Lesson 2

Intelligent Agents

Artificial Intelligence

- A branch of computer sciencce dealing with the simulation of intelligent behaviour in computers.
- The capablity of a machine to imitate intelligent human behaviour.

AI strategies

- Think as a human: the **cognitive model**.
- **Rational thinking:** to draw justifable concludions from data, rules, and logic.
- Rational acting: **intelligent agents**.

Intelligent Agents

An **agent** is anything that can be viewed as **perceiving** its environment through **sensors** and **acting** upon that environment through **effectors**.

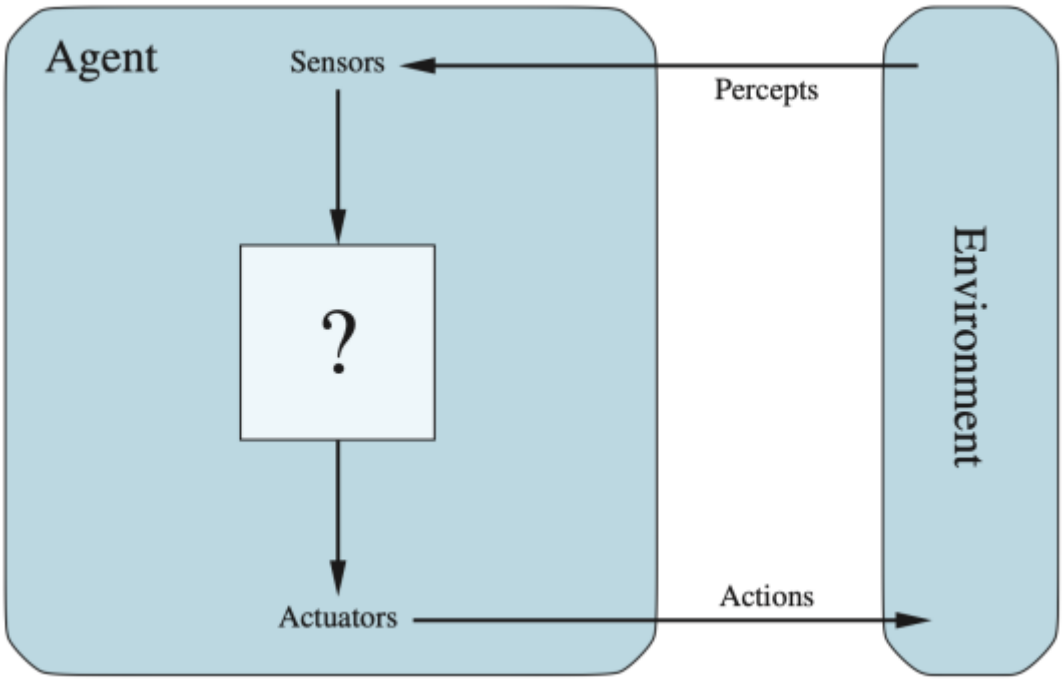
A **rational agent** is one that does the "right thing". We use the term *performance measure* for the how the criteria that determine how successful an agent is.

A definition of **artificial intelligence research:** "The study and design of rational agents"

Agent function

The agent function maps from perception to actions: f: P → A The agent program runs on the physical architecture to produce **f**
Agent = Architecture + Program

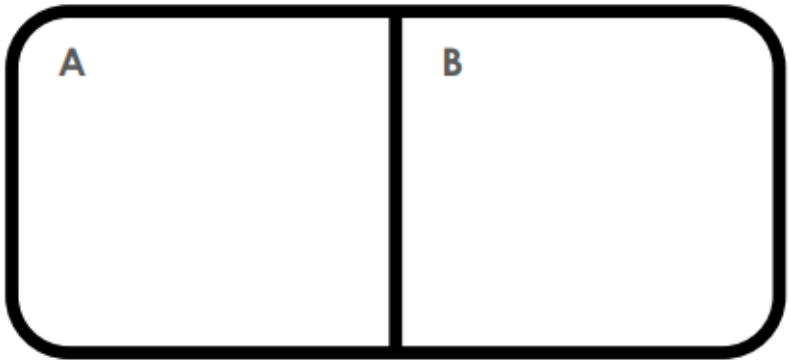
Agent Interaction



Ex: The vaccum-cleaner agent

A vacuum-cleaner world with just two locations. Each location can be clean or dirty, and the agent can move left or right and can clean the square that it occupies:

- Percepts: location, contents – e.g., $[$ A, dirty $]$
- Actions: ${$ left, right, suck, noOp $}$



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*
⋮	⋮
$[A, Clean], [A, Clean], [A, Clean]$	*Right*
$[A, Clean], [A, Clean], [A, Dirty]$	*Suck*
⋮	⋮

Partial tabulation of a simple agent function for the vacuumcleaner world. The agent cleans the current square if it is dirty, otherwise it moves to the other square.

