CS:4420 Artificial Intelligence

Spring 2017

Intelligent Agents

Cesare Tinelli

The University of Iowa

Readings

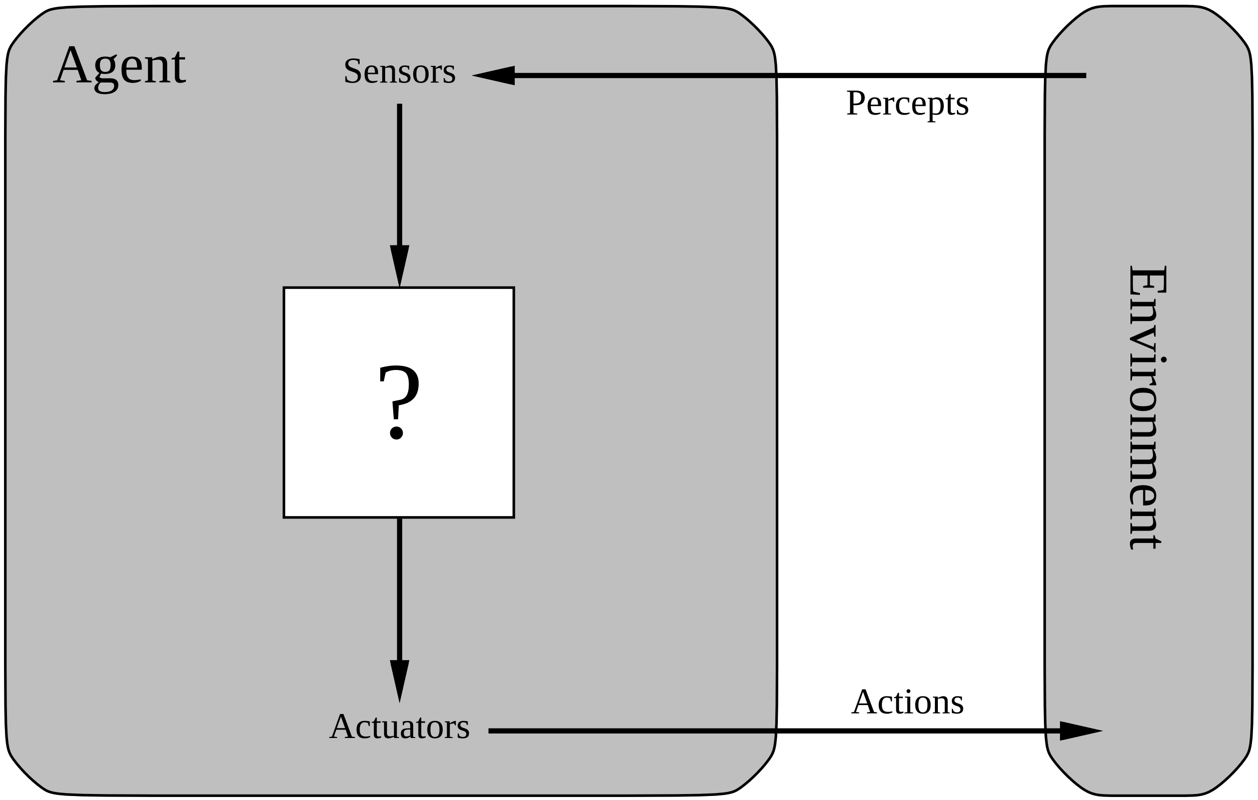
- Chap. 2 of [Russell and Norvig, 2012]

Intelligent Agents

- An agent is a system that perceives its environment through sensors and acts upon that environment through effectors.

- A rational agent is an agent whose acts try to maximize some performance measure.

Agents and Environments



Agents include humans, robots, softbots, thermostats, etc.

