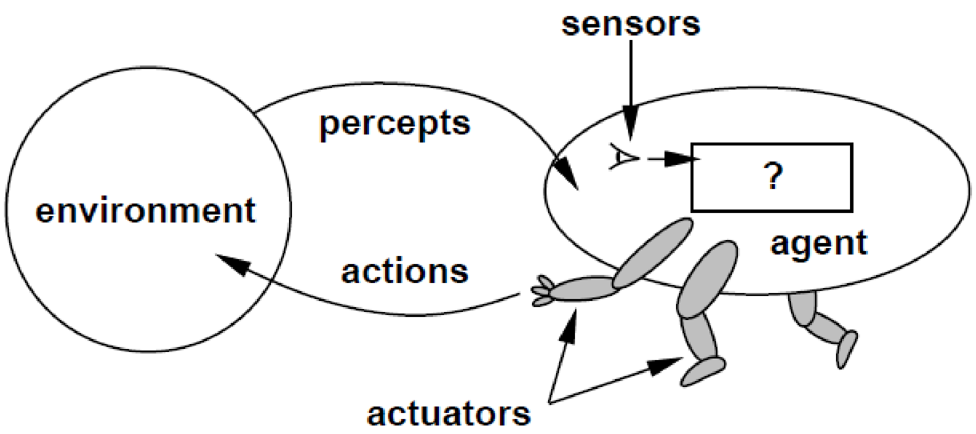
Chapter 2: Intelligent Agent

Agents

An *agent* is anything that can be viewed as

- *perceiving* its *environment* through *sensors* and
- *acting* upon that environment through *actuators*



Agents

- Agents include humans, robots, softbots,…etc
- Human agent:
  - Sensors: eyes, ears, ...
  - Actuators: hands, legs, mouth, …
- Robotic agent:
  - Sensors: cameras and infrared range finders
  - Actuators: various motors

- Software agent:
  - Sensors: keystrokes, file contents, and network packets
  - Actuators: screen, writing files, and sending network packets

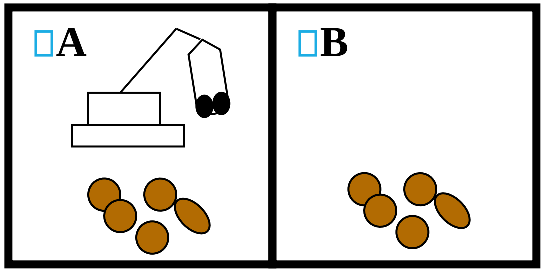
Agent function and agent program

- The agent function maps from percept histories to actions:

$$\mathcal{P}^* \rightarrow \mathcal{A}$$

- An agent program implements the agent function to run on a physical architecture.

Example: A Vacuum-cleaner agent



Percepts:location and contents, e.g. [***A, dirty]***

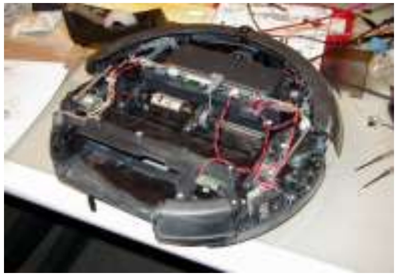
- *(Idealization: locations are discrete)*

Actions: ***LEFT, RIGHT, SUCK, NOP***

A Reflex Vacuum-Cleaner

```
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 $Left$
```

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



Rationality

Rationality depends on:

- Performance measure
- Agent’s (prior) knowledge
- Agent’s percepts to date
- Available actions

- **Rational Agent Definition**

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 Agent

- Rational agent maximizes the expected utility.
- We take the expectation of the utility due to uncertainty in environment (stochastic and partially-observable).

Rational Agent

- Consider the simple vacuum cleaner that cleans a square if it is dirty and moves to the other square if not. Is it rational?
  - It depends!
- Assume that there is a penalty of one point for each movement left or right, is ~~the~~ this simple cleaner rational?
  - No, it would perform poorly as it oscillates between right and left locations after cleaning them.
- What if clean squares can become dirty again?
  - The agent should occasionally check and re-clean them if needed.

