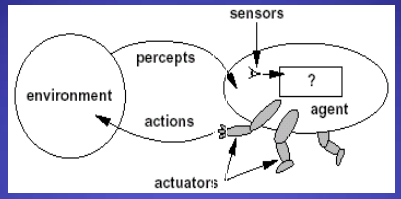
Chapter 2

Intelligent Agents

Outline

- Agents and Environments
- Rationality
- PEAS: Specification of Task Environment
  - Performance Measure ⇒ Goals
  - Environment
  - Actuators ⇒ Actions
  - Sensors ⇒ Percepts
- Environment types
- Agent types

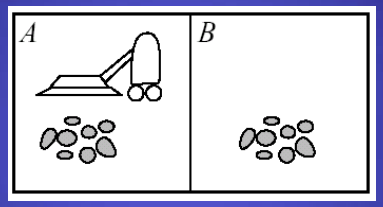
Agents and Environments



- An *agent* is an entity that can be viewed as *perceiving* its *environment* through *sensors* and *acting* upon that environment through *actuators*.
  - e.g.) humans, robots, softbots, thermostats, etc.
    - Softbots (Software robots) are intelligent software agents that use software tools and services on a person's behalf
  - *Percept*: the agent's perceptual inputs at any given instant.
  - *Percept sequence*: the complete *history* of everything the agent has ever perceived.
- The *agent function* maps from *percept histories* to *action*:
  $f : P^* \rightarrow A$
- The *agent program* runs on the physical architecture to produce f.
- Agent = architecture + agent program
- **Goal:** To design the rational agents (which do a good action on its environment)

Agents and Environments

Vacuum-Cleaner World



- Percepts: location and contents, e.g.) [*A, Dirty*]
- Actions: *Left, Right, Suck, NoOp.*

5

A Vacuum-Cleaner Agent

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*
:	:

```
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
```

- What is the *right* function?
  - The action which will cause the agent to be *successful*.
- Can it be implemented in a small agent program?

6

Rationality

- 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 the *most successful*.
- *Performance measure*: an objective criterion for success of an agent's behavior.
  -- Evaluate any given sequence of environment states.
- Fixed performance measure evaluates the environment sequence
  - e.g.) amount of dirt cleaned up, amount of electricity consumed, amount of noise generated, time taken, etc
  - 1 point per square cleaned up in time *T* ?
  - 1 point per clean square per time step - 1 point per move?
  - Penalize for > *k* dirty squares?
- A *rational agent* chooses whichever action maximizes the *expected* value of the performance measure *given the percept sequence to date*.

Rational ≠omniscient (all knowing with infinite-knowledge)
Rational ≠ clairvoyant
Rational ≠ successful

7

.. continued

- Rational ⇒ *exploration, learning, autonomy*
  - Agents can perform actions in order to modify future percepts so as to obtain useful information (*information gathering* provided by *exploration*)
  - An agent is *autonomous* if its behavior is determined by its own experience (with ability to learn and adapt)
- *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.*

8

The Nature of Environment -- PEAS

- To design a rational agent, we must specify the *task environment*
- Consider, e.g. the task of designing an automated taxi:
- Performance measure??
- Environment??
- Actuators??
- Sensors??

9

PEAS

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

  Consider, e.g. the task of designing an Automated Taxi:
- *Agent:* Automated Taxi
- Performance measure??
  - *Safety, speed, destination, profits, legality, comfort ..*
- Environment??
  - *US streets/freeways, traffic, pedestrians, customers, weather…*
- Actuators??
  - *Steering, accelerator, brake, signal, horn, speaker/display,…*
- Sensors??
  - *Cameras, speedometer, accelerometers, gauges, engine sensors, GPS,..*

10

.. Continued: PEAS

- *Agent*: Medical Diagnosis System
  - *Performance measure*:
    - Healthy patient, minimize costs, lawsuits
  - *Environment*:
    - Patient, hospital, staff
  - *Actuators*:
    - Screen display of questions, tests, diagnoses, treatments, referrals, images of organ with a tumor, etc.
  - *Sensors*:
    - Keyboard entry of symptoms, findings, patient's answers, CT, PET/CT, MRI, etc.

11

.. Continued: PEAS

- *Agent:* Intelligent Shopping Agent
  - *Performance measure:*
    - price, quality, appropriateness, efficiency
  - *Environment:*
    - E-shopping Websites, vendors, shippers
  - *Actuators:*
    - Display to user, follow URL, fill in form.
  - *Sensors:*
    - HTML pages (text, graphics, scripts)

12

.. Continued: PEAS

- *Agent:* Drone Delivery Agent (in Amazon)
  - *Performance measure:*
  - *Environment:*
  - *Actuators:*
  - *Sensors:*

13

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.
- *Single agent* (vs. *multiagent*):
  - An agent operating by itself in an environment.
  - *Multiagent env.*– competition, cooperation, communication
- *Deterministic* (vs. *stochastic(or nondeterministic)*):
  - The *next state* of the environment is *completely determined* by the *current state* and the *action* executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is *strategic*)
- *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* (not on the actions taken in previous episode).