Performance measure

An objective criterion for success of an agent's behaviour (“cost”, “reward”, “utility”, “sustainable”...).

E.g., performance measure of a vacuum-cleaner?

- amount of time taken
- amount of dirt cleaned up
- amount of electricity consumed
- world cleaned?
- ...

Rational agents

Rational Agent: For each possible percept sequence, a rational agent should select an action that is expected to *maximize* its performance measure, based on 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 perceptions so as to obtain useful information (exploration).
An agent is **autonomous** if its behaviour is determined by its own perceptions & experience (with ability to **learn** and adapt) without depending solely on build- in knowledge.

Task environment

To design a rational agent, we must specify the task environment **P.E.A.S.**:

- Performance measure
- Envirnoment conditions
- Actions allowed
- Sensors for perception

Example table:

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments	Touchscreen/voice entry of symptoms and findings
Satellite image analysis system	Correct categorization of objects, terrain	Orbiting satellite, downlink, weather	Display of scene categorization	High-resolution digital camera
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, tactile and joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, raw materials, operators	Valves, pumps, heaters, stirrers, displays	Temperature, pressure, flow, chemical sensors

Environment types I

- 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): An agent’s action is divided into atomic episodes. Decisions do not depend on previous decisions/actions.

Environment types II

- 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 perceptions and actions. How do we represent or abstract or model the world?
- Single agent (vs. multi-agent): an agent operating by itself in an environment. Does the other agent interfere with my performance measure?

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Environment types III

- Ideal environment:
  - Fully
  - Deterministic
  - Episodic
  - Static
  - Discrete
  - Single
- Real environment:
  - Partially
  - Stochastic
  - Sequential
  - Dynamic
  - Continuous
  - Multi-agent

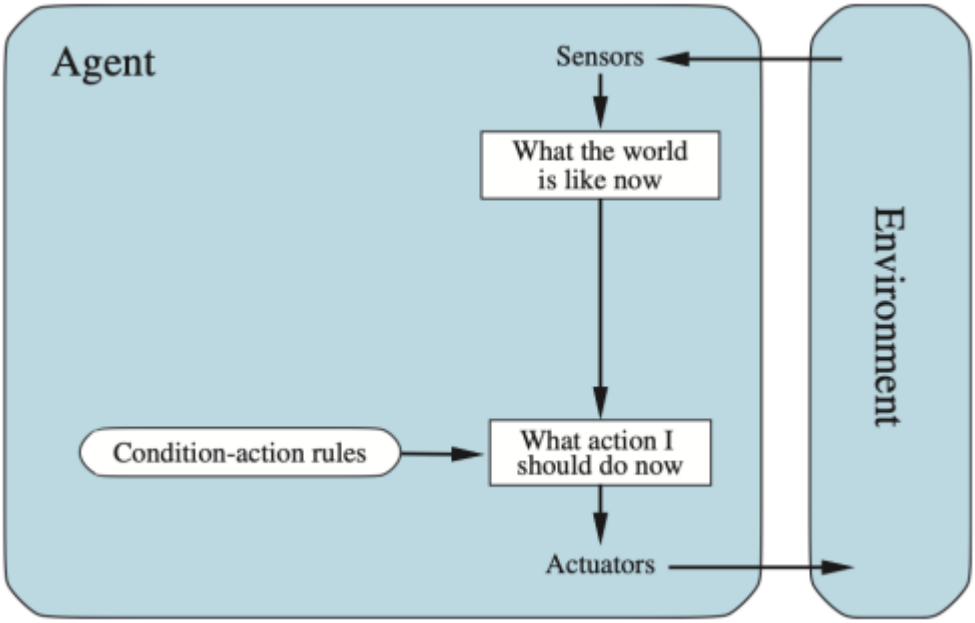
Agent Types

Five basic types in order of increasing generality:

- Table Driven agents
- Simple reflex agents
- Model-based reflex agents
- Goal-based agents

- Utility-based agents

Reflex agent



We use rectangles to denote the current internal state of the agent's decision process, and ovals to represent the background information used in the process.