Rationality versus Omniscience

- A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date.
- Rational ≠ omniscient
- An omniscient agent knows the actual outcomes of actions and acts accordingly.
- Rationality does not mean perfection!
- Rationality maximizes the *expected* performance, while perfection maximizes the *actual* performance.

Learning and autonomy

- An agent can learn from what it perceives.
- A rational agent should be autonomous—it should learn what it can to compensate for partial or incorrect prior knowledge.
- For example, a vacuum-cleaning agent that learns to forecast where and when additional dirt will appear will do better than one that does not.

Task environment

To design a rational agent, we need to specify a *task environment*

- a problem specification for which the agent is a solution

PEAS: ***to* specify a task environment**

- ***P***erformance measure
- ***E***nvironment
- ***A***ctuators
- ***S***ensors

PEAS: Specifying an automated taxi driver

***P*erformance measure**:
- ?

***E*nvironment**:
- ?

***A*ctuators**:
- ?

***S*ensors**:
- ?

PEAS: Specifying an automated taxi driver

***P*erformance measure**:

- safety, speed, legal, comfortable, maximize profits

***E*nvironment**:

- ?

***A*ctuators**:

- ?

***S*ensors**:

- ?

PEAS: Specifying an automated taxi driver

***P*erformance measure**:

- safe, fast, legal, comfortable, maximize profits

***E*nvironment**:

- roads, other traffic, pedestrians, customers

***A*ctuators**:

- ?

***S*ensors**:

- ?

PEAS: Specifying an automated taxi driver

Performance measure:

- safe, fast, legal, comfortable, maximize profits

Environment:

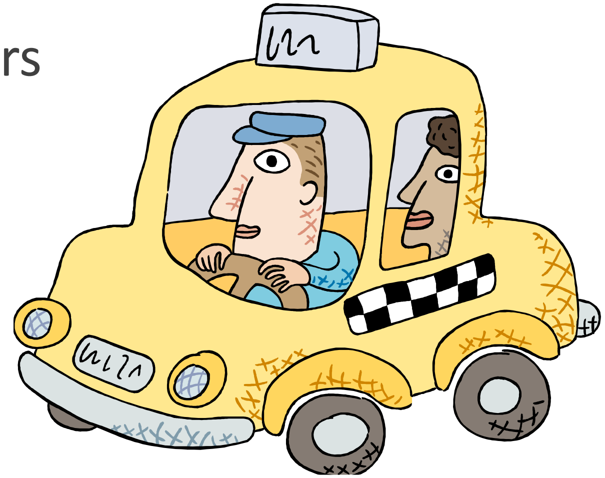
- roads, other traffic, pedestrians, customers

Actuators:

- steering, accelerator, brake, signal, horn

Sensors:

- ?



PEAS: Specifying an automated taxi driver

***P*erformance measure**:

- safe, fast, legal, comfortable, maximize profits

***E*nvironment**:

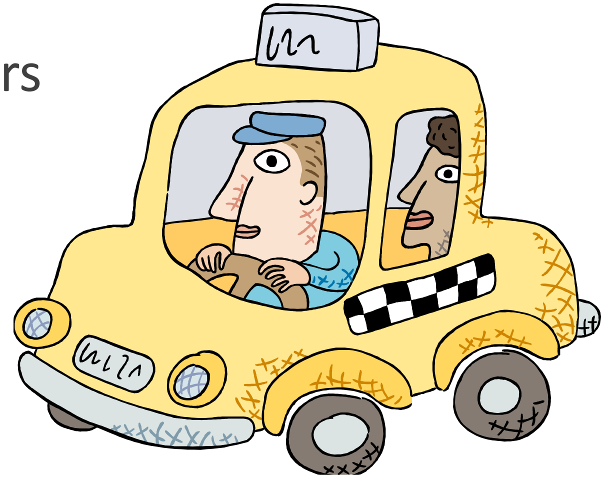
- roads, other traffic, pedestrians, customers

***A*ctuators**:

- steering, accelerator, brake, signal, horn

***S*ensors**:

- cameras, sonar, speedometer, GPS



PEAS: Internet Shopping Agent

- Performance measure: price, quality, appropriateness, efficiency
- Environment: current and future WWW sites, vendors, shippers
- Actuators: display to user, follow URL, fill in form
- Sensors: HTML pages (text, graphics, scripts)

PEAS: Spam Filtering Agent

- Performance measure: false positives, false negatives
- Environment: email client or server
- Actuators: mark as spam, delete,…
- Sensors: emails , traffic, etc.