Agents as Mappings

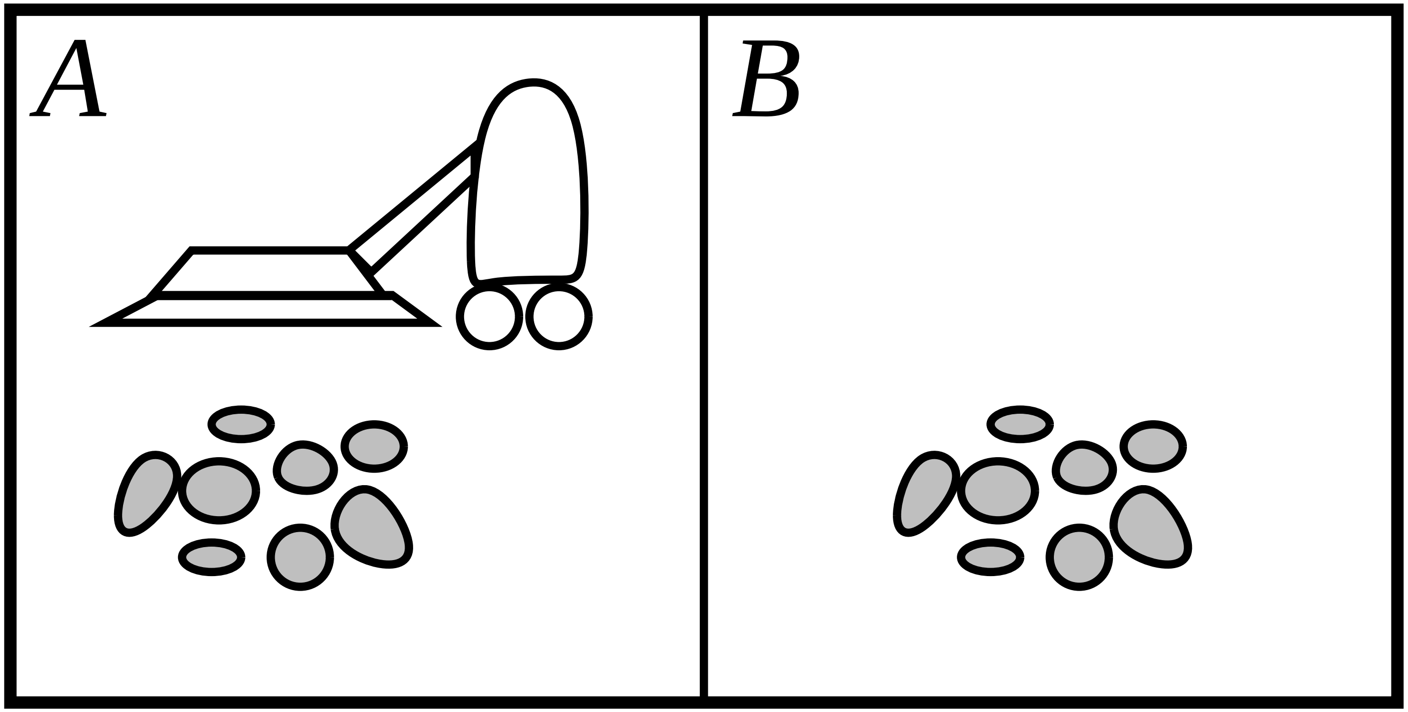
An agent can be seen as a mapping between percept sequences and actions.

$$f : Percept^* \longrightarrow Action$$

The agent program runs on a physical architecture to produce f

The less an agents relies on its built-in knowledge, as opposed to the current percept sequence, the more autonomous it is

Vacuum-cleaner world



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

Actions: $Left$, $Right$, $Suck$, $NoOp$

A vacuum-cleaner agent

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean]$, $[A, Clean]$	$Right$
$[A, Clean]$, $[A, Dirty]$	$Suck$
⋮	⋮

function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** action

if *status* = *Dirty* **then return** *Suck*
else if *location* = *A* **then return** *Right*
else if *location* = *B* **then return** *Left*

More Examples of Artificial Agents

Agent Type	Percepts	Actions	Goals	Environment
Medical diagnosis system	Symptoms, findings, patient's answers	Questions, tests, treatments	Healthy patient, minimize costs	Patient, hospital
Satellite image analysis system	Pixels of varying intensity, color	Print a categorization of scene	Correct categorization	Images from orbiting satellite
Part-picking robot	Pixels of varying intensity	Pick up parts and sort into bins	Place parts in correct bins	Conveyor belt with parts
Refinery controller	Temperature, pressure readings	Open, close valves; adjust temperature	Maximize purity, yield, safety	Refinery
Interactive English tutor	Typed words	Print exercises, suggestions, corrections	Maximize student's score on test	Set of students

Rational Agents

The *rationality* of an agent depends on

- the performance measure defining the agent's degree of success
- the percept sequence, the sequence of all the things perceived by the agent
- the agent's knowledge of the environment
- the actions that the agent can perform

For each possible percept sequence, an ideal rational agent does whatever possible to maximize its performance, based on the percept sequence and its built-in knowledge.

