Introduction to Intelligent Agents

Maria Gini
CSci 8551, Fall 2017

Outline

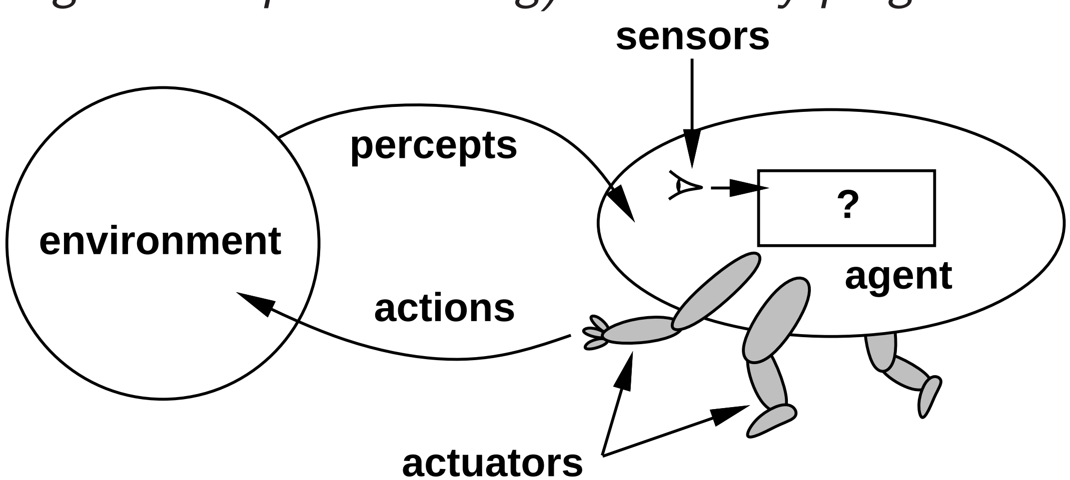
- ♢ What is an agent? (slides from Russell and Norvig)
- ♢ Types of agents (slides from Russell and Norvig)
- ♢ Why should we use agents?
- ♢ What's special about agent-based computing?

Thanks to Russell and Norvig for slides 3 to 12

What is an agent? (Part I)

"An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors," (from Russell and Norvig, Artificial Intelligence: a Modern Approach)

If we define the environment as whatever provides input and receives output (input = sensing and output = acting) then every program is an agent.



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

Rationality

Fixed performance measure evaluates the environment sequence

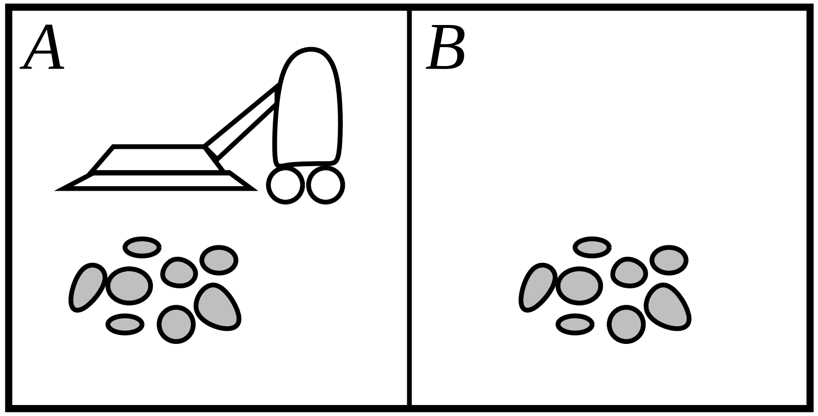
- one point per square cleaned up in time T?
- one point per clean square per time step, minus one per move?
- penalize for $> k$ dirty squares?