Task 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.
- In a fully-observable and deterministic environment the agent need not deal with uncertainty.

- Episodic (vs. sequential): An episodic environment means that **subsequent episodes do not depend on what actions occurred in previous episodes**. Such environments do not require the agent to plan ahead.

Task Environment Types

- Static (vs. dynamic): **An environment which does not change** while the agent is thinking is static.
- In a static environment the agent need not worry about the passage of time while he is thinking, nor does he have to observe the world while he is thinking.
- In static environments the time it takes to compute a good strategy does not matter.
- If the environment itself does not change with the passage of time but the agent's performance score does, then we say the environment is **semi-dynamic**.

Environment Types

Discrete (vs. continuous): : If the number of **distinct percepts and actions is limited the environment is discrete**, otherwise it is continuous.

Single agent (vs. multi-agent): An agent operating by itself in an environment.

- If more than one agent exists consider cooperation, coordination, competition, communication or random behavior.

What’s the real world like?

Environment Types

	Crossword puzzle	Back-gammon	Part-picking Robot	Taxi
Fully-Observable	✓	✓	✗	✗
Deterministic	✓	✗	✗	✗
Episodic	✗	✗	✓	✗
Static	✓	✓	✗	✗
Discrete	✓	✓	✗	✗
Single-Agent	✓	✗	✓	✗

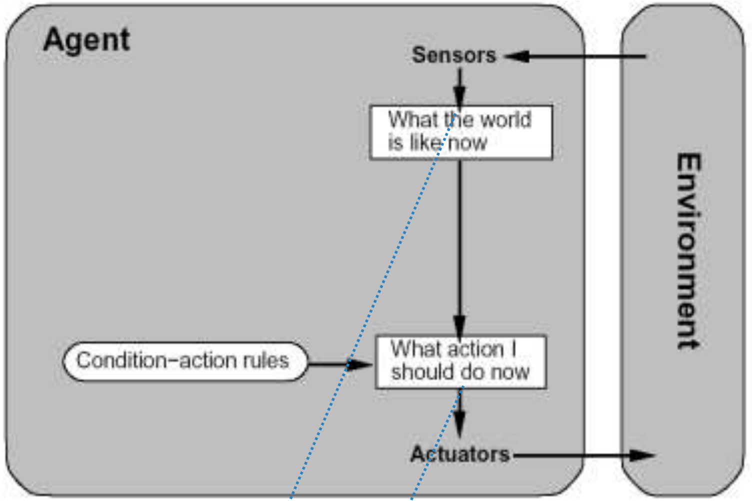
- The environment type largely determines the agent design
- The real world is partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Agent Structure

- Agent= Architecture+ Program
- The job of AI is to design the agent program that implements the agent function mapping percepts to actions.

da byst5dm el sensors, 34an ye2ra el enviroment 3amel ezay.
b3d keda bykon 3ndo table, mwgod feh instructions, by2olo lama tla2y el state el folanya, khud el action el folany.. .
w bynfz el action da bl actuators bto3o 3la el enviroment.