Rationality

- What is the right function?
- Can it be implemented in a small agent program?
- 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?

Rationality

- What is the right function?
- Can it be implemented in a small agent program?
- 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?

- Rational $\neq$ omniscient
- Rational $\neq$ clairvoyant
- Rational $\neq$ successful
- Rational $\implies$ exploration, learning, autonomy

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?
- 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?
  Streets/freeways, traffic, pedestrians, weather . . .
- 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?
  Streets/freeways, traffic, pedestrians, weather ...
- Actuators?
  Steering, accelerator, brake, horn, speaker/display, ...
- 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?
  Streets/freeways, traffic, pedestrians, weather . . .
- Actuators?
  Steering, accelerator, brake, horn, speaker/display, . . .
- Sensors?
  Cameras, accelerometers, gauges, engine sensors, keyboard, GPS, . . .

Internet shopping agent

- Performance measure?

- Environment?

- Actuators?

- Sensors?

Internet shopping agent

- Performance measure?
  price, quality, appropriateness, efficiency
- Environment?
- Actuators?
- Sensors?

Internet shopping agent

- Performance measure?
  price, quality, appropriateness, efficiency
- Environment?
  current and future WWW sites, vendors, shippers
- 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?

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 and data (text, graphics, scripts)

Environment Types

With respect to an agent, an environment may or may not be:

- observable: the agent's sensors detect all aspects relevant to the choice of action
- deterministic: the next state is completely determined by the current state and the actions selected by the agent
- episodic: the agent's experience is divided into "episodes"; the quality of the agent's actions does not depend on previous episodes
- static: it does not change while the agent is deliberating
- discrete: there are a limited number of distinct, clearly defined percepts and actions
- single-agent: there are not more agents in the environment

Environment Types

	Solitaire	Backgammon	E-shopping	Taxi
Observable?				
Deterministic?				
Episodic?				
Static?				
Discrete?				
Single-agent?				

Environment types

	Solitaire	Backgammon	E-shopping	Taxi
Observable?	Yes	Yes	No	No
Deterministic?				
Episodic?				
Static?				
Discrete?				
Single-agent?				

Environment types

	Solitaire	Backgammon	E-shopping	Taxi
Observable?	Yes	Yes	No	No
Deterministic?	Yes	No	Semi	No
Episodic?				
Static?				
Discrete?				
Single-agent?				

Environment types

	Solitaire	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

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

Environment types

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

Environment types

	Solitaire	Backgammon	E-shopping	Taxi
Observable?	Yes	Yes	No	No
Deterministic?	Yes	No	Partly	No
Episodic?	No	No	No	No
Static?	Yes	Semi	Semi	No
Discrete?	Yes	Yes	Yes	No
Single-agent?	Yes	No	Yes/No	No

Environment types

The environment type largely determines the agent design

The real world is (of course)

- partially observable,
- stochastic (instead of deterministic),
- sequential (instead of episodic),
- dynamic (instead of static),
- continuous (instead of discrete),
- multi-agent (instead of single-agent).

Agent Programs

Since an agent is just a mapping from percepts to actions, we can design a program to implement this mapping.

An agent program could be as simple as a table lookup.
However:

- that might be practically infeasible (a chess playing agent, for instance, would need 35^{100} table entries)
- there might be a much more efficient solution
- the agent would have no autonomy

Different Types of Agents

Agents programs can be divided in the following classes, with increasing level of sophistication:

