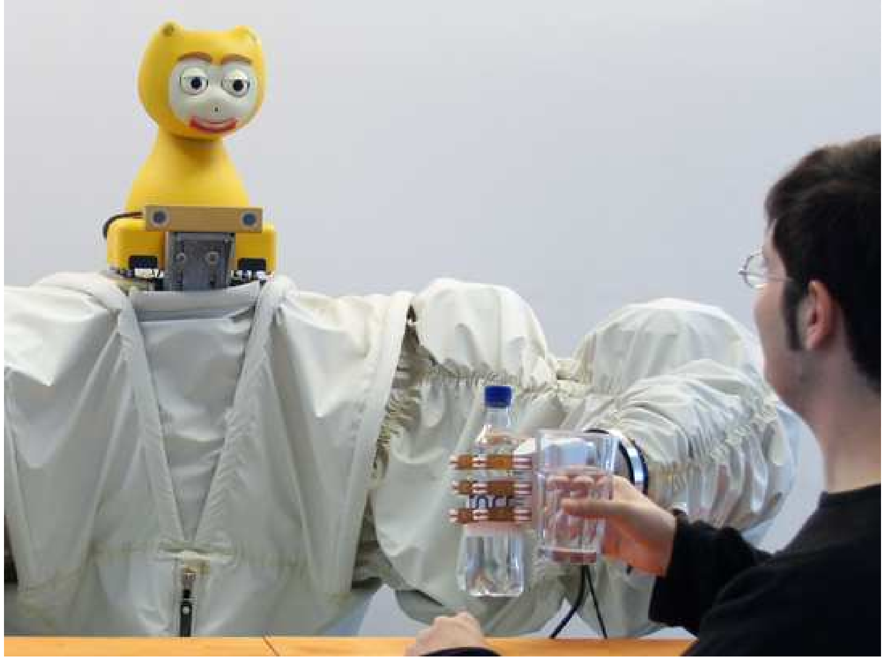
Vorlesung

Grundlagen der Künstlichen Intelligenz

Reinhard Lafrenz / Prof. A. Knoll

Robotics and Embedded Systems
Department of Informatics – I6
Technische Universität München

www6.in.tum.de
lafrenz@in.tum.de
089-289-18136
Room 03.07.055



Wintersemester 2012/13 22.10.2012

Grundlagen der Künstlichen Intelligenz – Techniques in Artificial Intelligence

Chapter 2

Intelligent Agents

R. Lafrenz Wintersemester 2012/13 22.10.2012

What is an Agent?

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

- Sensing modalities

Human	Robot
vision	(computer) vision
hearing	Audio (or speech) recognition
smelling	odometry
touch	force/torque
proprioceptive sensors	encoders and force/torque

What is an Agent?

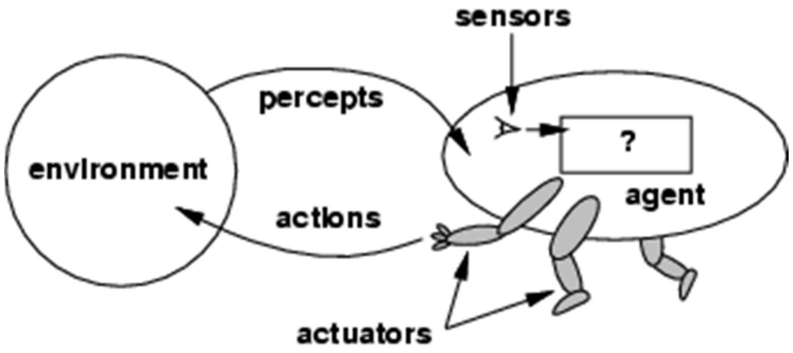
- An *agent* is anything that can be viewed as *perceiving* its *environment* through *sensors* and *acting* upon that environment through *actuators*

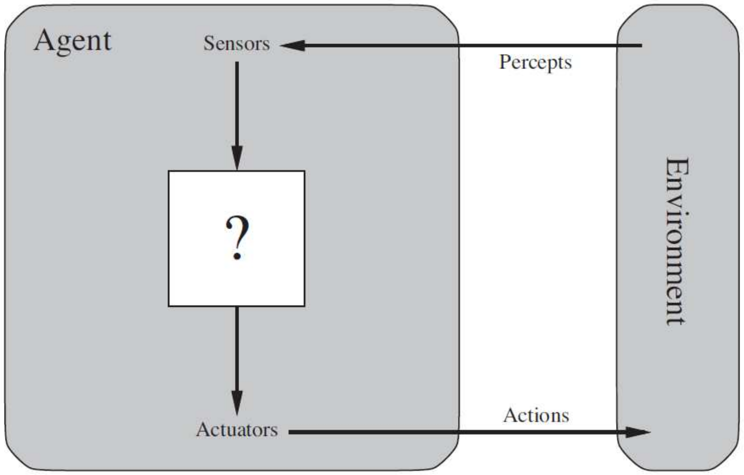
- Acting

Human	Robot
arms and legs	arms, legs, and/or wheels
hands	end-effectors (grippers, tools for e.g. drilling, welding)
other body motion	depending on system
facial expression	artificial emotion expression
talking	speech output

What is an Agent?