A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date

Rational $\neq$ omniscient
Rational $\neq$ clairvoyant
Rational $\neq$ successful

Rational $\Rightarrow$ exploration, learning, autonomy

Vacuum-cleaner world

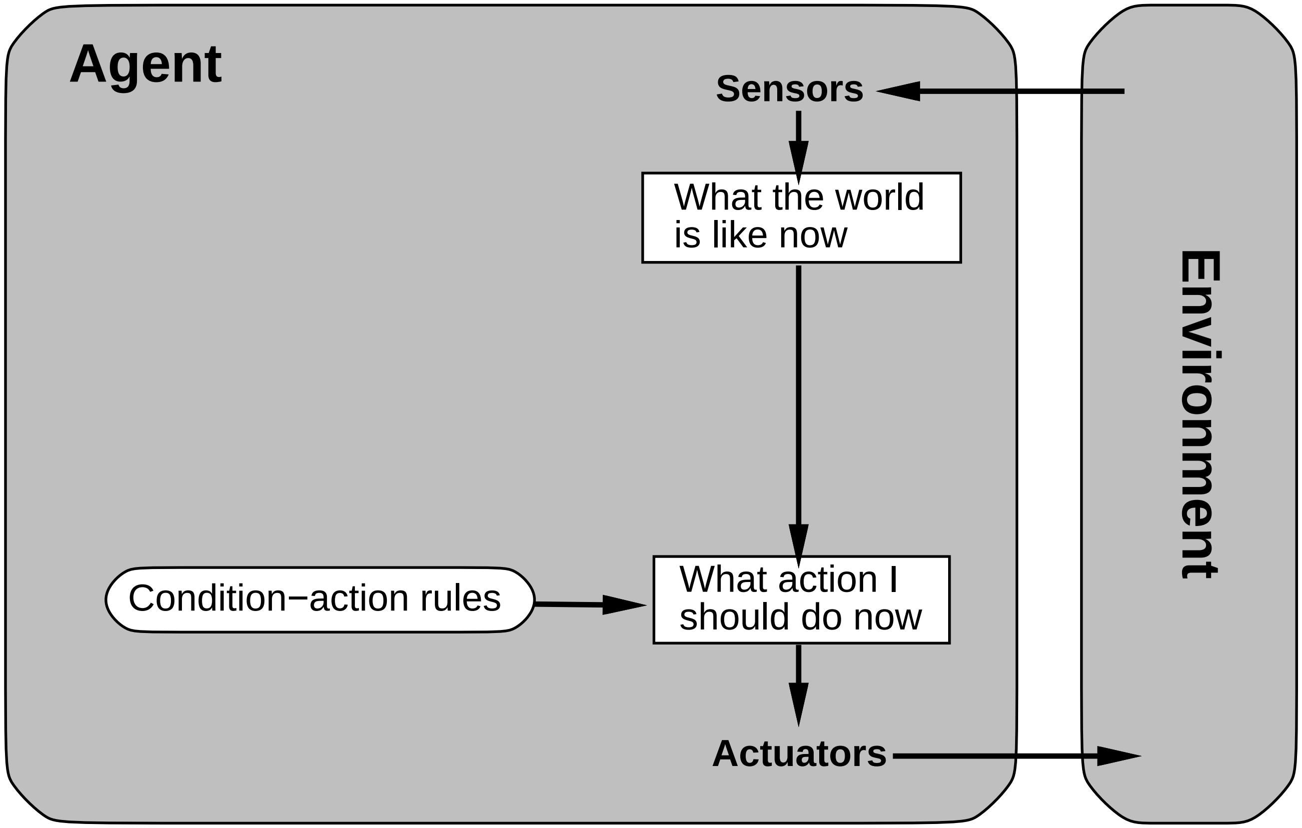


Percepts: location and contents, e.g., $[A, Dirty]$

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

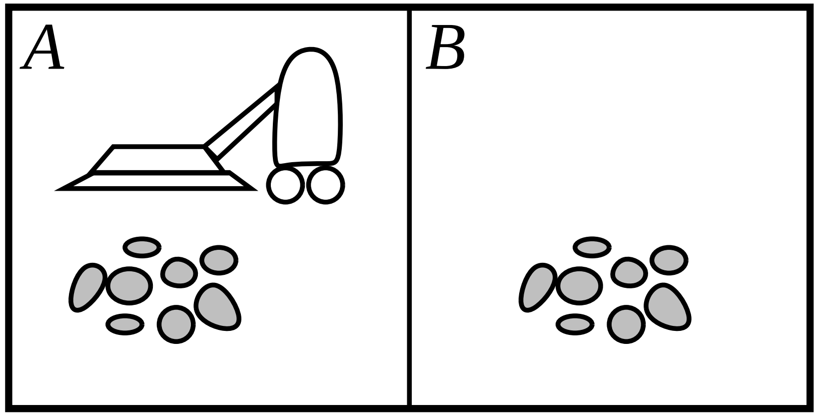
Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[A, Clean]$, $[A, Clean]$	$Right$
$[A, Clean]$, $[A, Dirty]$	$Suck$
⋮	⋮

Simple reflex agents



The agent works by finding a rule whose condition matches the current situation, as defined by perception, and then doing the action associated with the rule. The agent has no memory.