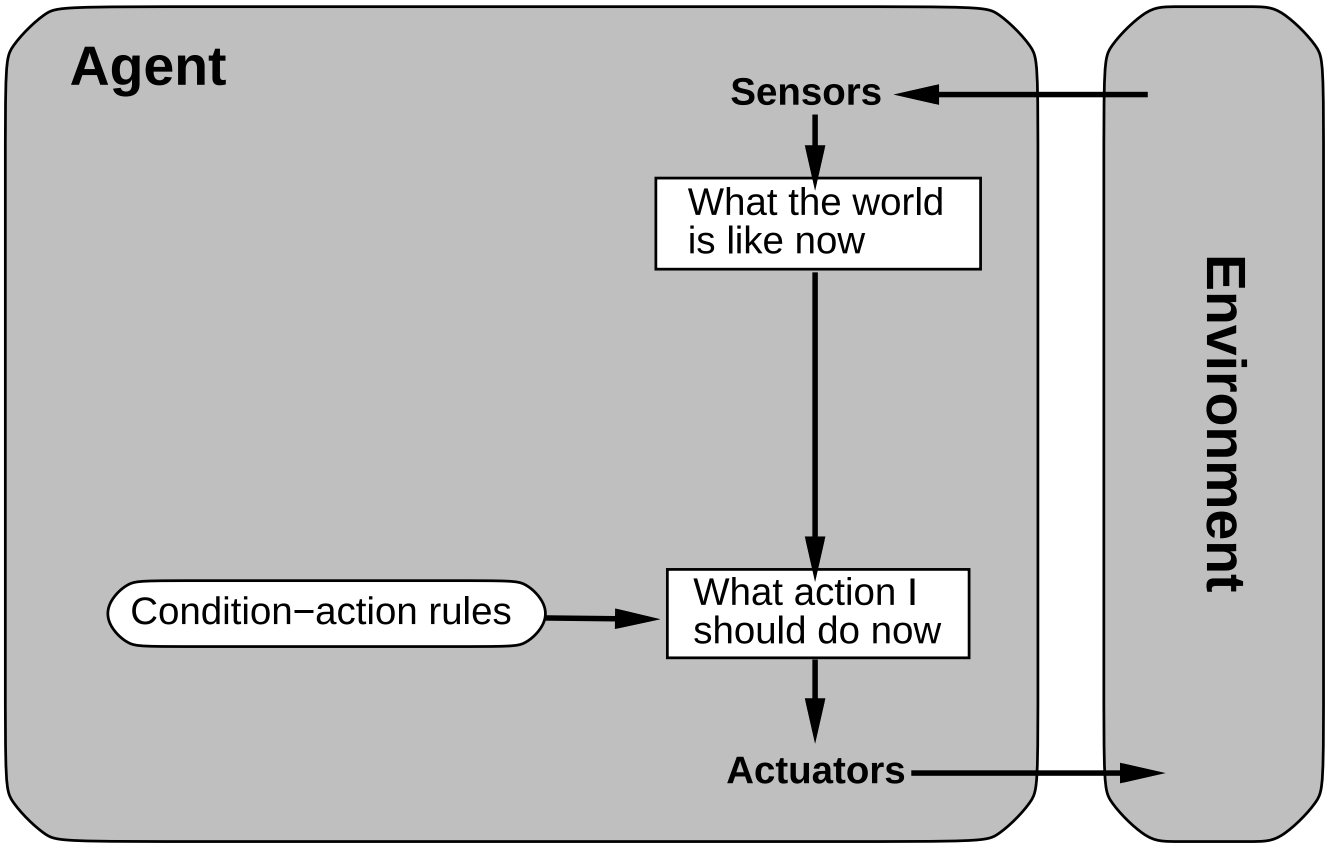
- Stateless reflex agents
- Stateful reflex agents
- Goal-based agents
- Utility-based agents

All these can be designed to be learning agents

A Reflex Taxi-Driver Agent

- We cannot implement it as a table-lookup: the percepts are too complex.
- But we can abstract some portions of the table by coding common input/output associations.
- We do this with a list of condition/action rules:
  - **if** *car-in-front-is-braking* **then** *brake*
  - **if** *light-becomes-green* **then** *move-forward*
  - **if** *intersection-has-stop-sign* **then** *stop*

Simple Reflex Agents



Reflex agents can be implemented very efficiently.

However, they have limited applicability.

Reflex Taxi-Driver Agent with State

- Often, the agent must remember some of its percepts to take an action.
  Ex: car in front signals it is turning left.
- It must also remember which actions it has taken.
  Ex: loaded/unloaded passenger.
- In jargon, it must have internal state.