- Goal: Design an agent that acts “successfully“ in its environment





Examples of (robotic) agents

http://www6.in.tum.de/Main/ResearchJast

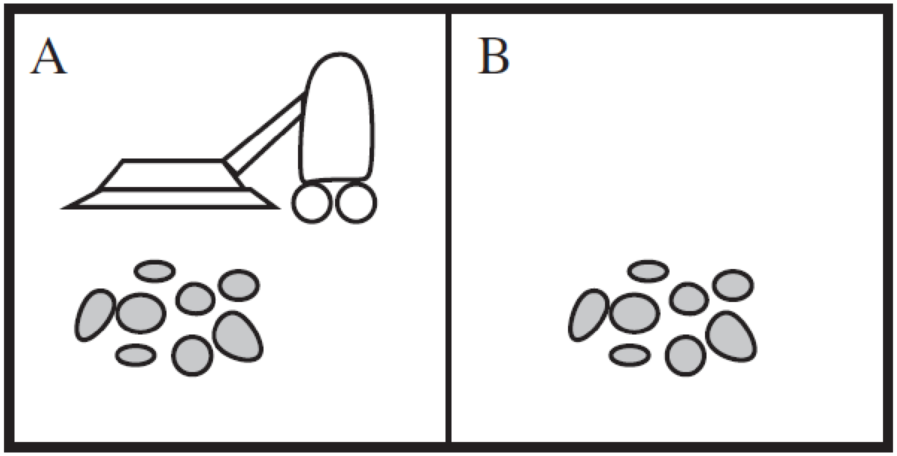
http://mars.jpl.nasa.gov/explore/learnaboutrover/

http://f.asset.soup.io/asset/0059/6767_8701.jpeg

http://www.echord.info/wikis/website/gop

http://upload.wikimedia.org/wikipedia/commons/c/c7/
Danfos-Thermostatventil.jpg

The Vacuum World



- Percepts: location and contents: < *A*|*B*, *Clean*|*Dirty* >
  e.g. <*A, Dirty*>
- Actions: *Left*, *Right*, *Suck*, *NoOp*

- How to describe the agent‘s behavior?

The Vacuum Cleaning Agent

- The agent function maps from percept histories to actions:
-

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

- The agent program is an implementation of *f*, given an agent and its architecture

- agent = architecture + program

- How to describe and implement *f*?

The Vacuum Cleaning Agent

- Defining the agent function by a table

Sequence of percepts	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 >, < B, Clean >	Left
…	…
< A, Clean >, < B, Clean >, < A, Dirty >	Suck
…	…

- Is this an appropriate representation?

The Vacuum Cleaning Agent

- What is the (or a?) correct function
- Can it be implemented in a compact way?

```
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
```

- This describes a finite-state automaton or finite-state machine
- Does this lead to a rational behavior?

Rational agents

- Doing the “right“ thing based on the available information

- Rationality depends on
  - Performance measure: An objective criterion for success of an agent's behavior (degree of goal achievement)
  - A priori knowledge about the environment
  - Possible actions
  - Sequence of percepts

- For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and the built-in knowledge

Rational Agents

- Rational vs. Omniscient
- Maximizing the expected performance, not the actual one
- Examples:
  - Distance to goal given the current state estimation (map)

Euclidian distance?
Dynamic obstacles!

  - Number of work pieces processed by an industrial robot
  - Value of collected information (software agent)

Rational Agents

- Autonomy: An agent is *autonomous* if its behavior is determined by its own experience, i.e. the sequence of percepts. Automomy implies the ability to learn and adapt.