Vacuum-cleaner world



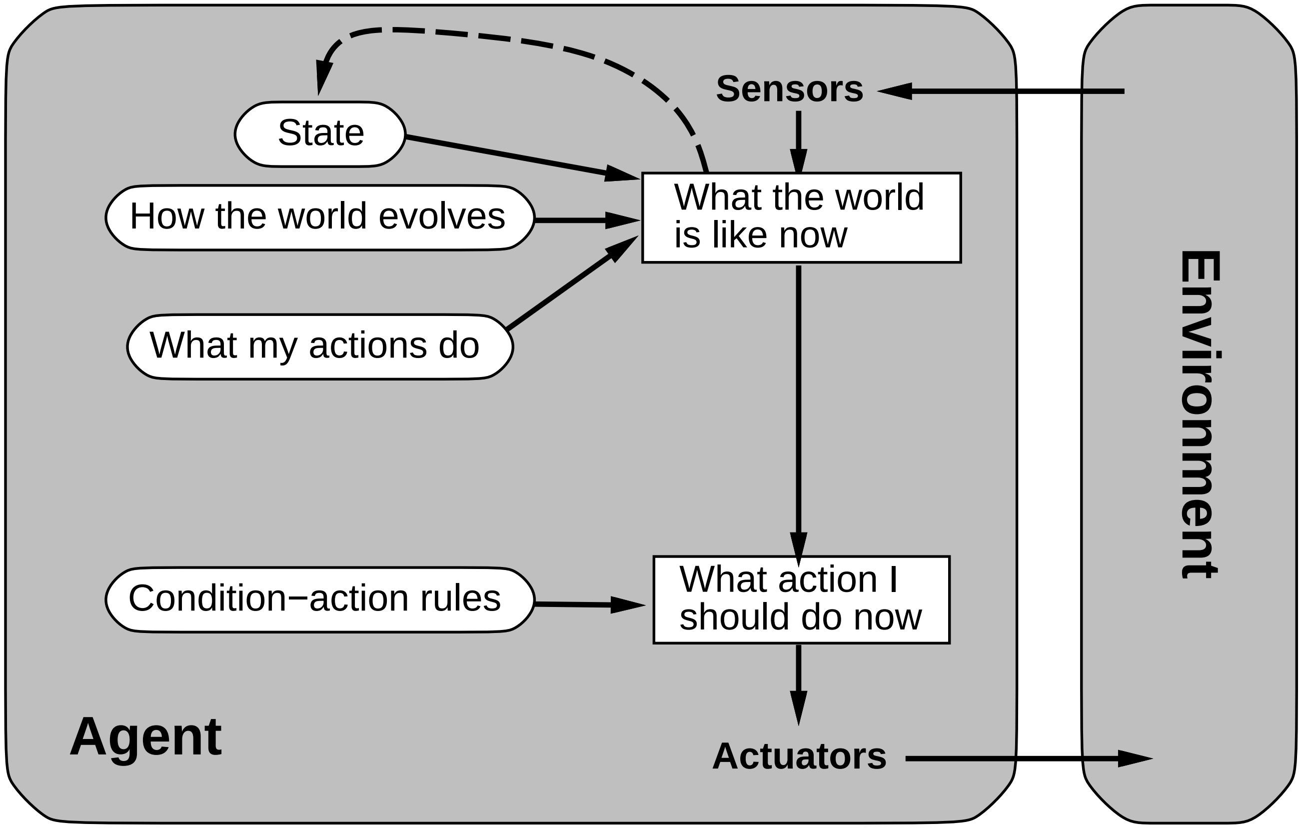
Percepts: location and contents, e.g., $[A, Dirty]$
Actions: $Left$, $Right$, $Suck$, $NoOp$