Simple reflex agent

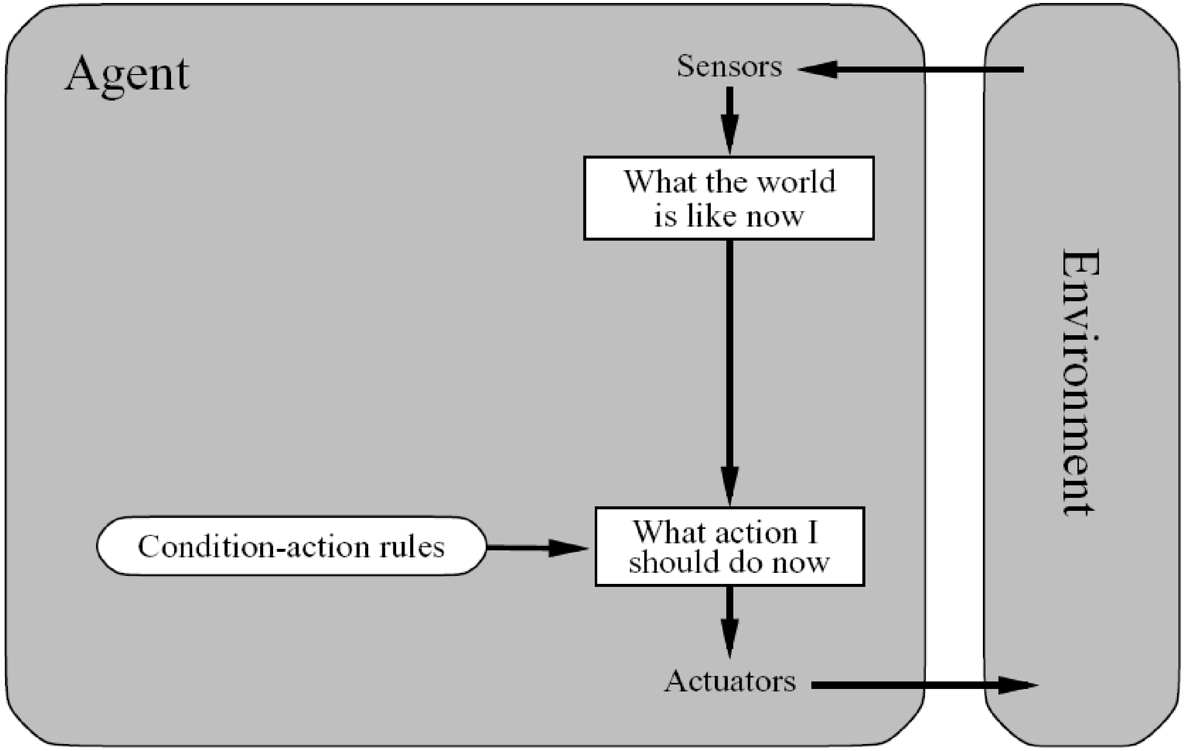


```
function REFLEX_VACUUM_AGENT( percept )
        returns an action

        (location,status) = UPDATE_STATE( percept )

        if status = DIRTY then return SUCK;
        else if location = A then return RIGHT;
        else if location = B then return  LEFT;
```

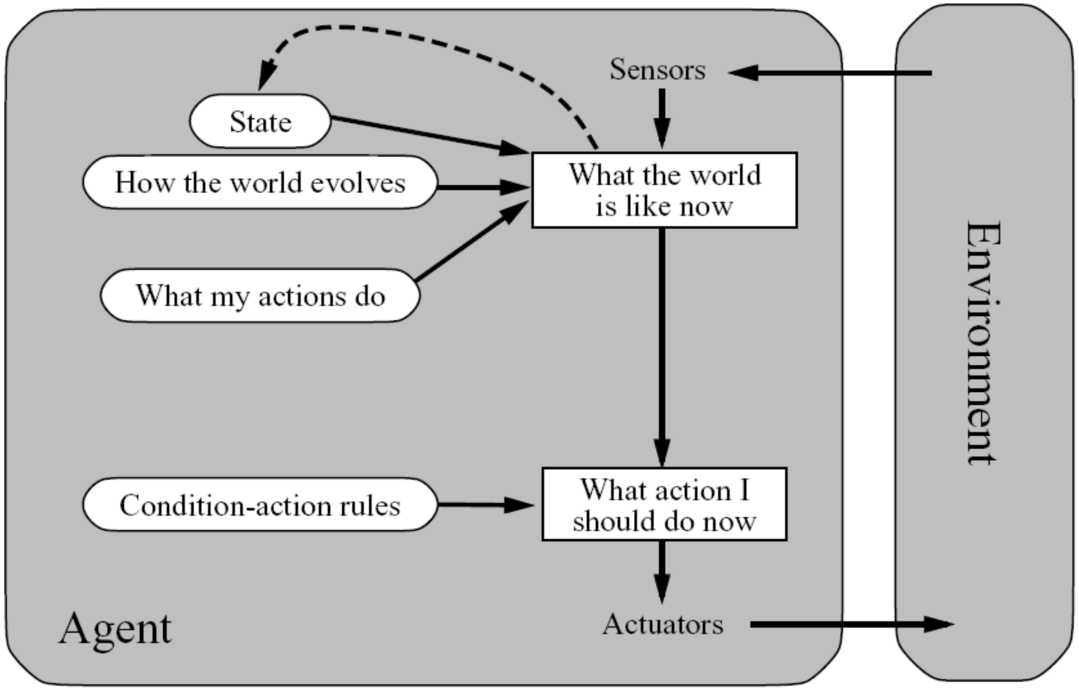
Simple Reflex Agents



not real, and very simple.