---

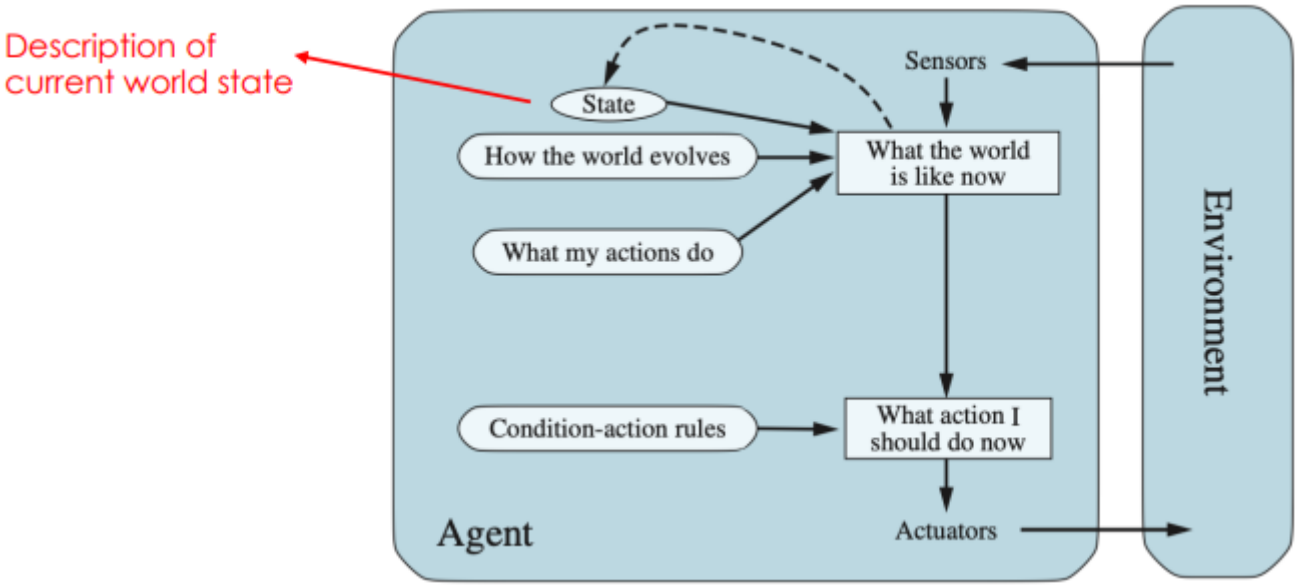
function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *rules*, a set of condition–action rules

  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.

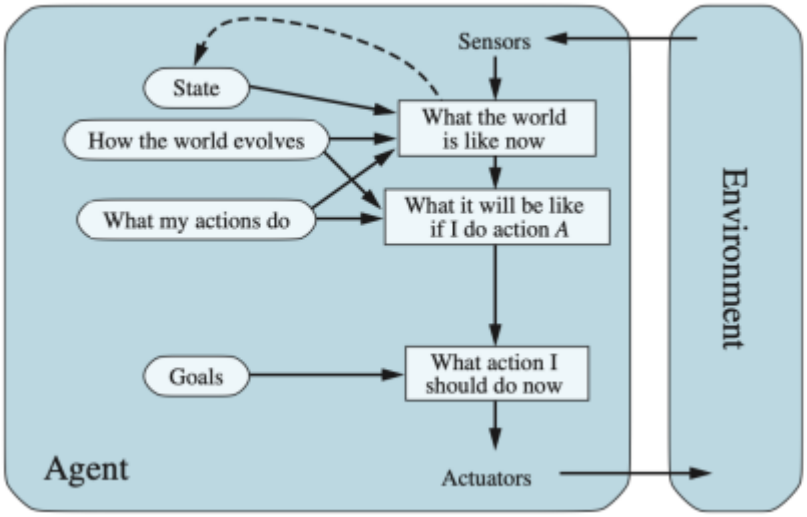
Will work only if the correct decision can be made on the basis of only the current percept—that is, only if the environment is fully observable.

Model-based reflex agent



Model the state of the world by: modelling how the world changes how its actions change the world. This can work even with partial information.