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*

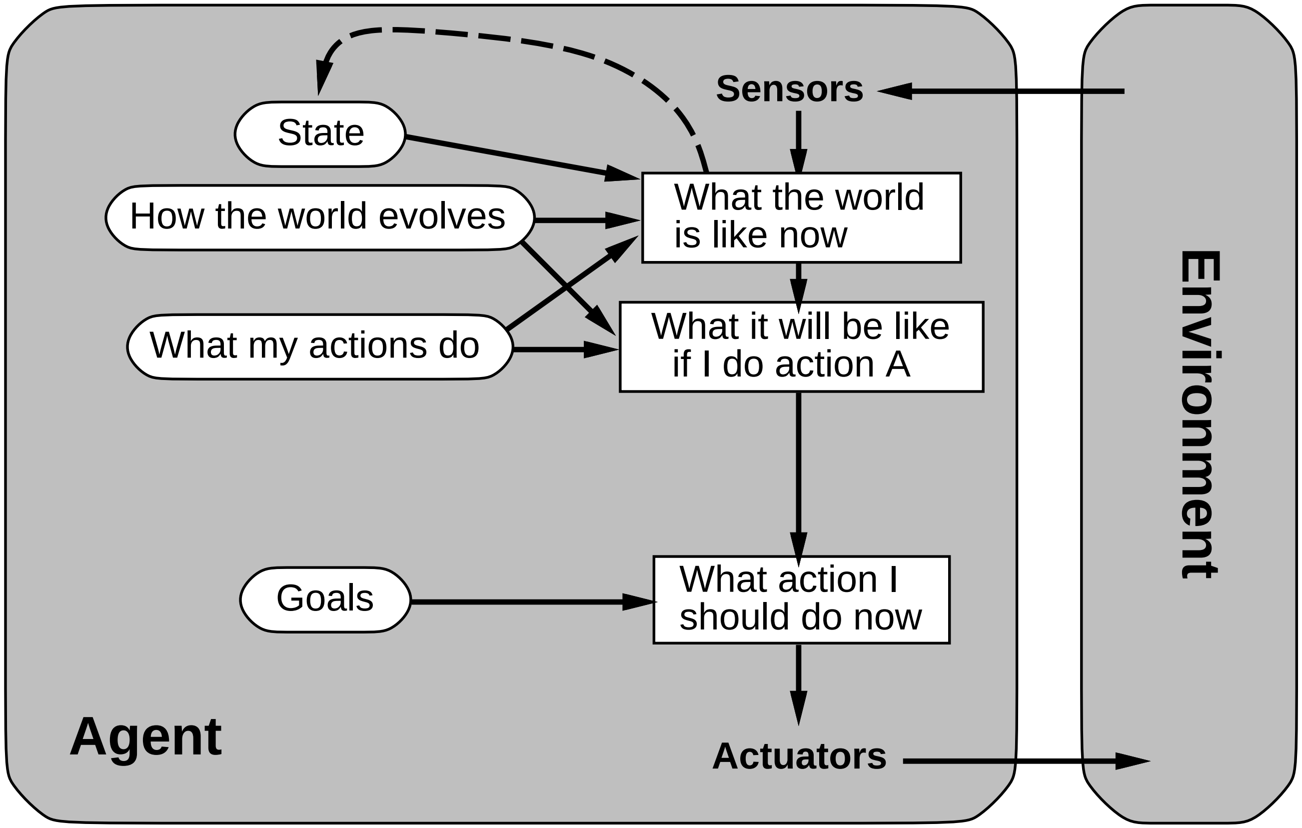
What is the right function? Can it be implemented in a small agent program?

