

Intelligent agents

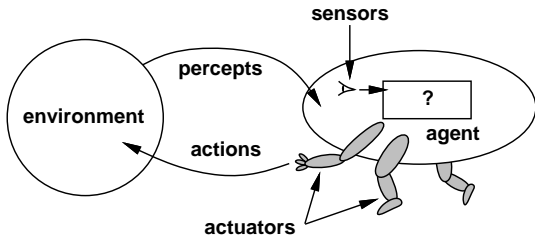
As seen from Russell & Norvig perspective

Slides from Russell & Norvig book, revised by Andrea Roli

Outline

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

Agents and environments

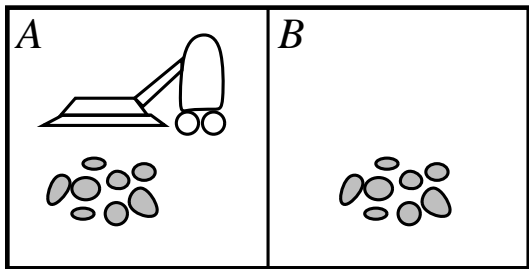


Agents include humans, robots, softbots, thermostats, etc.
The **agent function** maps from percept histories to actions:

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

The **agent program** runs on the physical **architecture** to produce f

Vacuum-cleaner world



Percepts: location and contents, e.g., [A, *Dirty*]

Actions: *Left*, *Right*, *Suck*, *NoOp*

A vacuum-cleaner agent

Percept sequence	Action
<i>[A, Clean]</i>	<i>Right</i>
<i>[A, Dirty]</i>	<i>Suck</i>
<i>[B, Clean]</i>	<i>Left</i>
<i>[B, Dirty]</i>	<i>Suck</i>
<i>[A, Clean], [A, Clean]</i>	<i>Right</i>
<i>[A, Clean], [A, Dirty]</i>	<i>Suck</i>
<i>⋮</i>	<i>⋮</i>

