INTELLIGENT AGENTS (IA)

Definition
An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators (effectors). Intelligent agents may also learn or use knowledge to achieve their goals. They may be very simple or very complex: a reflex machine such as a thermostat is an intelligent agent

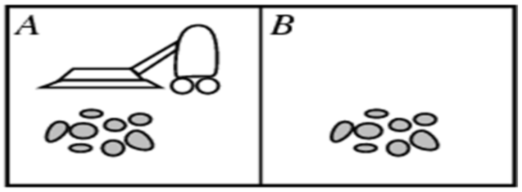
Human agent: eyes, ears, and other organs for sensors; hands, legs, mouth, and other body parts for actuators
Robotic agent: cameras and infrared range finders for sensors; various motors for actuators

Agents are the basic building blocks for applications, and applications are organized as networks of collaborating agents. Example: a desktop agent "recruits" the services of a screen and a connection agent to physically connect a call.

The notion of an agent is meant to be <u>a tool for analyzing systems</u>, not an absolute characterization that divides the world into agents and non-agents. Much like, e.g. object-oriented vs. imperative program design approaches.

Example

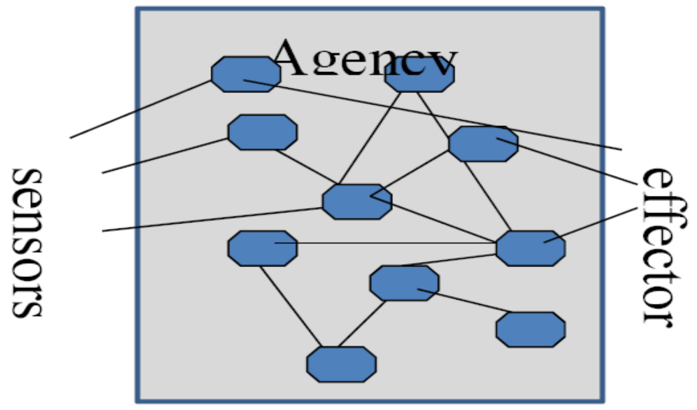
Vacuum-cleaner world



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

The need for agents
The following issues underline the need for agents:
Information overload. The managers have so much information and they need some kind of help to cope.
Massive bank of information over the years. A lot of information has been accumulated over the years and there is need to analyze it and discover any other knowledge held.
Internet. The Internet requires search tools.
Serv*ice support.* Service support is necessary in many areas including network security, electronic commerce or employee support.
Simplification of distributed computing. Agents can act as intelligent resource managers.
Overcome user interface problems. Agents act as personal assistants adapting to the users.
Handle information service management problem. Agents can provide services, service customization, monitor interacting features where systems are combined, enable using varied terminals, enable resource sharing and selection, diagnose problems, bill electronic users and provide security(firewalls).

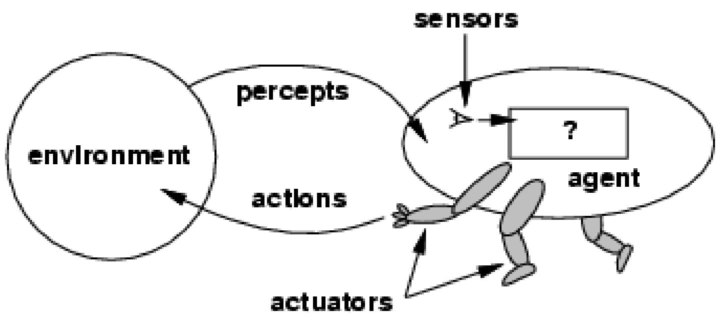
Intelligent Agents and Artificial Intelligence
Human mind as network of thousands or millions of agents all working in parallel. To produce real artificial intelligence, this school holds, we should build computer systems that also contain many agents and systems for arbitrating among the agents' competing results.



Challenges:

- Action selection: What next action to choose
- Conflict resolution

Agents and environments



The **agent function** maps from percept histories to actions: [*f*: P* → A]
The **agent program** runs on the physical **architecture** to produce *f* i.e. agent = architecture + program
Therefore, 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

Agent and conventional programs
Common properties that make agents different from conventional programs:

- Agents are **autonomous**, that is they act on behalf of the user
- Agents contain some level of **intelligence**, from fixed rules to learning engines that allow them to adapt to changes in the environment
- Agents don't only act **reactively**, but sometimes also **proactively**
- Agents have **social ability**, that is they communicate with the user, the system, and other agents as required
- Agents may also **cooperate** with other agents to carry out more complex tasks than they themselves can handle
- Agents may **migrate** from one system to another to access remote resources or even to meet other agents

Characteristics of intelligent agents
Intelligent agents have several characteristics that are discussed below.
Autonomy. Agents act or decide on their own. Sometimes they may do this to circumvent obstacles, or as they handle high level requests and seek more clarifications if necessary.
Background operation. Agents work in the background, usually out of sight, perhaps somewhere in the cyberspace.
Singularity of task. Agents work on a single task.