Reflex Taxi-Driver Agent with State

To update its state the agent needs two kinds of knowledge:

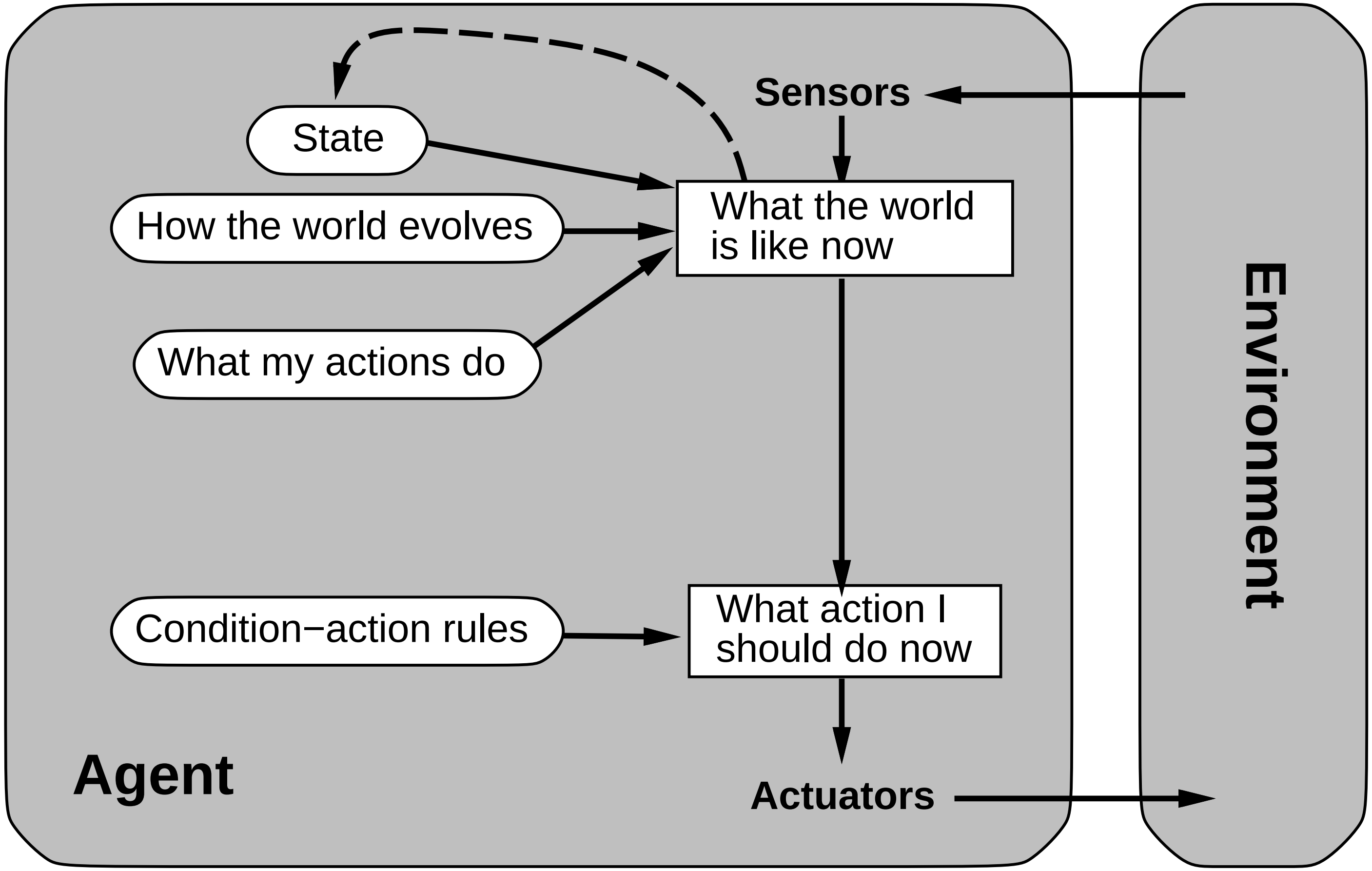
1. how the world evolves independently from the agent;

    Ex: an overtaking car gets closer with time.

2. how the world is affected by the agent's actions.

    Ex: if I turn left, what was to my right is now behind me.

Reflex Agents with Internal State



A Goal-based Taxi-Driver Agent

- Knowing about the world is not always enough to decide what to do.
  Ex: what direction do I take at an intersection?