Reflex agents with state



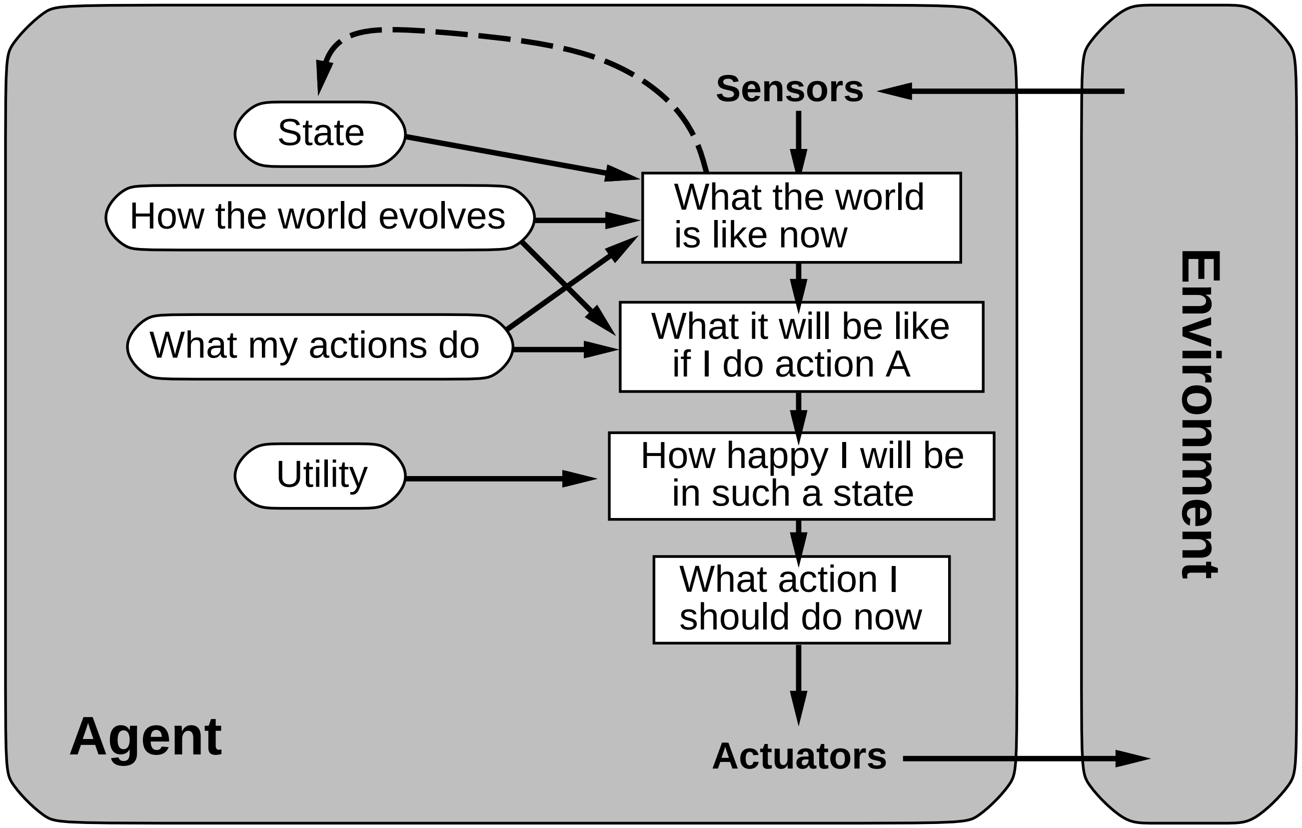
The agent works by finding a rule whose condition matches the current situation, as defined by perception and by the stored internal state, and then doing the action associated with the rule. The internal state acts as a memory and allows for a better selection of the rule to apply.