What is the problem of this design?

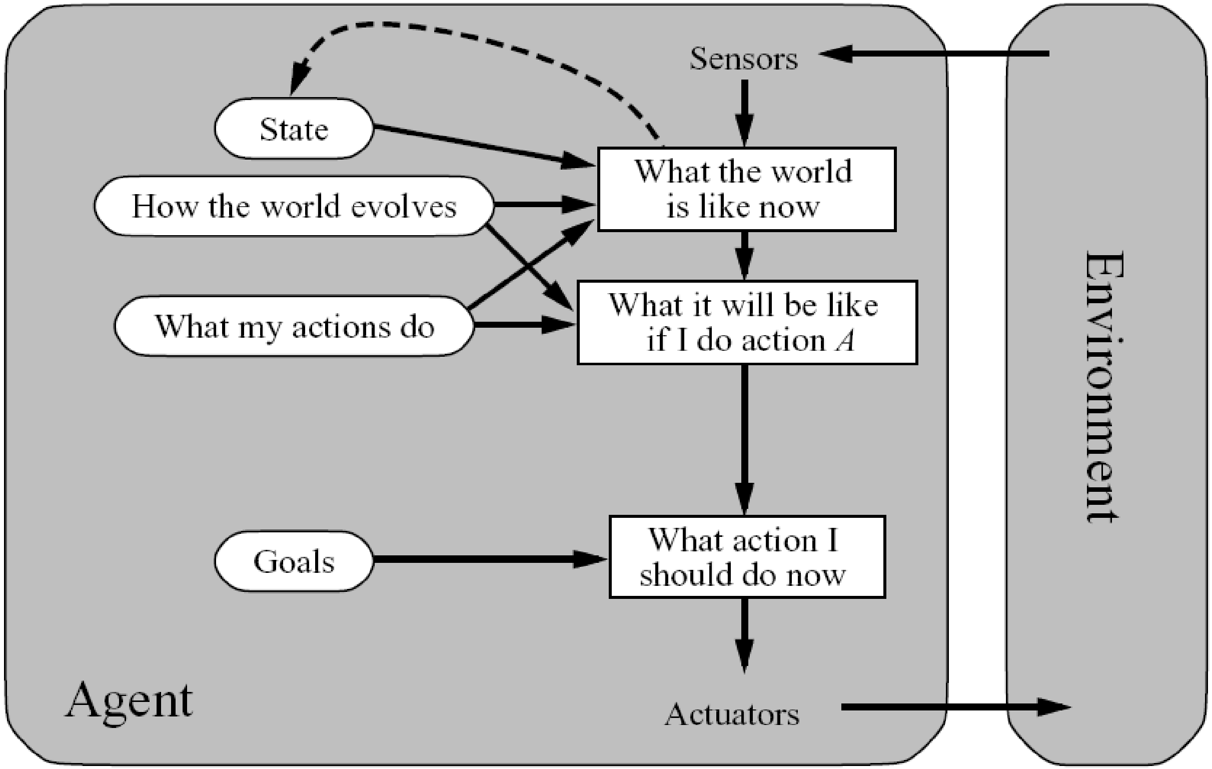
Reflex Agents with State (Model)



- Can handle partially observable environments.
- By keeping an internal state of the world (a model of the world) defining how the world evolves. Not exactly (uncertainty/inference).

here we do not explicitly define the actions that the agent should do, but we define for him a goal, that he should always try to achieve.

