Chapter 2

The equation: **agent = architecture + program**
agent = hardware + software

agent program: that implements the agent function mapping percepts to actions. This program will run on some sort of computing device(**architecture).**

The architecture : makes percepts from sensors available to the program,
runs the program, and
feeds the program's action choices to the actuators.

- **Ideal Rational Agent (does the right thing)**
  For each possible **percept** (input) sequence,
  an ideal rational agent should do whatever **action(output)**
  expected to **maximize its performance** measure,
  on the basis of built-in **knowledge** the agent has.
- An **omniscient** agent **knows the *actual* outcome of its actions**, and can act accordingly; but omniscience is impossible in reality.
  Example: When you use a glove to hold **a hot plate** in the kitchen. This was because you were omniscient in this situation.
- If the agent's actions are based completely on built-in knowledge, we say that the agent lacks **autonomy.** A system is **autonomous** to the extent that its behavior is determined its own **experience**.

PEAS

Agent Type	Performance measure	Environmnt	Actuators	Sensors
Taxi driver	Safe, Fast, Legal, Comfort, Maximize Profits	Roads, Other traffics, Pedestrian, Customers	Steering, Accelerate,Brake, Signal, Alarm, Display	Cameras, Speedmeter, Odometer, sensors, Keyboard
Medical diagnosis system	Healthy patient, minimize costs,	Patient, hospital, Staff	Questions, tests, diagnoses, Treatments	Keyboard entry of symptoms and answers
Interactiv English tutor	Maximize student's score on test	Set of students, testing agency	Print exercises, suggestion, correction	Keyboard Entry
Mathematic teacher	Maximize student's score on test	students, testing agency, Books	Printer, display, Alarm Hand	Keyboard Entry, …..
Football player	Maximize team's number of goals	Stadium Players, Ball , audience	Hand , head, arm, leg	Camera ….

Specifying the environment (ODESA D)

1- Observability (Fully Observable vs. Partial Observable)

The environment is **fully observable** to that agent, if an agent's sensors give it access to the complete state of the environment, else the environment is **partially observable**.

2- Deterministic vs. stochastic.

The environment is **deterministic**, if the **next state** of the environment is completely determined by the **current state** and **actions** executed by the agents, else it is **stochastic** or **ghosty.**

It is **strategic**, if it is deterministic except for actions of other agents.

3- Episodic vs. Sequential.

In an **episodic** environment, the agent's experience is divided into "episodes." Each episode consists of the agent perceiving and then performing a single action, else it is **Sequential.**

4- Static vs. dynamic.

If the environment can change while an agent is deliberating , then we say the environment is **dynamic** for that agent; otherwise it is **static**.

If the environment does not change with the passage of time but the agent's *performance score* does, then we say the environment is **semi-dynamic**, **Exam** environment is example.

5- Agents (Single agent vs. multi-agent)

Distinction between single and multi-agent is simple.

In **competitive** multi-agent environment, trying to maximize the performance measure of the opponent entity B, will minimize the agent A's performance measure, else it is **cooperative**.

Taxi driving is **partially cooperative** environment (avoiding collisions) and **partially competitive** (only one car can occupy parking space). Any **team game** is partially.

6- Discrete (Discrete vs. Continous)

Any board game is **discrete**.

Examples

Crossword puzzle: Fully Observable, Deterministic, Sequential, Static, Single Agent
Chess with a clock: Fully Observable, Strategic, Sequential, Semi-dynamic, Multi
Taxi driving: Partially Observable, Stochastic, Sequential, Dynamic, Multi Agent
Medical diagnoses: Partially Observable, Stochastic, Sequential, Dynamic, Single
Image-analysis: Fully Observable, Deterministic, Episodic, Semi-dynamic, Single
Part-picking robot: Partially Observable, Stochastic, Episodic, Dynamic, Single
Refinery controller: Partially Observable, Stochastic, Sequential, Dynamic, Single
Interactive English tutor: Partially Observable, Stochastic, Sequential, Dynamic, Multi Agent

Types of agent program:

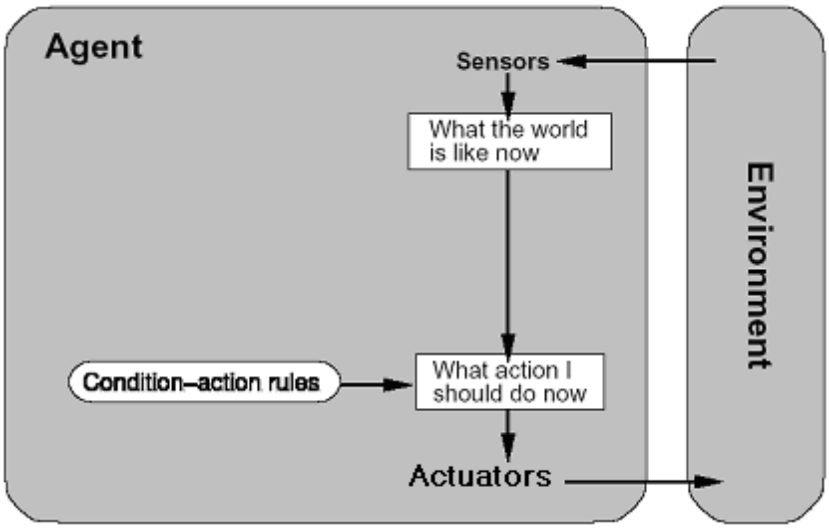
There are four basic types of agents, that are :

- Simple reflex agents
- Model-based reflex agent
- Goal-based agents
- Utility-based agents

1- Simple reflex agents

Agents select actions on the basis of the **current percept** ignoring the rest of the percept history, i.e. **environment is fully observable**. Agents make connection from percepts to actions through **condition-action rules** written as :

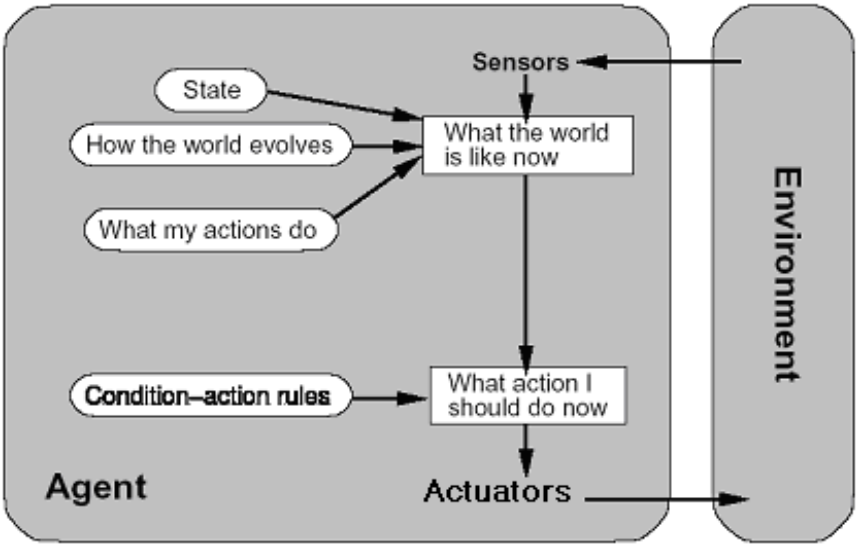
if car-in-front-is-braking **then** initiate-braking



2-Model-based reflex agent

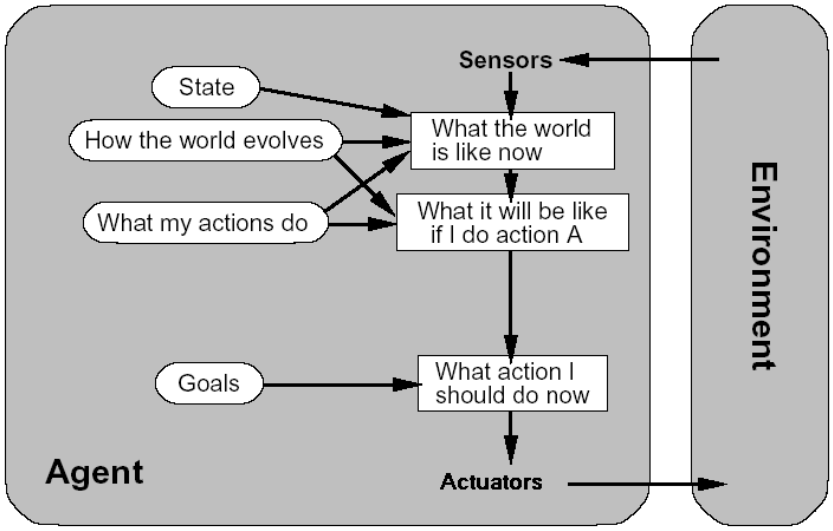
The agent should maintain **internal state** in order to choose an action. Updating state requires two kinds of knowledge (**MODEL**):

- Information about **how world evolves** independently of agent
- Information about how agent's own **actions affect the world**