14

.. Continued: Environment Types

- *Static* (vs. *dynamic*):
  - The environment is *unchanged* while an agent is deliberating. (The environment is *semidynamic* if the environment itself does not change with the passage of time but the agent's performance score does)
- *Discrete* (vs. *continuous*):
  - The environment has a finite number of distinct states, clearly defined set of percepts and actions.
    - E.g.) chess game vs. taxi-driving
- *Known*(vs. unknown):
  - Distinction of the agent's state of knowledge a/b the laws of physics.
  - Known env. - the outcomes for all actions are given.
  - Unknown env. - the agents will have to learn how it works in order to make good decisions.

15

Environment Types

	Crossword puzzle	Backgammon	I. Shopping	Taxi
Observable??	Yes	Yes	Partial	Partial
Deterministic??	Yes	Stochastic	Partly	Stochastic
Episodic??	Sequential	Sequential	Sequential	Sequential
Static??	Yes	Yes	Semi	Dynamic
Discrete??	Yes	Yes	Yes	Continuous
Single-agent??	Yes	Multi	Yes (except auctions)	Multi

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

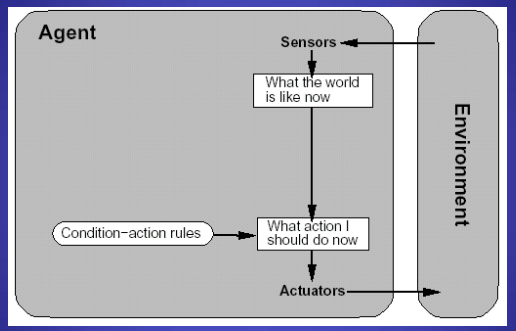
16

Agent Types

- Four basic types in order of increasing generality:
  - Simple reflex agents
  - Model-based Reflex agents
  - Goal-based agents
  - Utility-based agents
- All these can be turned into learning agents

17

Simple Reflex Agents



- An agent whose action depends only on the ***current percept***

18

Continued.. Simple Reflex Agents

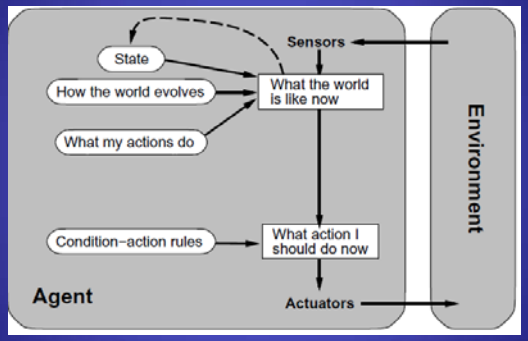
```
function SIMPLE-REFLEX-AGENT(percept) returns an action
  persistent: rules, a set of condition–action rules

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

Figure 2.10 A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

- a condition-action rule: e.g.
  if *car-in-front-is-breaking* then *initiate-breaking*
- Work only if the current decision can be made on the basis of the ***current percept*** only, *ignoring the percept history*.
- Find the rule whose condition matches with the current state, then does action associated with that rule.

19

Model-Based Reflex Agents
-- reflex agents with state



- An agent whose action is derived directly from an *internal mode of the current world state* that is updated over time.
  – keep track of the part of the world it can't see now to handle partial observability.

20

Continued.. Model-Based Reflex Agents

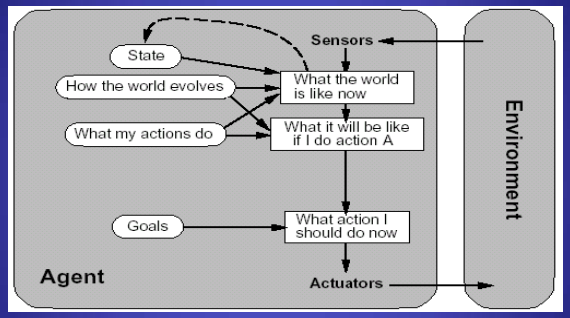
```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
              model, a description of how the next state depends on current state and action
              rules, a set of condition–action rules
              action,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