Goal-Based Agents

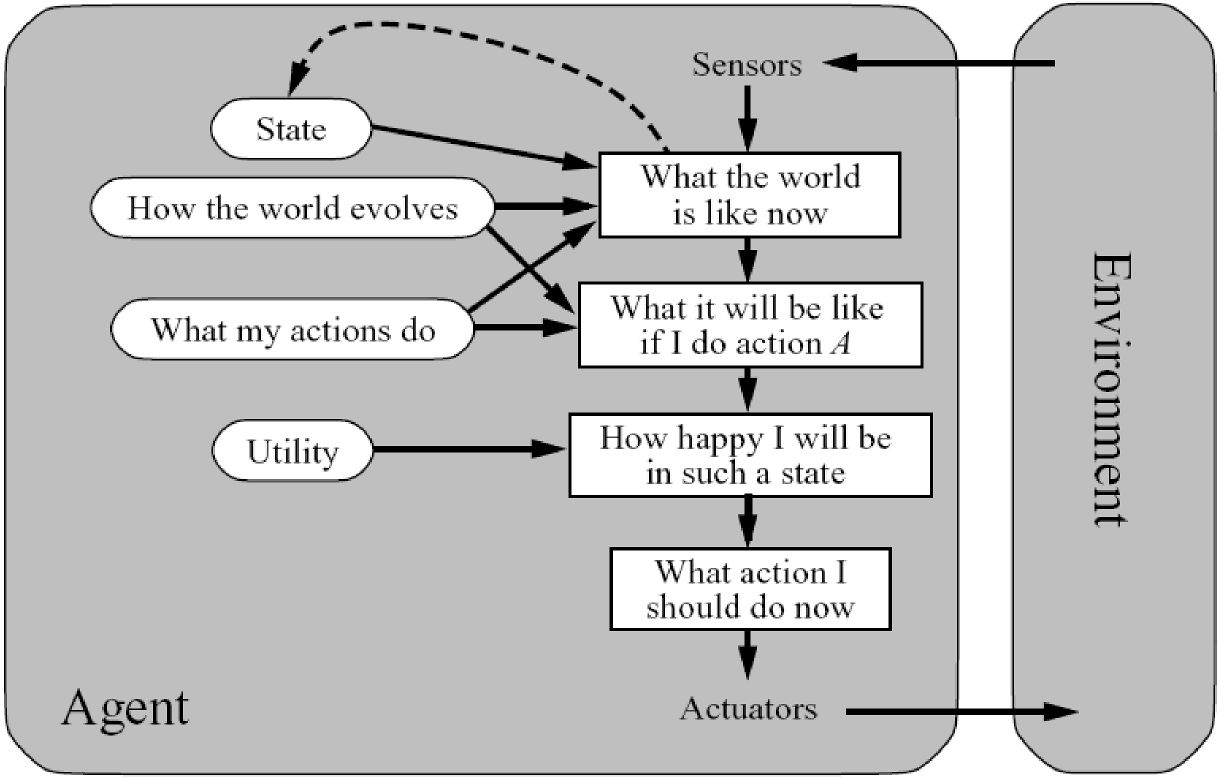

Sensors
State
What the world is like now
How the world evolves
What it will be like if I do action *A*
What my actions do
Goals
What action I should do now
Agent
Actuators
Environment

Goal-Based Agents

- Embed the goal info describing the agent desirable behavior.
- These agents usually first find plans then execute them.
- Examples: Search (Ch3-5) and planning (Ch10)

- More adaptive to different environments than reflex agents.

Utility-Based Agents


Sensors
State
What the world is like now
How the world evolves
What it will be like if I do action *A*
What my actions do
Utility
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

