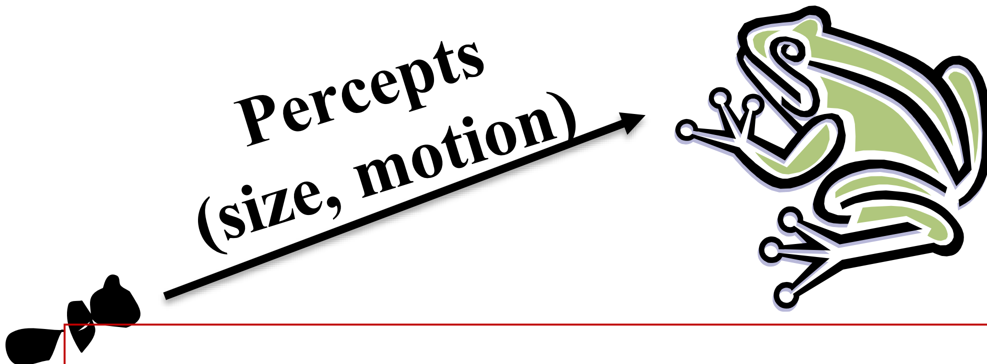
Figure: Structure of a simple reflex agent



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



RULES

(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

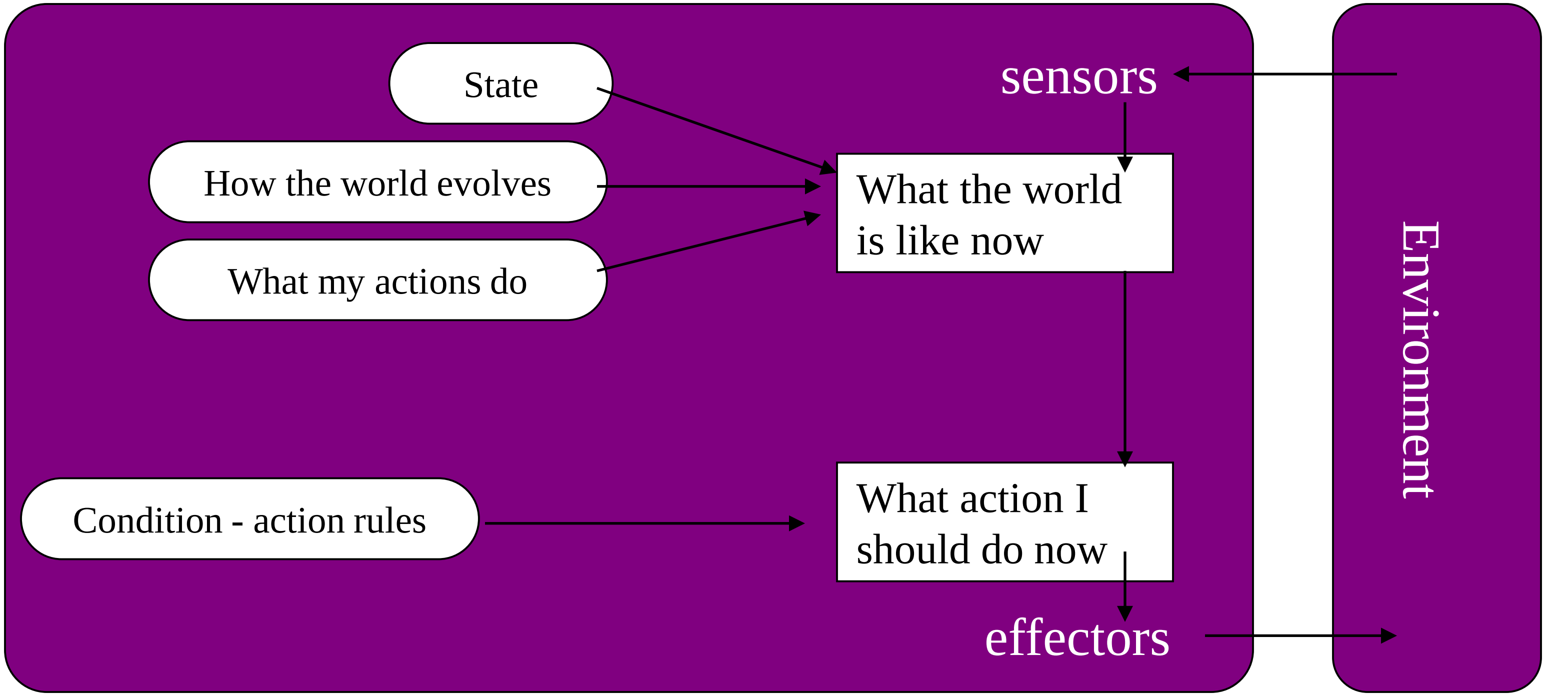
II. Model-Based Reflex Agents

- Maintains internal state that keeps track of aspects of the environment that cannot be currently observed
- For the world that is partially observable
  - **If the car is a recent model** - there is a centrally mounted brake light. With older models, there is no centrally mounted, so what if the agent gets confused?