Goal-based agents



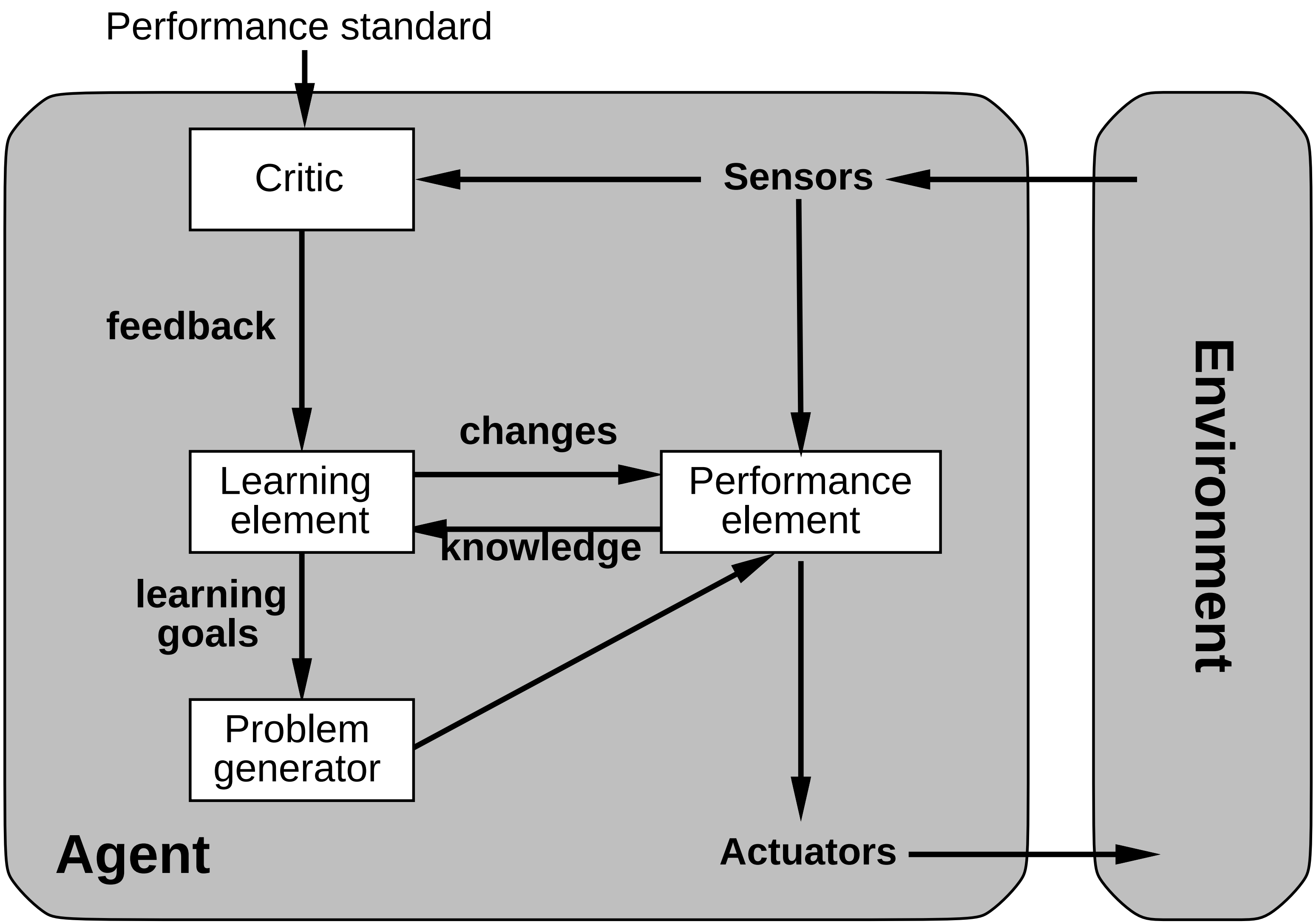
The agent has an explicit goal, and when choosing an action it will select an action that achieves the goal. The decision process requires some plannig ("what will happen if I do action A"). This agent is more flexible than the simple reflex agent, and is capable of achieving goals.

Utility-based agents



The agent has a utility function that maps a state onto real number that describes how well the agent is performing. This agent not only achieves goals, but it maximizes some measure of performance.

Learning agents



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?? US streets/freeways, traffic, pedestrians, weather, . . .

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

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

The environment type largely determines the agent design

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

What is an agent? (Part II)

A weak notion (from Woodridge and Jennings):

- *autonomy:* agents operate without direct intervention of humans, and have control over their actions and internal state;
- *social ability:* agents interact with other agents (and possibly humans) via an agent communication language;
- *reactivity:* agents perceive their environment and respond in a timely and rational fashion to changes that occur in it;
- *pro-activeness:* agents do not simply act in response to their environment, they are capable of taking the initiative (generate their own goals and act to achieve them).

for details see M. Wooldridge and N. R. Jennings. Intelligent Agents: Theory and Practice. In Knowledge Engineering Review 10(2), 1995.

What is an agent? (Part III)

A stronger notion (from Woodridge and Jennings):

An agent has mental properties, such as *knowledge, belief, intention, obligation*. In addition, and agent has other properties such as:

- *mobility:* agents can move around from one machine to another and across different system architectures and platforms;
- *veracity:* agents do not knowingly communicate false information;
- *benevolence:* agents always try to do what they are asked of;
- *rationality:* agents will try to achieve their goals and not act in such a way to prevent their goals from being achieved.

Discussion

Are these agents?

- a thermostat
- a payroll program
- a spell checker

Can you suggest domains and applications where agents could/should be used?

Why should we use agents?

Agents provide three fundamental abstractions:

1. the *autonomy* abstraction;
2. the *cognitive* abstraction;
3. the *society of agents* abstraction.

