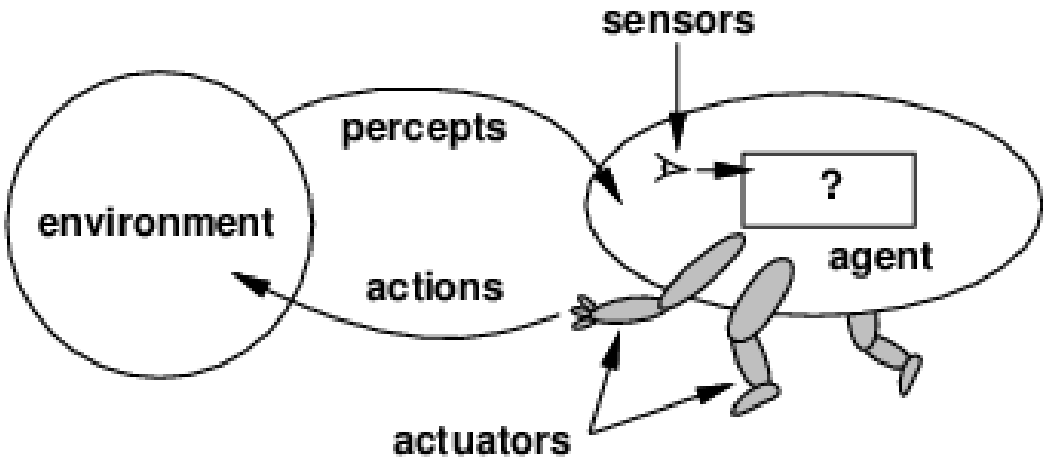
Intelligent Agents: acting rationally

AIMA chapter 2

Summary

- ♢ Agents and environments
- ♢ Rationality
- ♢ PEAS (Performance measure, Environment, Actuators, Sensors)
- ♢ Environment types
- ♢ Agent types

Agents and environments

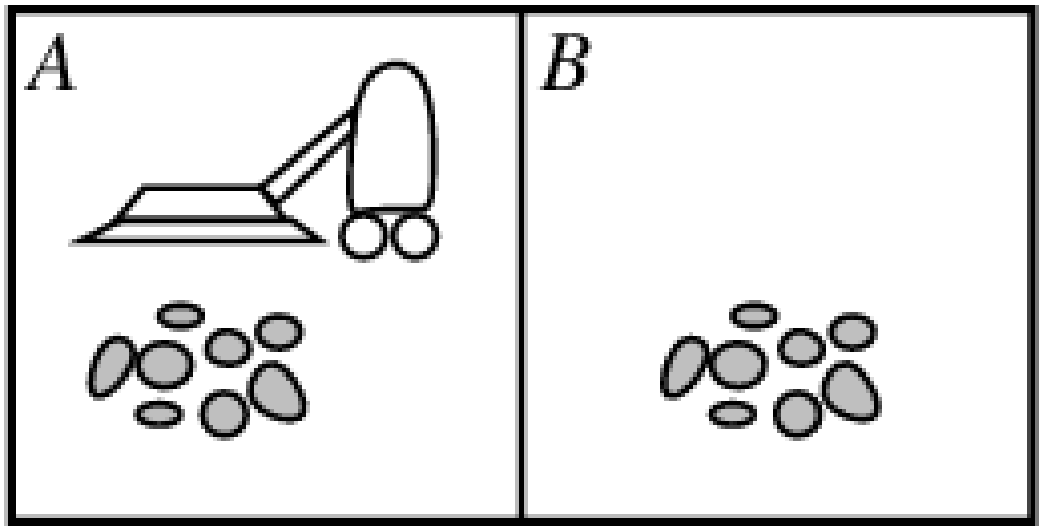


Agents include humans, robots, softbots, thermostats, etc.
The agent function maps from percept histories to actions:

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

The agent program runs on the physical architecture to produce f

Example: Vacuum-cleaner world



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

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*
⋮	⋮

What is the **right** function?