    Is it a parking light? Is it a brake light? Is it a turn signal light?

    - ✓ The agent has to keep track of an internal state
      - That depends on the percept history
      - Reflecting some of the unobserved aspects
      - Example, 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
  - ✓ The agent is with memory

Figure: Structure of Model-Based reflex agent



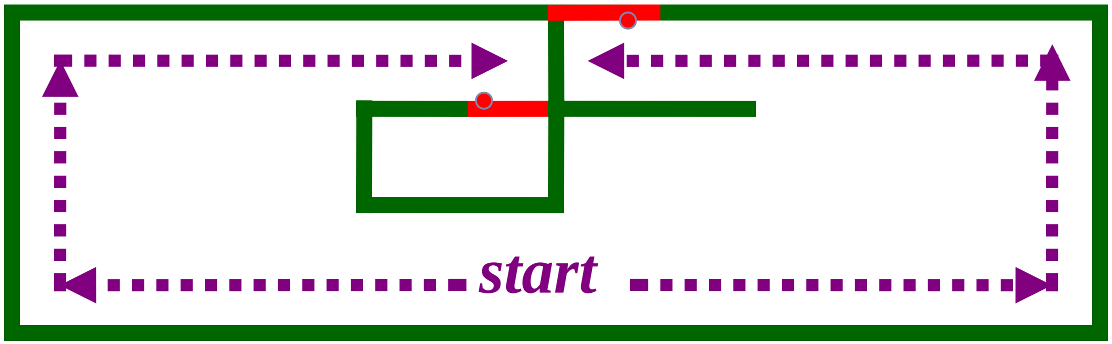
function REFLEX-AGENT-WITH-STATE (*percept*) **returns** action
static: *state*, a description of the current world state
rules, a set of condition-action rules

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

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

- Case Example for A Reflex Agent With Internal State
  - ✓ The **Wall-Following**



- The *Actions* are:-
  - ✓ Left
  - ✓ Right
  - ✓ Straight
  - ✓ Open-Door

- The ***Rules*** are:-
    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