Computer scientists use abstraction mechanisms (procedural abstractions, abstract data types, objects, etc.) all the time.

The autonomy abstraction

- Autonomy implies ability to make decisions and to initiate actions without direct human supervision. This allows us to view agents as entities that have goals to achieve and are capable of initiating actions to achieve their goals.
- Autonomy often implies the notion of persistence and identity.
- Autonomy also often implies the existence of other autonomous enties. This allows us to see agents as members of a society of agents (see last point).

The cognitive abstraction

- In everyday life, we use folk psychology to explain and predict the behavior of complex intelligent systems, such as people.

  Example: Maria *intended* to prepare her slides.

  Example: Jane *believed* it was raining.

- The term *intentional* system was conied by Daniel Dennett to describe entities "whose behavior can be predicted by the method of attributing belief, desires and rational acumen."

- the term BDI (belief, desires, and intentions) is used to describe agent architecture that have those properties.

- The intentional stance is a convenient way of talking about complex systems, which allows us to predict and explain their behavior without having to understand how their mechanism actually works.

Example of cognitive abstraction

Consider a light switch: "It is perfectly coherent to treat a light switch as a (very cooperative) agent with the capability of transmitting current at will, who invariably transmits current when it believes that we want it transmitted and not otherwise; flicking the switch is simply our way of communicating our desires." (Yoav Shoham)

The intentional stance description is consistent, but ".. it does not buy us anything, since we essentially understand the mechanism sufficiently to have a simpler, mechanistic description of its behaviour." (Yoav Shoham)

The society of agents abstraction

Agents are not entities living alone, they are part of a society of agents. This means agents can share knowledge (ontologies), communicate with each other (Agent communication languages, speech acts), cooperate, coordinate, and negotiate ... but members of a society can also lie, take advantage of others, masquerade themselves as someone else, etc.

Social laws need to be developed to ensure that agents deliver what they promise, do not harm other agents, and act rationally.

Cooperation = working together as a team to achieve a shared goal

Coordination = managing the interdependencies between activities.

Negotiation = reaching agreements on matters of common interest.

What's special about agent-based computing?

- *Openess*. When components of the system are not known in advance, change over time, and are highly heterogeneous (e.g. programming the Internet) an agent-based approach allows to create systems that are flexible, robust, and can adapt to the environment by using their social skills, ability to negotiate with other agents, and ability to take advantage of opportunities.
- *Handle complexity*. With large and complex problems, agents offer a natural way to partition the problem into smaller and simpler components, that are easier to develop and maintain, and are specialized.
- *Natural Metaphor.* Agents provide an easy way to conceptualize metaphors. For instance, an e-mail filtering program can be presented using the metaphor of a personal digital assistant.

What's special about agent-based computing

- *Non determinism.* It is almost impossible to predict in advance how agents will interact. Agents can learn and change their behavior, it is not know what other agents will exist and what they will do.
- *Global behavior emerges from local behavior.* Individual behaviors are designed for individual agents, the global behavior of the system will emerge at run time. Minsky's "Society of Mind" treats agents as individually very simple, but giving rise to "intelligence" when acting together, "in certain very special ways" in societies.
- *New social laws.* Social laws need to be developed to ensure that agents deliver what they promise, do not harm other agents, and act rationally. Agents are to carry out tasks on behalf of the user in an environment where the user has something significant on the line, such as money, reputation, privacy, security, etc.

Disciplines used in agents research

- *Artificial Intelligence* provides methods for representing knowledge and reasoning about it
- *Cognitive Science* provides tools for modeling cognitive processes, such as desires, beliefs, intentions, and commitments.
- *Game Theory* provides mathematical foundations to decision making in the context of interactions with other players.
- *Algorithms and Distributed Systems* provide theory and practical methods for solving didtributed problems, such as constraint satisfaction and optimization.

Where to look for papers on agents

- Journals
  - Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)
  - Journal of AI Research (JAIR)
  - Artificial Intelligence Journal (AIJ)
  - specialized journals
- Conferences
  - AAMAS (Autonomous Agents and Multi-Agent Systems)
  - AAAI annual conference
  - IJCAI (Int'l Joint Conference on AI)
  - specialized conferences for robotics (ICRA, DARS, IAS, etc)