Communication. Agents interact with other agents or humans.
Automation of repetitive tasks. Agents work on special repetitive tasks.
Support conditional processing. Agents may be rule-based systems thereby showing flexibility.
Learning. Agents can learn; this goes beyond rule-based systems.
Reactivity. Agents can perceive the environment and then respond.
Proactiveness. Agents can take initiatives such as inhibiting behavior instead of just acting in response to environmental inputs.
Temporal continuity. Agents are continuously running processes.
Personality. Agents have a personality; they can interact with humans.
Mobility. Agents can move across different architectures.

Behavior and performance of IAs

- **Perception** (sequence) to **Action Mapping:** $f : P^* \rightarrow A$
    - ✓ **Ideal mapping:** specifies which actions an agent ought to take at any point in time
    - ✓ **Description:** Look-Up-Table vs. Closed Form
- **Performance measure:** a *subjective* measure to characterize how successful an agent is (e.g., speed, power usage, accuracy, money, etc.)
- (degree of) **Autonomy:** to what extent is the agent able to make decisions and actions on its own?

Structure of Intelligent Agents

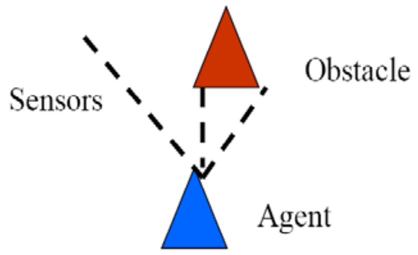
- Agent = architecture + program
- **Agent program:** the implementation of `f : P* → A,` the agent's perception-action mapping

```
function: Skeleton-Agent(Percept) returns Action
memory ← UpdateMemory(memory, Percept)
Action ← ChooseBestAction(memory)
memory ← UpdateMemory(memory, Action)
return Action
```

- **Architecture:** a device that can execute the agent program (e.g., general-purpose computer, specialized device, beobot, etc.)

Using a look-up-table to encode $f : P^* \rightarrow A$

- Example: Collision Avoidance
    - ✓ Sensors: 3 proximity sensors
    - ✓ Effectors/Actuators: Steering Wheel, Brakes



- How to generate: for each $p \in P_l \times P_m \times P_r$
  generate an appropriate action, $a \in S \times B$
- How large: size of table = #possible percepts times # possible actions = $|P_l|\,|P_m|\,|P_r|\,|S|\,|B|$
  E.g., $P = \{\text{close, medium, far}\}^3$
  $A = \{\text{left, straight, right}\} \times \{\text{on, off}\}$
  then size of table = 27*3*2 = 162
- How to select action? Search.