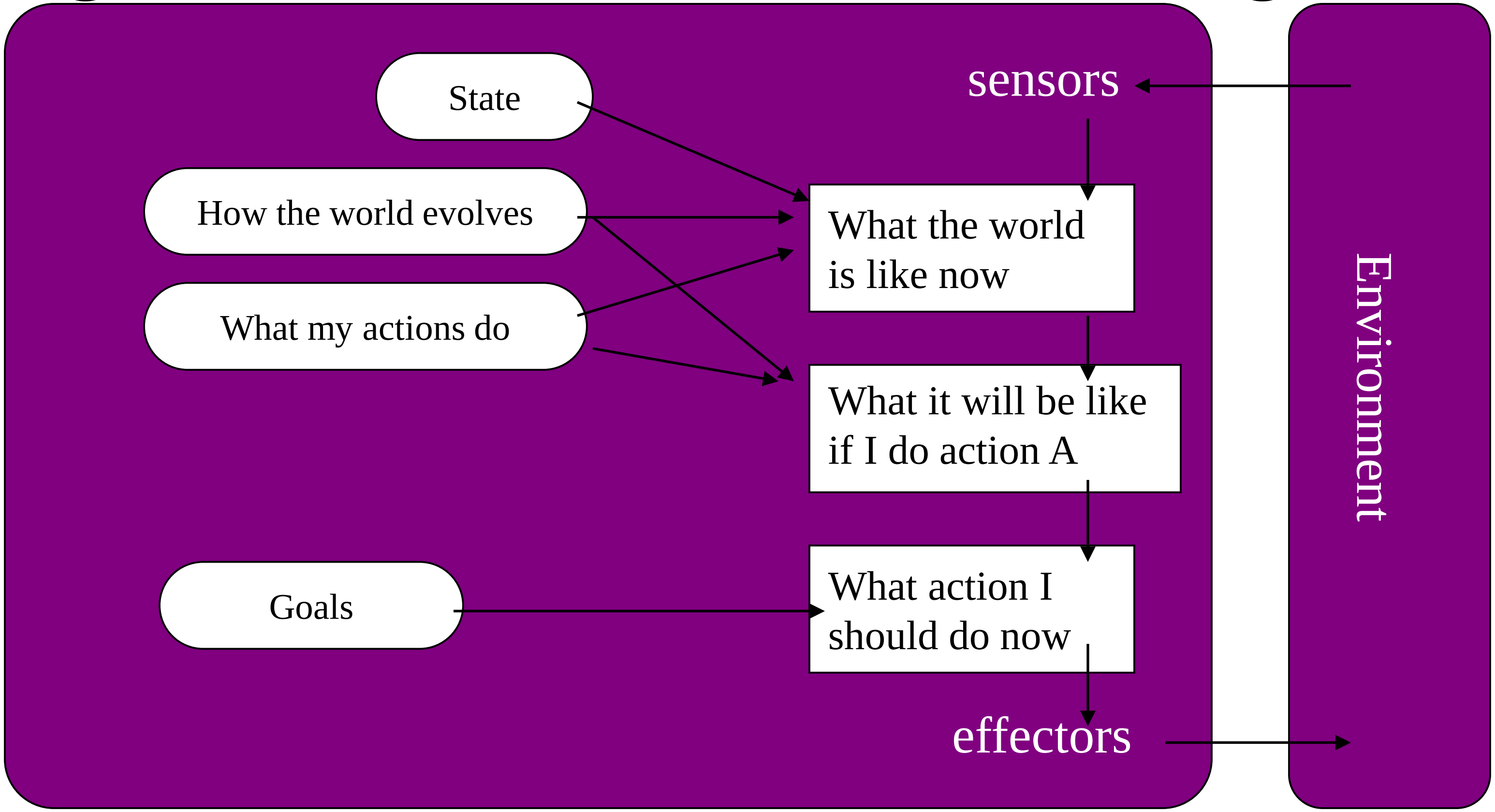
III. Goal-Based Agents

- The agent uses goal information to select between possible actions in the current state
- 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

- E.g. At a road junction, the taxi can turn left, right or go straight.