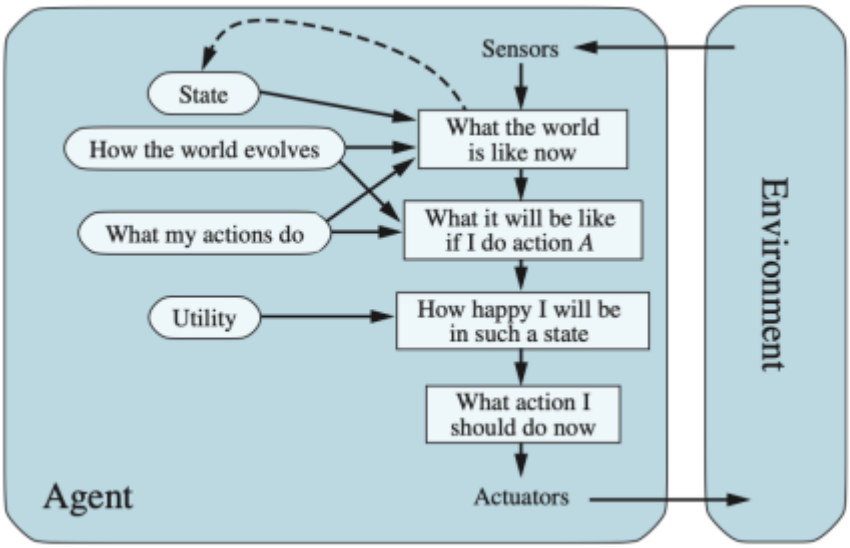
Model-based, Goal-based agent



Some solutions to goal states are better than others. Which one is best is given by a utility function. Which combination of goals is preferred?

It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.
We need to predict: plan & search.

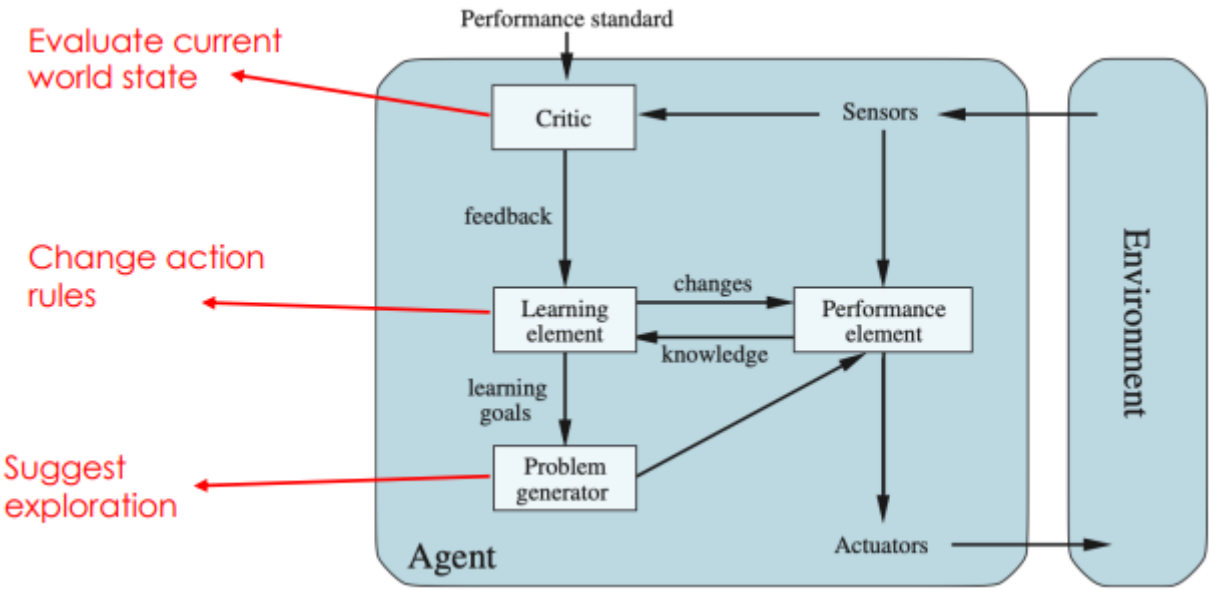
Model-based, utility-based agent



How does an agent improve over time?
By monitoring it's performance and suggesting better modeling, new action rules, etc.

It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.

General learning agent



The "performance element" box represents what we have previously considered to be the whole agent program. Now, the "learning element" box gets to modify that program to improve its performance.

Summary

- **Agent** is anything that can be viewed as **perceiving** and **acting**
- **Performance measurement** evaluates the agent behaviour, and a **rational agent** maximizes this measure
- **Environment** types and problem description (P.E.A.S.)
- The **Agent program**: Reflex, Model-based, Goal-based, Utilitybased and General (learning)