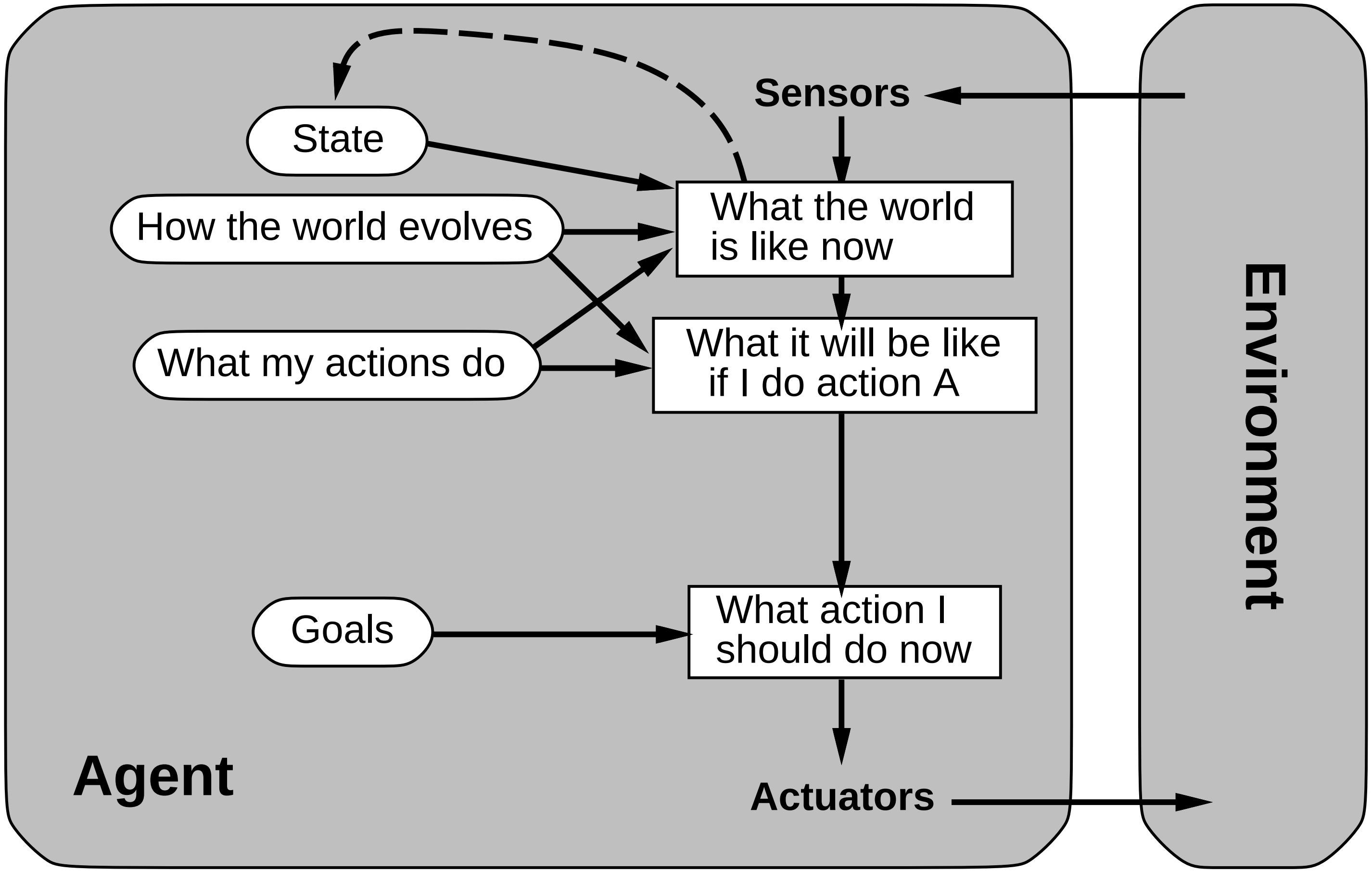
- The agent needs goal information.
  Ex: passenger's destination

- Combining goal information with the knowledge of its actions, the agent can choose those actions that will achieve the goal.

- A new kind of decision-making is required ("what-if reasoning").

- Search and Planning are devoted to find action sequences that achieve an agent's goal.