- 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

Figure: Structure of a Goal-based agent

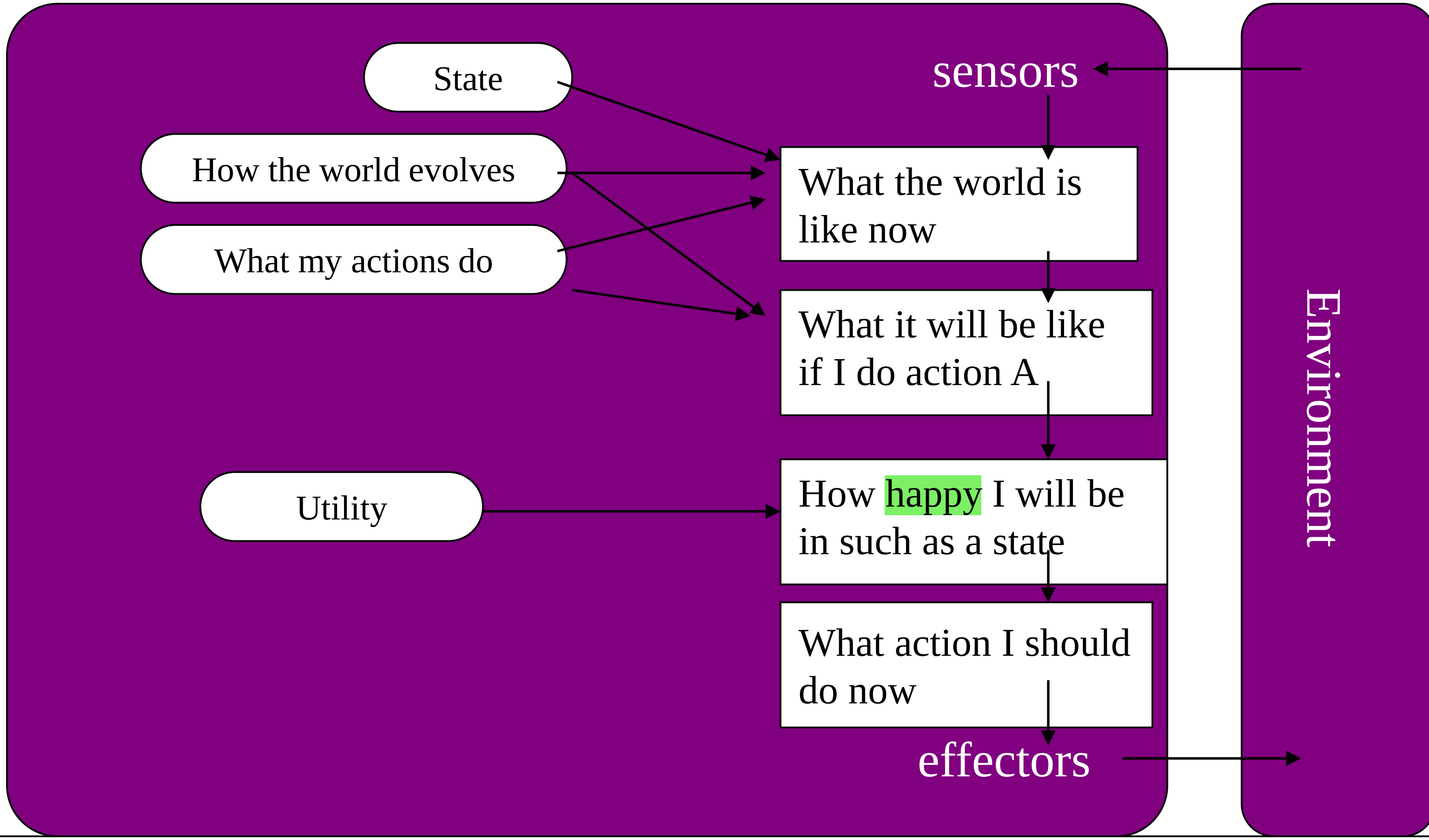


function GOAL_BASED_AGENT (*percept*) **returns** action
state ← UPDATE-STATE (*state*, *percept*)
action ← SELECT-ACTION [*state*, *goal*]
state ← UPDATE-STATE (*state*, *action*)
return *action*

IV. Utility-Based Agents

- The agent uses a utility function to evaluate the desirability of states that could result from each possible action
- Goals alone are not enough
  - ✓ To generate **high-quality** behavior
  - ✓ Example, 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).
- E.g. route recommendation system
  - – There are many action sequences that will get the taxi to its destination, thereby achieving the goal.
  - – Some are quicker, safer, more reliable, or cheaper than others. We need to consider Speed and safety

Figure: Structure of a utility-based agent



function UTILITY_BASED_AGENT (*percept*) **returns** action

state ← UPDATE-STATE (*state*, *percept*)
action ← SELECT-OPTIMAL_ACTION [*state*, *goal*]
state ← UPDATE-STATE (*state*, *action*)
return *action*