Can it be implemented in a **small** agent program?

Agent Programs vs. Agent Functions

Question

If an agent has $|\mathcal{P}|$ possible perceptions, how many entries will the agent function have after T time steps ?

Agent Programs vs. Agent Functions

Question

If an agent has $|\mathcal{P}|$ possible perceptions, how many entries will the agent function have after T time steps ?

Sol

$\sum_{t=1}^{T} |\mathcal{P}|^t$

AI goal $\Rightarrow$ **Design small agent programs to represent huge agent functions**

A possible agent program

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*

Rationality

Fixed performance measure evaluates the environment sequence
– one point per square cleaned up in time T?
– one point per clean square per time step, minus one 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 $\neq$ omniscient
– percepts may not supply all relevant information

Rational $\neq$ clairvoyant
– action outcomes may not be as expected

Hence, rational $\neq$ successful

Rational $\Longrightarrow$ exploration, learning, autonomy

Rationality of the vacuum cleaner agent

Exercise

Consider the following hypothesis for the vacuum cleaner world:

- Performance measure awards 1 point per clean square per time step
- Map is known a-priori
- Environment is static (clean squares remain clean, dirty squares remain dirty if not cleaned)
- Actions and perceptions are correct and accurate

Show that the agent function defined above is indeed rational.

Multi-Robot Patrolling

Exercise

Consider the following environment:

- Three rooms (A,B,C) and two robots (r_1,r_2)
- r_1 can patrol A and B, r_2 can patrol B and C
- r_1 starts from A and r_2 starts from C
- travel time between rooms is zero
- Performance measure: minimise sum of all rooms' average Idleness
- Average idleness = sum of time interval for which the room is not visited by any robot / total time interval

What would be a rational behavior for this 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??

PEAS

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

Consider, e.g., the task of designing an automated taxi:

Performance measure?? safety, destination, profits, legality, comfort, ...

Environment?? city streets/freeways, traffic, pedestrians, weather, ...

Actuators?? steering, accelerator, brake, horn, speaker/display, ...

Sensors?? video, accelerometers, gauges, engine sensors, keyboard, GPS, ...

Internet shopping agent

Performance measure??
Environment??
Actuators??
Sensors??

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)

Environment types

	Crossword	Backgammon	e-shopping	Taxi
Observable??				
Deterministic??				
Episodic??				
Static??				
Discrete??				
Single-agent??				

Environment types

	Crossword	Backgammon	e-shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??				
Episodic??				
Static??				
Discrete??				
Single-agent??				

Environment types

	Crossword	Backgammon	e-shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??				
Static??				
Discrete??				
Single-agent??				

Environment types

	Crossword	Backgammon	e-shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??				
Discrete??				
Single-agent??				

Environment types

	Crossword	Backgammon	e-shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Yes	Semi	No
Discrete??				
Single-agent??				

Environment types

	Crossword	Backgammon	e-shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Yes	Semi	No
Discrete??	Yes	Yes	Yes	No
Single-agent??				

Environment types

	Crossword	Backgammon	e-shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Yes	Semi	No
Discrete??	Yes	Yes	Yes	No
Single-agent??	Yes	No	Yes*	No

* = except auctions

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Agent programs

agent = architecture + **program**

General Skeleton for a program:

- input: **current** perception
- output: **next** action

Note: everything else that is important to decide next action must be stored/computed by the agent.