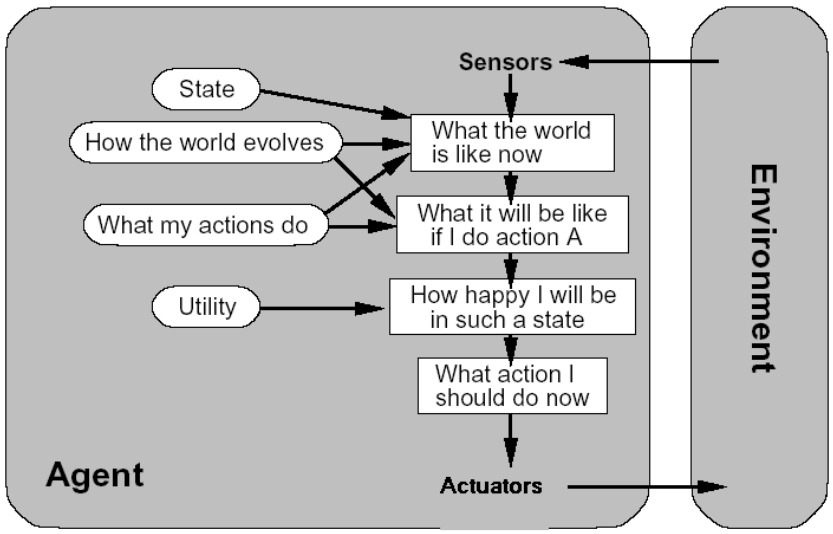
3-Goal-based agents

Search and **planning** are the subfields of AI devoted to finding action sequences that do achieve the agent's goals. goal-based agent appears **less efficient**, it is far **more flexible**.



4-Utility-based agents

When there are **conflicting goals,** e.g., speed and safety. **Utility** is a function that describes the **degree of happiness**.

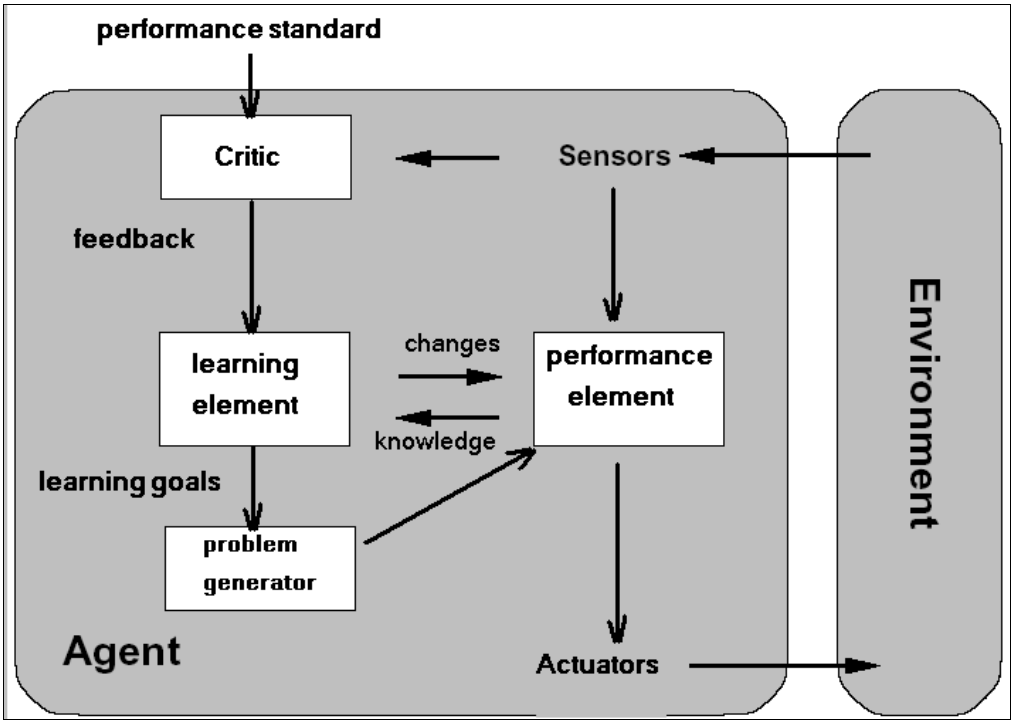


Algorithm of GENERAL-AGENT

function GENERAL-AGENT(*percept*) **returns** *action*
static:
state, a description of the current world state
rules, a set of condition-action rules
action, the most recent action
state ← INTERPRET-INPUT(*percept*)
rule ← RULE-MATCH(*state*, *rules)*
action ← RULE-ACTION[*rule*]
return *action*

Learning agent can be divided into four conceptual components :

1. **learning element:** responsible for making changes
2. **performance element :** responsible for experimental actions
3. **critic** : takes a fixed performance standard, gives feedback to the learning element on how performance element modified to do better in future.
4. **problem generator** : responsible for suggesting exams/problems that will lead to new experiences