  state ← UPDATE-STATE(state, action, percept, model)
  rule ← RULE-MATCH(state, rules)
  action ← rule.ACTION
  return action
```

Figure 2.12 A model-based reflex agent. It 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.

- Maintain an ***internal state*** that *depends on the percept history* and thereby reflects at least some of the *unobserved aspects* of the current state
  -- handle partial observability of the environment.
- ***Internal state Update*** requires two knowledge:
  - information a/b how the world evolves independently of the agent
    -- used to keep track of the unobserved parts of the environment.
  - information a/b how the agent's own actions affect the world.
    i.e. how the world works – a ***model*** of the world.
- Interpret the new percept in the light of existing knowledge about the state.

21

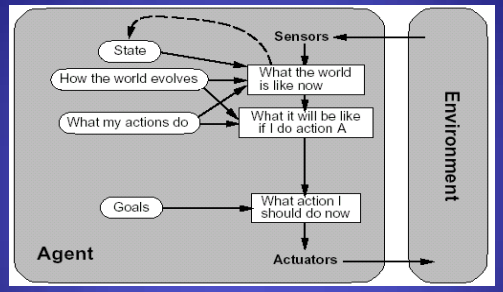
Goal-Based Agents



- An agent that selects actions that it believes will achieve explicitly represented ***goals.***

22

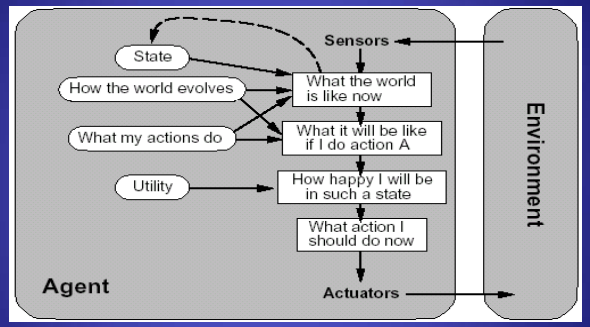
Continued.. Goal-Based Agents



- Need a goal information that describes desirable situations. E.g.) destination
- Goal based action selection:
  - goal information + the results of possible actions ⇒ select actions.
  - *search (chap 3-6)* & *planning (chap 11-12)* to find *action sequences*.
  - Consideration of the future.
- more flexible because the knowledge that supports its decisions is represented explicitly and can be modified.

23

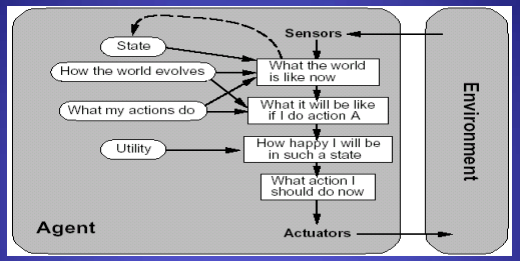
Utility-Based Agents



- An agent that selects actions that it believes will ***maximize*** expected utility of the outcome state.

24

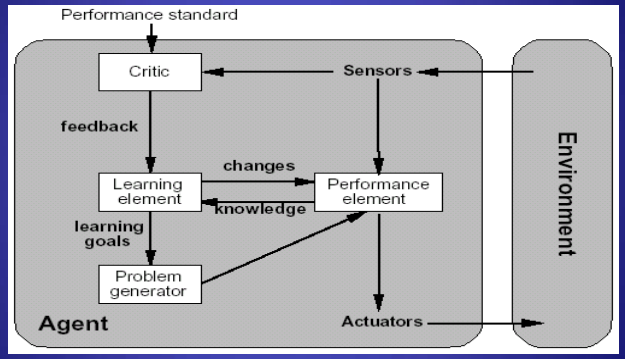
Continued.. Utility-Based Agents



- Measure *the degree of happiness* of different world state and their possible actions using performance measure – a high utility for the agent.
  - Choose actions that can maximize its own happiness.
  - -- the *high quality of state* or the *high degree of usefulness*.
- **Utility function**: state → real number
  - It allows rational decision in the cases such as:
    - Conflicting goals -- tradeoff
    - multi goals with no certain achievement – the likelihood of success
  - Partial observable and stochastic environment – the action of the maximum expected utility

25

Learning Agents



- An agent whose behavior *improves* over time based on its *experience.*

26

Continued.. Learning Agents

- Advantage:
  it allows the agent to operate in initially unknown environments and to become more competent than its initial knowledge alone might allow.
- 4 components:
  - ***Learning element***:
    - To make *improvements* on every part of the agent.
    - Its design depends on the design of the performance element.
    - use the feedback from the critic and determines how the performance element should be modified to do better in the future.
  - ***Performance element***:
    - To *select external actions*
  - ***Critic***:
    - To give the *learning element feedback* on the current performance of the agent
  - ***Problem generator***:
    - To suggest *exploratory actions* which will lead to new/informative experience.
    - Aim: *discover better actions for the long run* by trying some perhaps suboptimal actions in the short run, rather than keeping doing the best actions decided by the performance element, given what it knows.
- Example: An automated taxi

27

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 define task environments
- 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

28