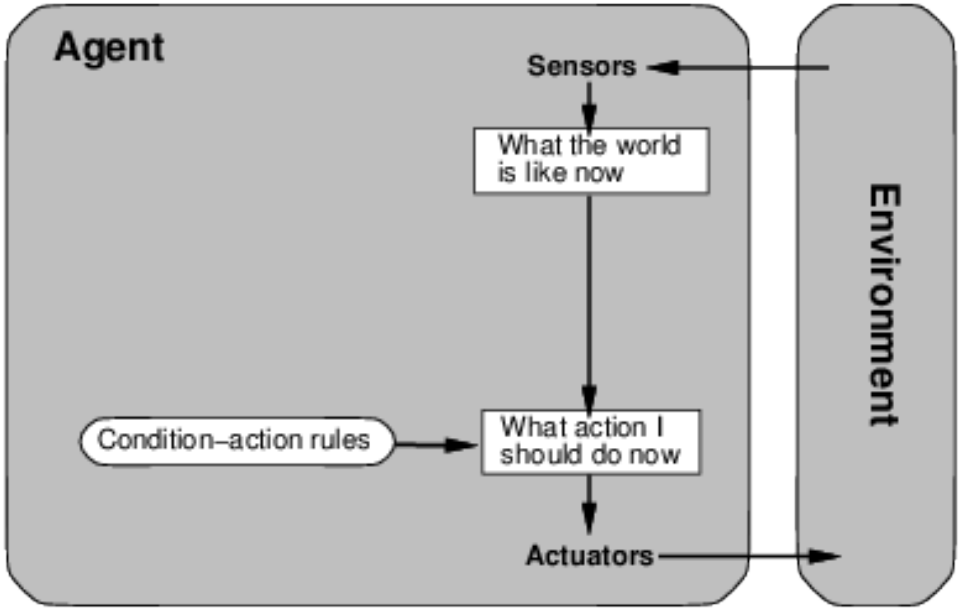
Agent types

Four basic types in order of increasing generality:

- simple reflex agents
- reflex agents with state
- goal-based agents
- utility-based agents