Goal-based Agents



Goal-based Agents

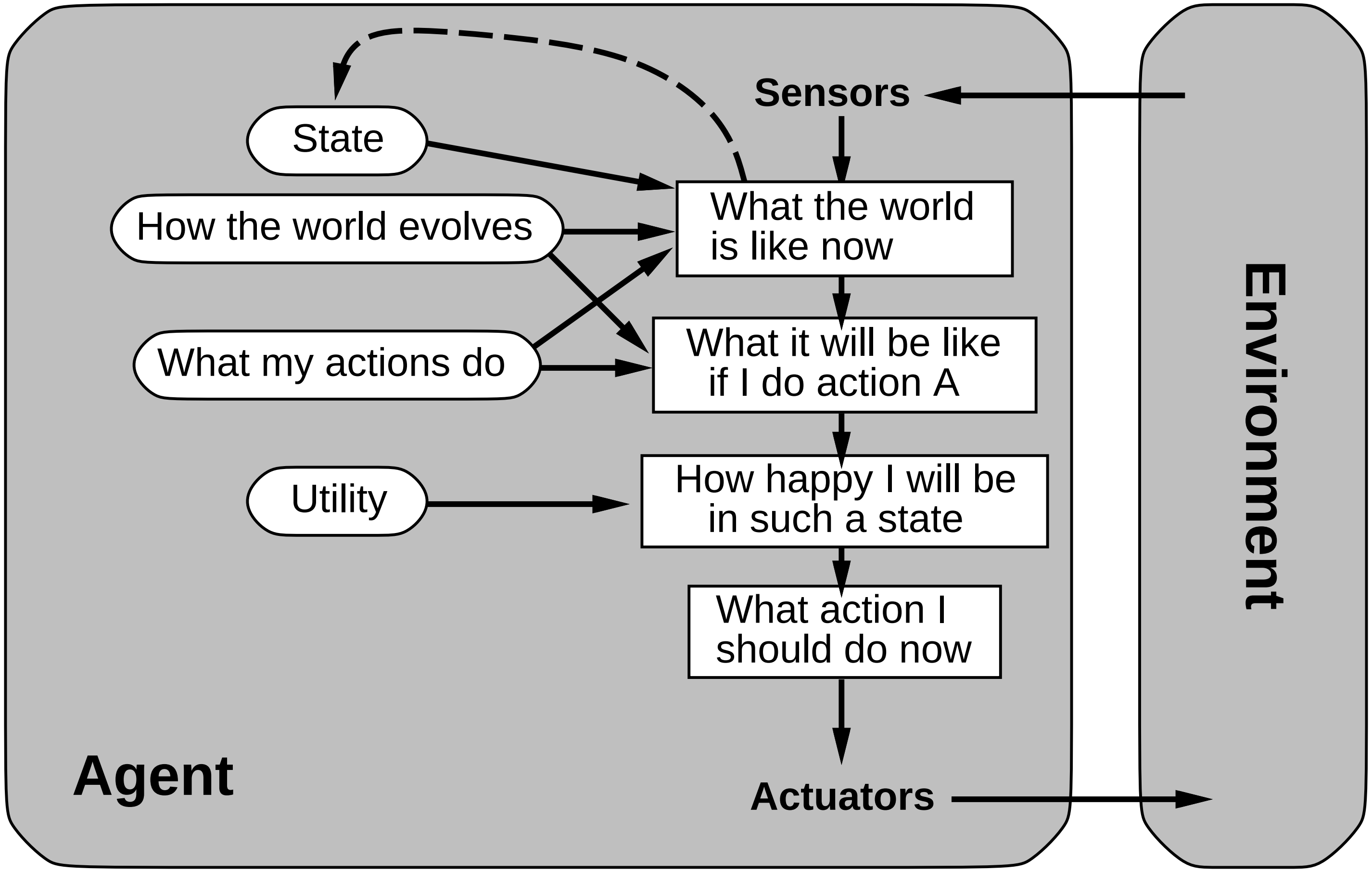
Goal-based Agents are much more flexible in

- responding to a changing environment;
- accepting different goals.