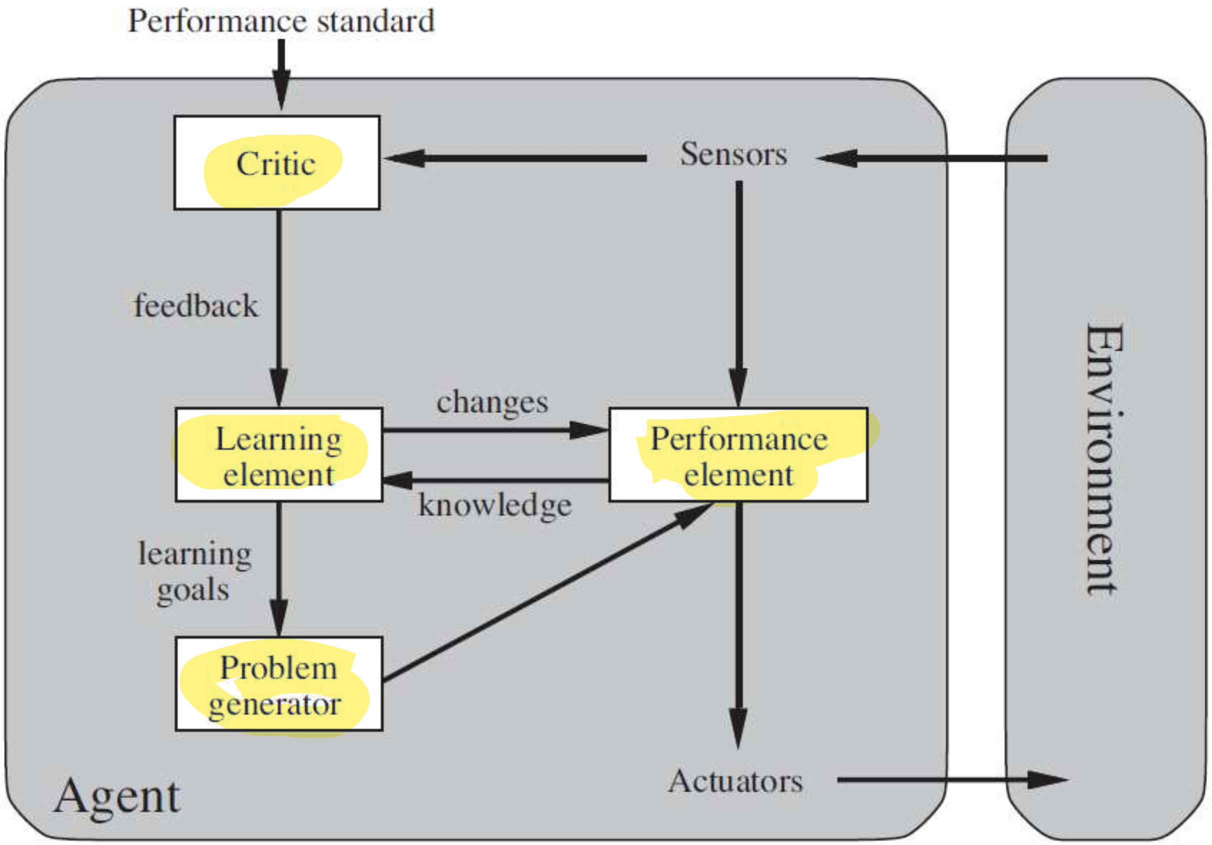
- **How is this different from a goal-based agent?**

Utility-Based Agents (cont.)

- Goals alone are not enough to generate high-quality behavior in most environments
- The utility defines performance measure.
- Can combine multiple goals into a single utility function and can weight them according to their importance.
- For example: The auto driver agent, which way is safer/quicker.
- Simple and complex decisions (Ch16,17)

Performance element: ay no3 mn anwa3 el agents elly etklmna 3nhom fo2.
Critic: Da module, by2olak enta shaghal kwys wla we7sh w bydeek feedback.
Learning element: da el module elly byt3lm w yebd2 y3ml t3delat fe el performance element.
Problem generator: da wazfto eno kol shwya ye5l2lk moshkela mn el la she2, 34an bs y5lek tgrb 7agat gededa w tt3lm menha.

Learning Agent



Learning Agent

- The learning agent allows the agent to operate in initially unknown environments and it can adapt to different environments as well.

- The components of the learning agent:
  - **Learning element:** it improves the agent's performance
  - P**erformance element**: it takes in percepts and decides on actions. (Agent itself in the previous structures)
  - **Critic:** It gives feedback to the learning element on how the agent is doing and determines how the performance element should be modified to do better in the future
  - **Problem generator**. It suggests actions that will lead to new and informative experiences. (exploration)

Summary

- Agents interact with environments through actuators and sensors
  - The agent function describes what the agent does in all circumstances
  - The agent program implements the agent function

- A rational agent maximizes expected performance
- PEAS descriptions define task environments
- Environments are categorized along several dimensions:
  - Observable? Deterministic? Episodic? Static? Discrete? Single-agent?
- Agent program types:
  - Simple reflex, model-based agents, reflex agents, goal-based agents, utility based agents, and learning agents