All these can be turned into learning agents

Simple reflex agents

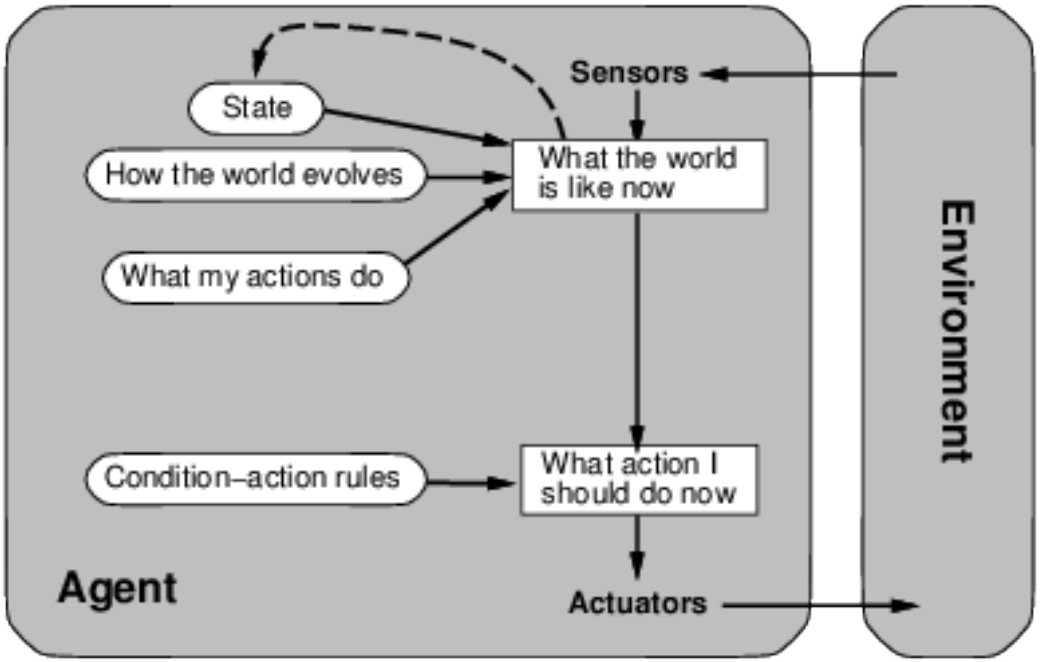


Example

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*

Reflex agents with state



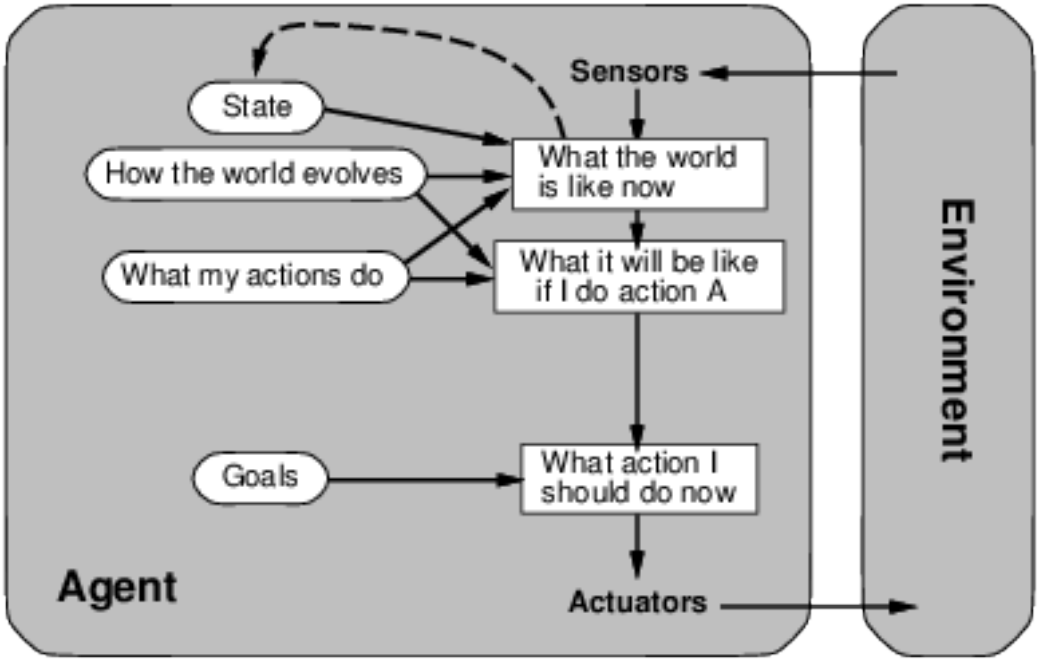
Usually called **model based** agents

Example

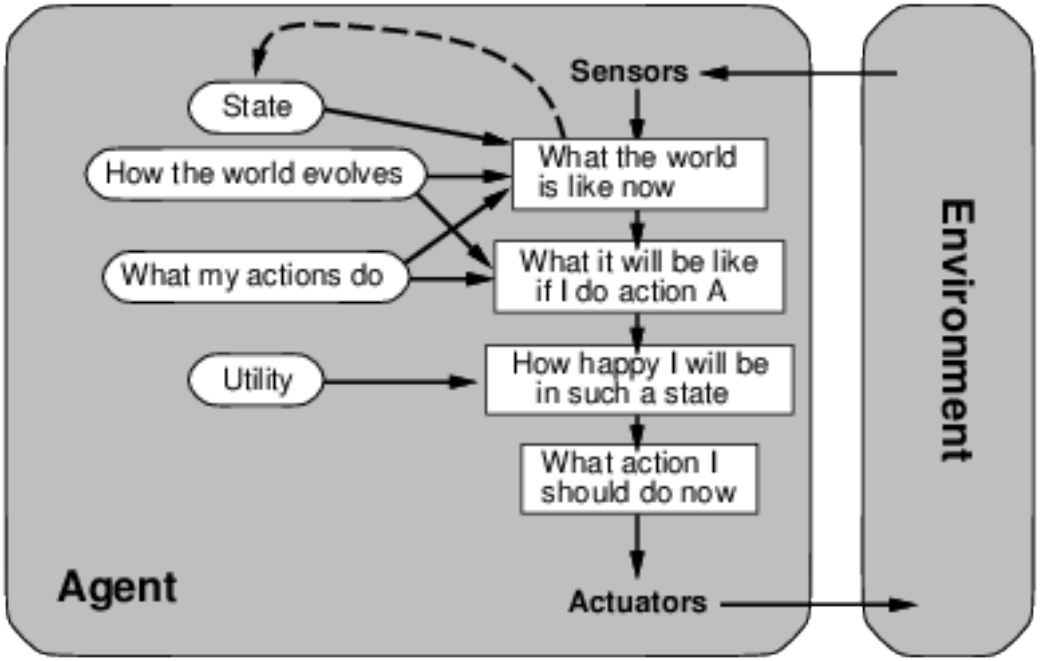
function Reflex-Vacuum-Agent([*status*]) **returns** an action
static: *current-location = initial-location,*
current-action = none,
next-action = current-action,