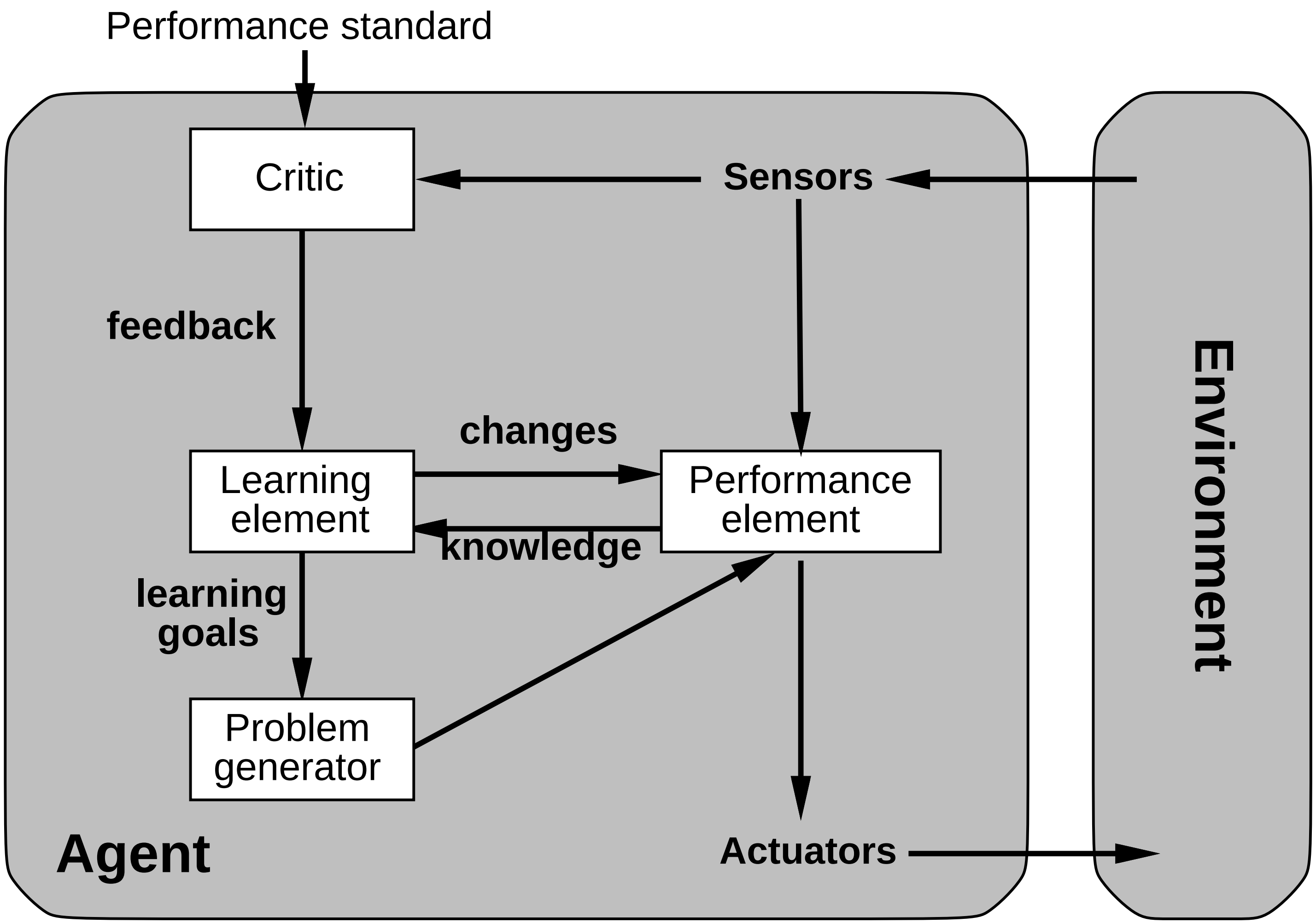
Utility-based Taxi-Driver Agent

- There may be many ways to get to a destination but some may be *better* than others.
  Ex: this way is faster/cheaper/more comfortable/. . .
- A particular configuration of the world, a world state, can be assigned a utility (the quality of being useful) value for the agent.
- A sequence of actions is preferred if it leads to a goal state with higher utility value.
- A utility function aids decision-making in case of
  1. conflicting goals (by helping find a trade-off).
     Ex: minimize trip time and also fuel consumption.
  2. several possible goals, none of which is achievable with certainty.

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

Summary

Environments are categorized along several dimensions:

- observable?
- deterministic?
- episodic? static?
- discrete?
- single-agent?

There are several basic agent architectures:

- reflex
- reflex with state
- goal-based
- utility-based