- 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

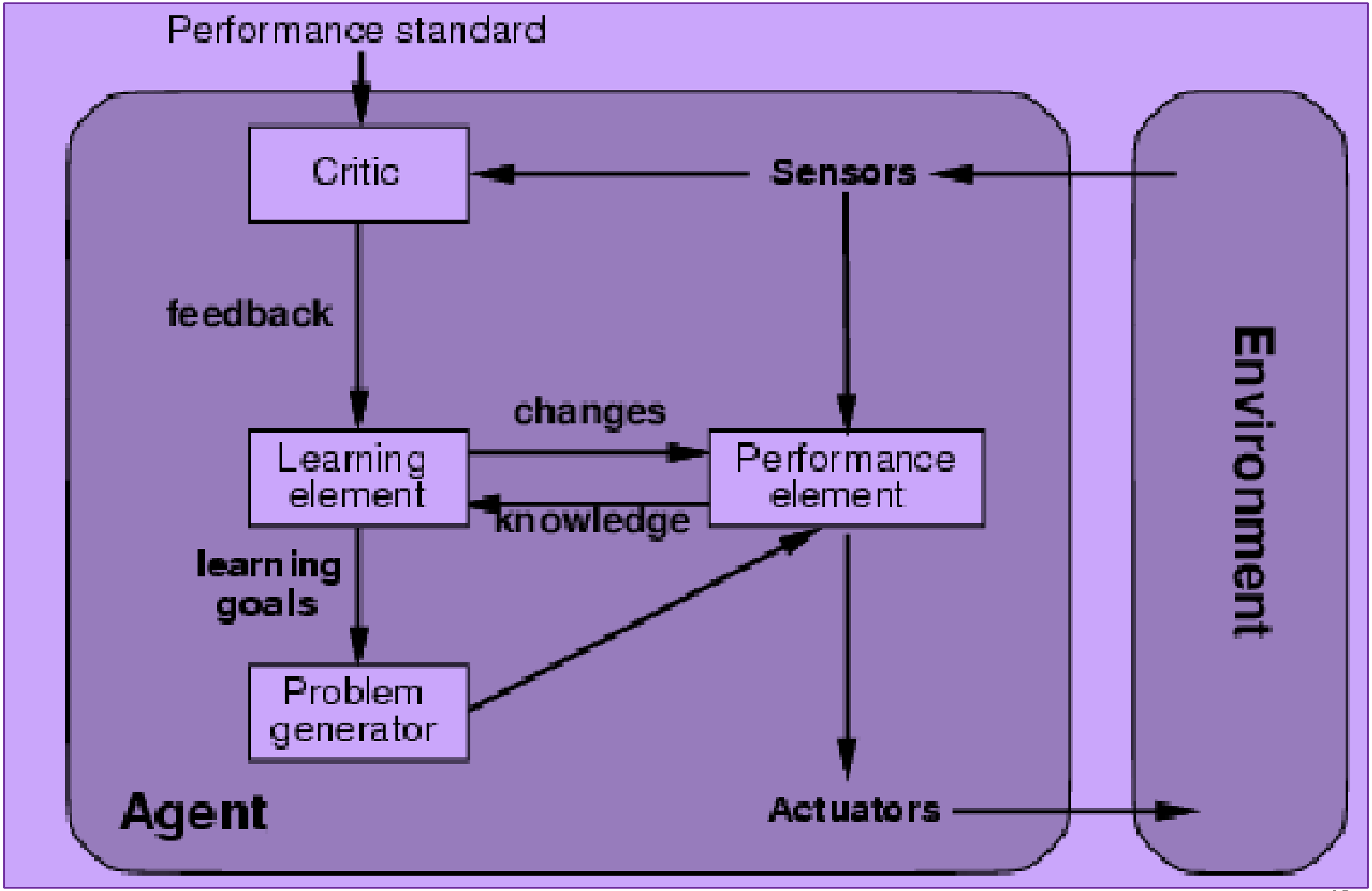
- Utility has several advantages:
  - When there are conflicting goals,
    - Only some of the goals but not all can be achieved
    - utility describes the appropriate trade-off
  - When there are several goals
    - None of them are achieved **certainly**
    - Utility provides a way for the decision-making

V. Learning Agents

- To build learning machines and then to teach them.
- 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 are able to perform tasks , analyze performance and look for new ways to improve on those tasks

- Four conceptual components
  - a) ***Learning Element***
    - ✓ Which is responsible for making improvement
  - b) ***Performance Element***
    - ✓ Which is responsible for selecting external actions
  - c) ***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.
    - ✓ Tells the Learning element how well the agent is doing with respect to fixed performance standard.
      - ○ Feedback from user or examples, good or not?
  - d) ***Problem Generator***
    - ✓ Responsible for suggesting actions that will lead to new and informative experiences

Figure: Structure of a Learning agent



Summery hierarchy in types of agents

- **Reflex agents:**
- ✓ These agents function in a current state, ignoring past history.
- ✓ Responses are based on the event-condition-action rule (ECA rule) where a user initiates an event and the agent refers to a list of pre-set rules and pre-programmed outcomes.
- **Model-based agents:**
- ✓ These agents choose an action in the same way as a reflex agent, but they have a more comprehensive view of the environment.
- ✓ A model of the world is programmed into the internal system that incorporates the agent's history.
- **Goal-based agents:**
- ✓ These agents expand upon the information model-based agents store by also including goal information, or information about desirable situations.

- **Utility-based agents:**

✓ These agents are similar to goal-based agents but provide an extra utility measurement which rates each possible scenario on its desired result and chooses the action that maximizes the outcome.

✓ Rating criteria examples could be the probability of success or the resources required.

- **Learning agents:**

✓ These agents have the ability to gradually improve and become more knowledgeable about an environment over time through an additional learning element.

✓ The learning element will use feedback to determine how performance elements should be changed to improve gradually.