current-location = Update-State(current-location,current-action)
if *status = Dirty* **then** *next-action = Suck*
else if *current-location = A* **then** *next-action = Right*
else if *current-location = B* **then** *next-action = Left*
current-action = next-action
return *current-action*

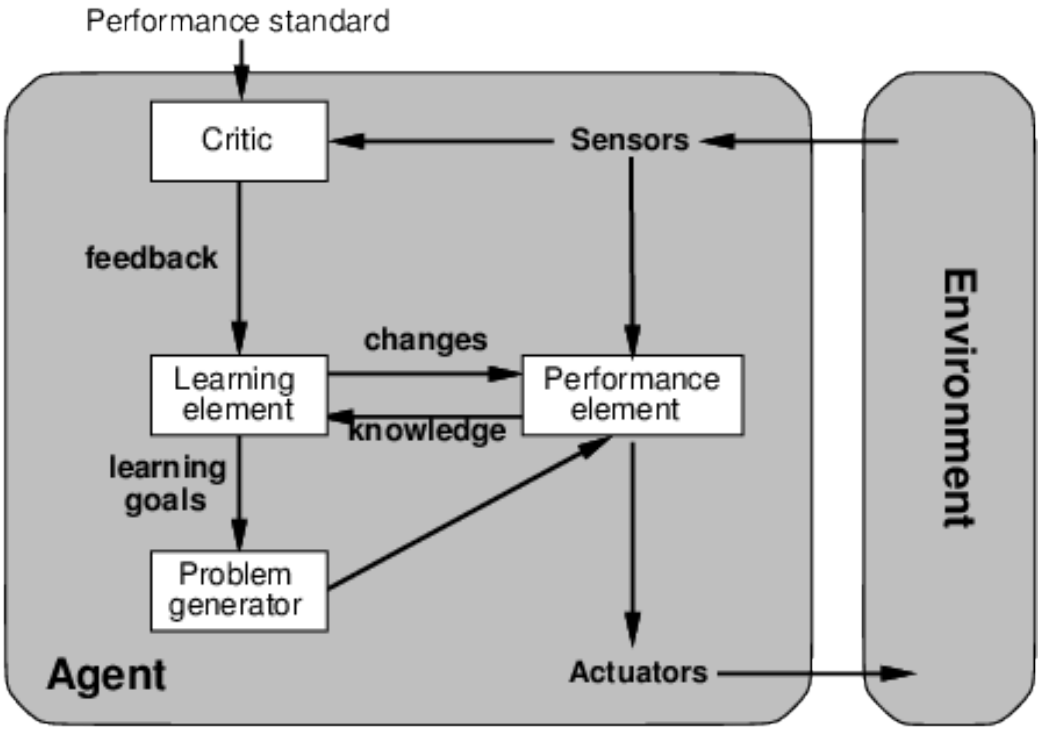
Goal-based agents



Utility-based agents



Learning agents



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