A pattern classification system is expected to perform

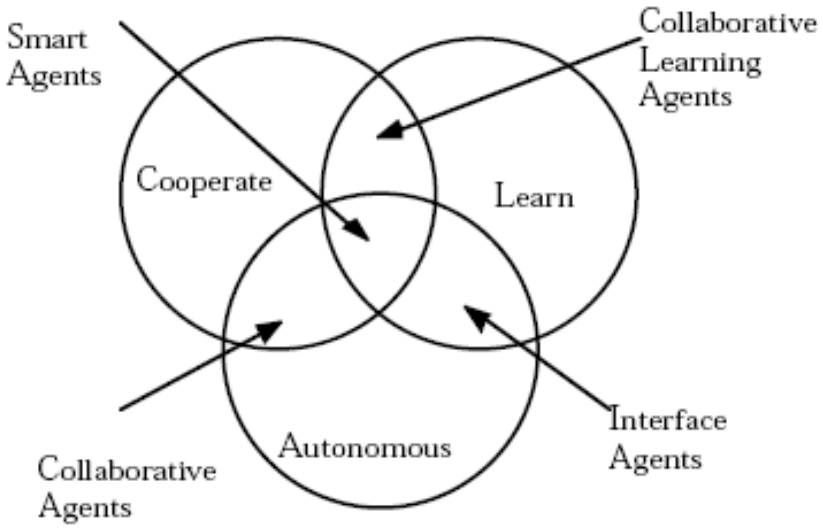
(1) **supervised** classification, where a given pattern has to be identified as a member of already known or defined classes;

(2) **unsupervised** classification or **clustering**, where a pattern is assigned to a so far unknown class of patterns.

A common example for **mobile** agent is **virus**.

•Property	•Common to all agents?	•Description
•Autonomous	•Yes	•Can act on its own
•Reactive	•Yes	•Responds timely to changes in its environment
•Proactive	•Yes	•Initiates actions that affects its environment
•Communicative	•Yes	•Can exchange information with users and other agents
•Continuous	•No	•Has a relatively long lifespan
•Mobile	•No	•Can migrate from one site to another
•Adaptive	•No	•Capable of learning

Mobile agent as Venn diagram.

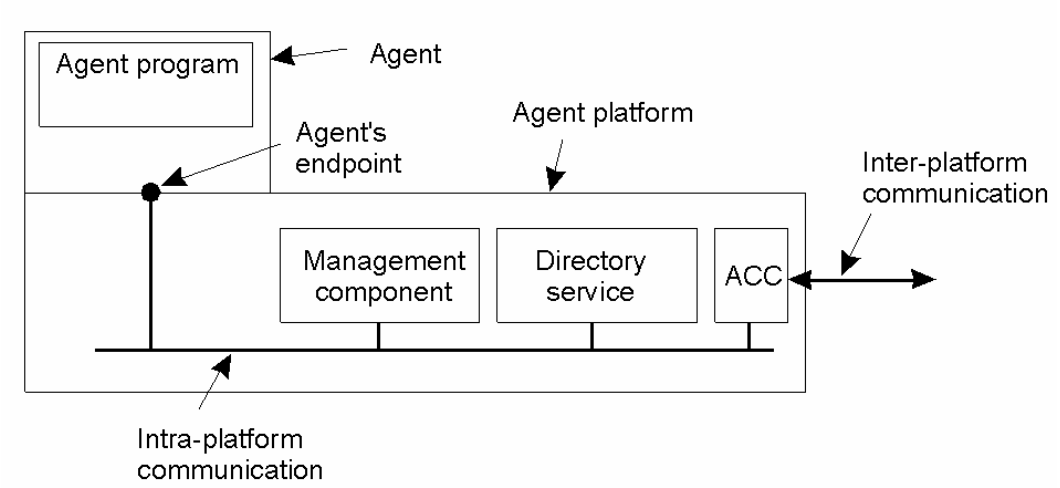


D' Agent mobile agent technology: Foundation for intelligent physical agents
(FIPA) develops general model for software agents.

Agent communication chanel (ACC) is responsible for inter-agent communication.

Management component facilities are to create,delete, look up current end point agent.

Discovery service provides description of services.



Weak mobility	strong mobility
▪ transfer only code segment with initialization of data ▪ example: java applet ▪ simplicity that target machine can execute that code	▪ transfer execute segment, so the running process can be stopped, moved to another machine then resume execution ▪ example: D'Agent ▪ powerful but harder to implement

Flow chart of machine learning for pattern recognition.