The following is an algorithm that describes an agent function that implements a table-driven 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 ← LOOKUP (percepts, table)
Return action
```

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

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

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.
Rationality is distinct from omniscience (all-knowing with infinite knowledge)
Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration)
An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt)

PEAS (Performance measure, Environment, Actuators, Sensors)
Must first specify the setting for intelligent agent design

Example
Consider the task of designing the following agents:

i). 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

ii). 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)

iii). 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

iv). 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.
- **Deterministic (vs. stochastic):** 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.
- **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):** A limited number of distinct, clearly defined percepts and actions.
- **Single agent (vs. multiagent):** An agent operating by itself in an environment.

Example

	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	No	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

TYPES OF AGENTS
Agents can be grouped into five classes based on their degree of perceived intelligence and capability:

i). simple reflex agents
ii). model-based reflex agents
iii). goal-based agents
iv). utility-based agents
v). learning agents

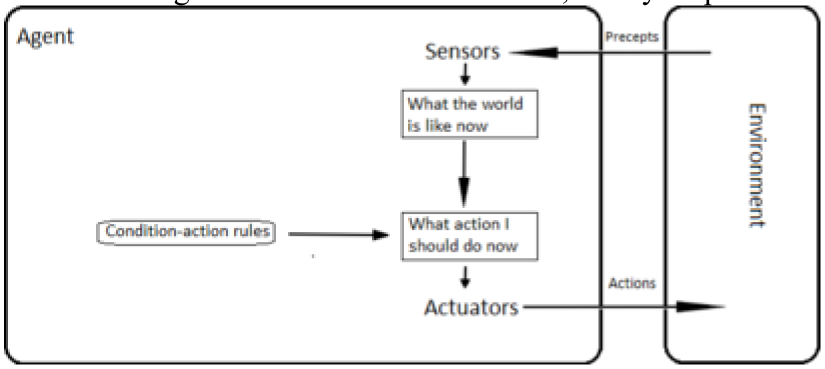
i). Simple reflex agents

Simple reflex agents act only on the basis of the current percept, ignoring the rest of the percept history. The agent function is based on the *condition-action rule*: if condition then action.

This agent function only succeeds when the environment is fully observable. Some reflex agents can also contain information on their current state which allows them to disregard conditions whose actuators are already triggered.

Infinite loops are often unavoidable for simple reflex agents operating in partially observable environments.

Note: If the agent can randomize its actions, it may be possible to escape from infinite loops.



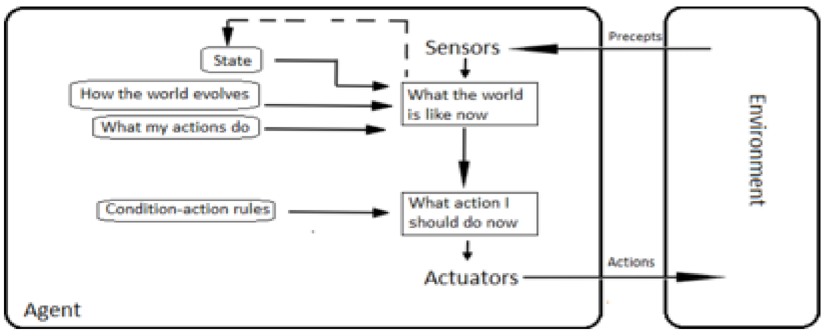
```
function SIMPLE-REFLEX-AGENT( percept) returns action
    static: rules, a set of condition-action rules
    state       INTERPRET-INPUT( percept)
    rule        RULE-MATCH(state, rules)
    action      RULE-ACTION[rule]
    return action
```

Generally, a simple reflex agent 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.

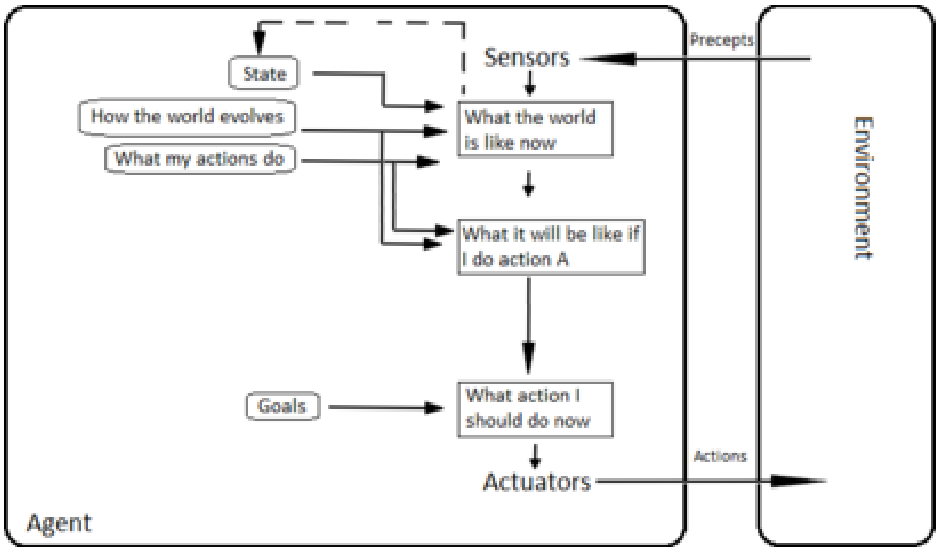
ii). Model-based reflex agents

A model-based agent can handle a partially observable environment. Its current state is stored inside the agent maintaining some kind of structure which describes the part of the world which cannot be seen. This knowledge about "how the world works" is called a model of the world, hence the name "model-based agent".