A vacuum-cleaner agent algorithm

```
function REFLEX-VACUUM-AGENT([location,status]) re-  
turns 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?

Can it be implemented in a small agent program?

Rationality

- Fixed **performance measure** evaluates the
 - 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 \implies exploration, learning, autonomy

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??

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?? US 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

	Solitaire	Backgammon	Internet shopping	Taxi
<u>Observable??</u>				
<u>Deterministic??</u>				
<u>Episodic??</u>				
<u>Static??</u>				
<u>Discrete??</u>				
<u>Single-agent??</u>				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
<u>Observable??</u>	Yes	Yes	No	No
<u>Deterministic??</u>				
<u>Episodic??</u>				
<u>Static??</u>				
<u>Discrete??</u>				
<u>Single-agent??</u>				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
<u>Observable??</u>	Yes	Yes	No	No
<u>Deterministic??</u>	Yes	No	Partly	No
<u>Episodic??</u>				
<u>Static??</u>				
<u>Discrete??</u>				
<u>Single-agent??</u>				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
<u>Observable??</u>	Yes	Yes	No	No
<u>Deterministic??</u>	Yes	No	Partly	No
<u>Episodic??</u>	No	No	No	No
<u>Static??</u>				
<u>Discrete??</u>				
<u>Single-agent??</u>				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
<u>Observable??</u>	Yes	Yes	No	No
<u>Deterministic??</u>	Yes	No	Partly	No
<u>Episodic??</u>	No	No	No	No
<u>Static??</u>	Yes	Semi	Semi	No
<u>Discrete??</u>				
<u>Single-agent??</u>				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
<u>Observable??</u>	Yes	Yes	No	No
<u>Deterministic??</u>	Yes	No	Partly	No
<u>Episodic??</u>	No	No	No	No
<u>Static??</u>	Yes	Semi	Semi	No
<u>Discrete??</u>	Yes	Yes	Yes	No
<u>Single-agent??</u>				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
<u>Observable??</u>	Yes	Yes	No	No
<u>Deterministic??</u>	Yes	No	Partly	No
<u>Episodic??</u>	No	No	No	No
<u>Static??</u>	Yes	Semi	Semi	No
<u>Discrete??</u>	Yes	Yes	Yes	No
<u>Single-agent??</u>	Yes	No	Yes (except auctions)	No

The environment type largely determines the agent design

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

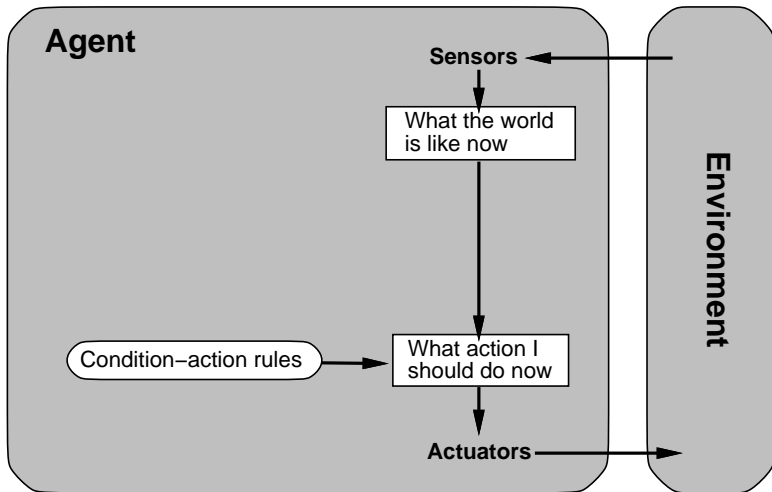
Agent types

Four basic types in order of increasing generality:

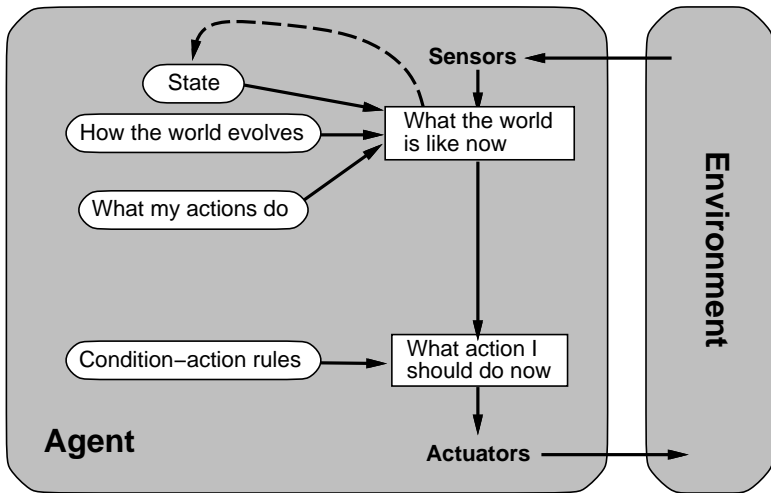
- simple reflex agents
- reflex agents with state
- goal-based agents
- utility-based agents

All these can be turned into learning agents

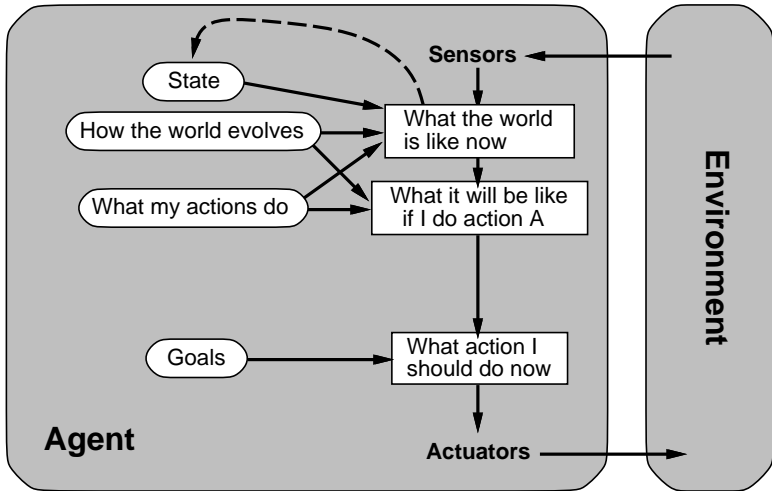
Simple reflex agents



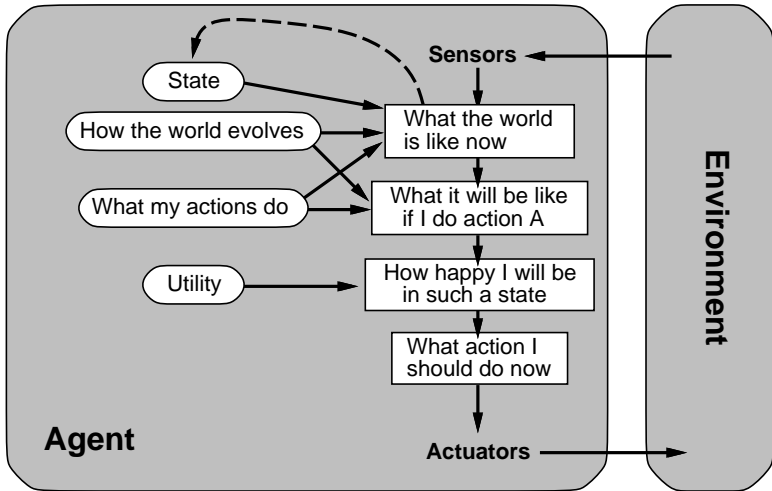
Reflex agents with state



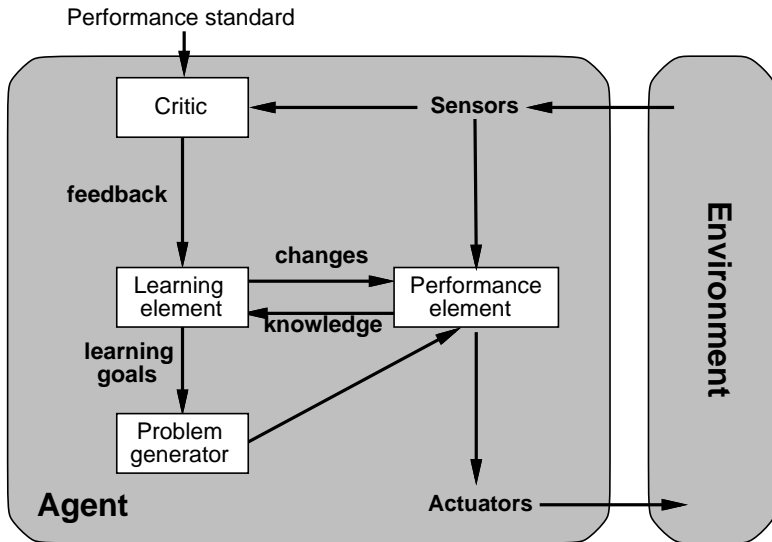
Goal-based agents



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