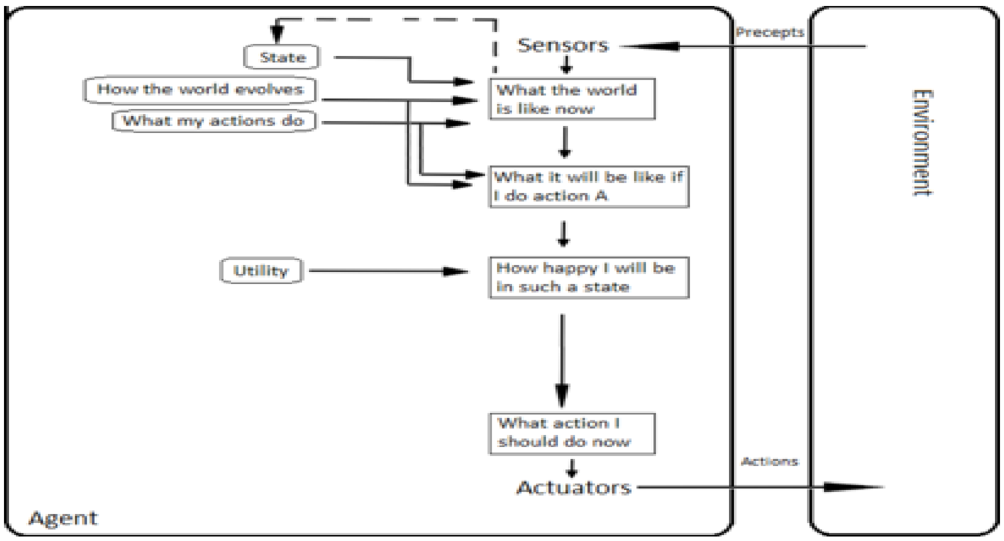
A model-based reflex agent should maintain some sort of internal model that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state. It then chooses an action in the same way as the reflex agent.



iii). Goal-based agents
Goal-based agents further expand on the capabilities of the model-based agents, by using "goal" information. Goal information describes situations that are desirable. This allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state. Search and planning are the subfields of artificial intelligence devoted to finding action sequences that achieve the agent's goals.
In some instances the goal-based agent appears to be less efficient; it is more flexible because the knowledge that supports its decisions is represented explicitly and can be modified.



iv). 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. A more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent. The term utility, can be used to describe how "happy" the agent is.
A rational utility-based agent chooses the action that maximizes the expected utility of the action outcomes- that is, the agent expects to derive, on average, given the probabilities and utilities of each outcome. A utility-based agent has to model and keep track of its environment, tasks that have involved a great deal of research on perception, representation, reasoning, and learning.

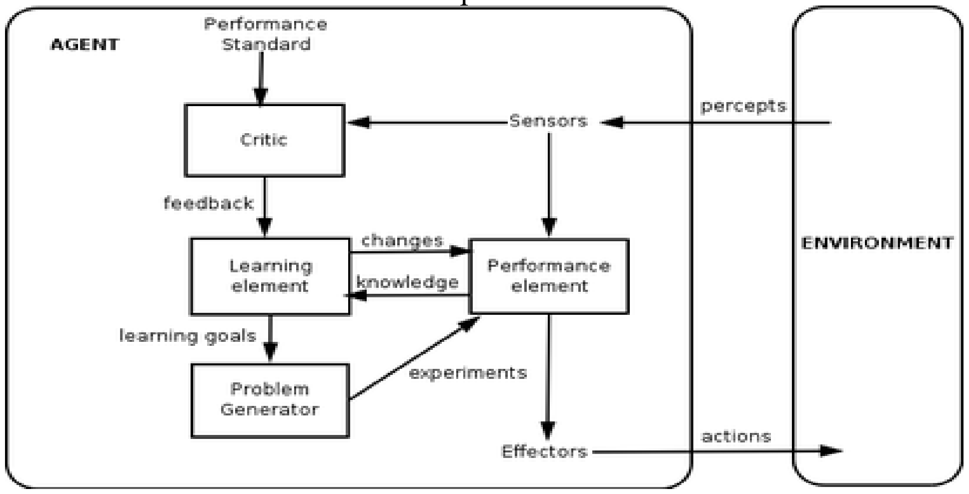

Precepts
Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Utility
How happy I will be in such a state
What action I should do now
Actuators
Actions
Agent
Environment

v). Learning agents

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. The most important distinction is between the "learning element", which is responsible for making improvements, and the "performance element", which is responsible for selecting external actions.

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. The performance element is what we have previously considered to be the entire agent: it takes in percepts and decides on actions.

The last component of the learning agent is the "problem generator". It is responsible for suggesting actions that will lead to new and informative experiences.



Exercise

Consider a snail enclosed in cuboid. The cuboid has got only one hole at the bottom corner. The snail starts from the centre of the cuboid at the base and seeks the hole by moving forward until it encounters an edge. It then turns and moves left up to the next corner where it seeks the hole. If it finds the hole it escapes and if it doesn't it turns to the left edge again and continues moving. This is repeated until it obtains the hole and escapes. Assuming the reasoning and actions of the snail are implemented as a table driven agent, show

a) A suitable algorithm than implements the snail's actions depicted in a percepts-action table until escape of snail.
b) Show how the same agent in (i) above can be implemented using a simple reflex agent's architecture.