- Suffitient time and sufficient number of percepts are needed for the ability to adapt to new situations and to learn new actions which are adequate in the given situation

PEAS description of a rational agent

- **P**erformance Measure: Assess the perfomance of an agent
- **E**nvironment: The environment in which the agent is able to act and to achieve ist goals
- **A**ctions: the actions tha agent is able to perfom
- **S**ensors: the information an agent can perceive

PEAS description of a thermostat

- **Performance measure:** constant room temperature
- **Environment:** room, house, car
- **Actions:** open or close valves, NoOp
- **Sensors:** “data“ from temperature sensor

http://upload.wikimedia.org/wikipedia/commons/c/c7/
Danfos-Thermostatventil.jpg

- Percepts: temperature T at a given time
- Actions: open or close valve by x%

The agent program of a thermostat

$$\Delta T = T_{current} - T_{desired}$$

- Agent function uses ΔT to select an appropriate action

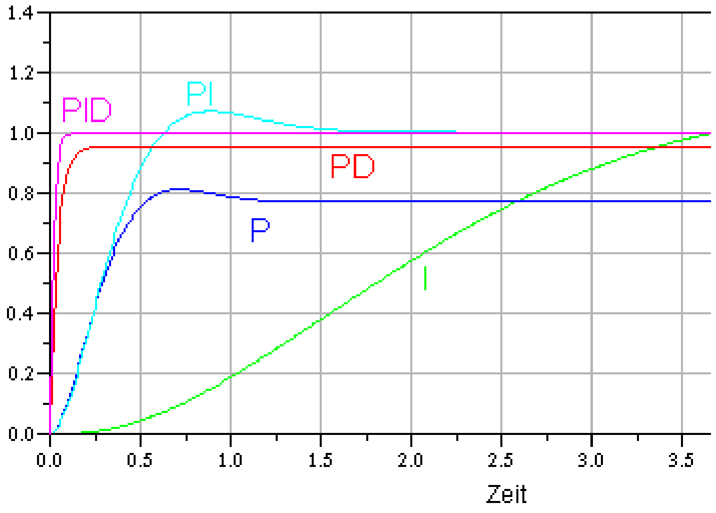
$$f\text{: } \Delta T \rightarrow A$$

```
function THERMOSTAT-AGENT(temperature)
returns an action

if temperature <= desired then return OpenValve(k* $\Delta T$)
else if temperature > desired then return CloseValve(k*$\Delta T$)
else return NoOp
```

- Is this sufficient? (consider that the general agent function takes the ***sequence*** of percepts)

The agent program of a thermostat

- How to measure the performance?
  - At each point in time?
    - Oscillation possible
  - Average over a certain period?
    - Slow reaction possible
  - Taking more than one sensor reading into account
    - How to determine the next action?
  - After goal achievement?
    - How to determine the next action?



http://www.rn-wissen.de/index.php/Regelungstechnik

- This also affects the implementation and the behavior of the agent function

PEAS descriptions of other agents

Agent	Performance measure	Environment	Actions	Sensors
Medical diagnosis system	Healthy patient, minimize costs, lawsuits	Patient, hospital, staff	Screen display (questions, tests, diagnoses, treatments, etc.)	Keyboard (entry of symptoms, findings, patient's answers)
Autonomous taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, traffic, pedestrians, customers	Steering wheel, accelerator, brake, horn, …	Cameras, sonar,, GPS, odometer, voice recognition
Controller in chemical industry	Puirity, quantity, safety,	Industrial environment	Pumps, valves, heating, etc.	Sensors for temp., pressure, spectrometers, ...
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts, bins	Robot arm and hand	Camera, joint angle sensors
Interactive English tutor	Maximize student's score on test	Set of students, examiners	Display of exercises, suggestions, corrections	Keyboard input

PEAS descriptions of other agents

Agent	Performance measure	Environment	Actions	Sensors
Soccer playing				
Searching for titanium on the sea floor				
Bidding for an item at an auction				

Classification of environments

Fully observable	Partially observable
Single agent	Multiple agents
Deterministic	Stochastic
Episodic	Sequential
Static	Dynamic
Discrete	Continuous

- How would you characterize
  - Medical diagnosis systems
  - Playing chess (with clock)
  - Playing soccer
  - Taxi driving

Taxonomy of agents

- Table-driven agent

```
function TABLE-DRIVEN-AGENT(percept)
returns  an action

percepts: a sequence, initially empty
table: the table with percet sequences and actions, fully specified

append percept to the end of percepts
action ← lookup (table, percepts)
return action
```

  - Might get very large
  - Generation takes time (by programmer or learning algorithm)

Taxonomy of agents

- Table-driven agent
  - What is the size of the table of the agent has P possible percepts and works for T time steps?
  - What would be the table size for a 1h drive with an autonomous taxi using a 30frames/sec 640x40 / 24bit camera as the percepts?

27 MB/sec ⇒ table size $10^{250.000.000.000}$

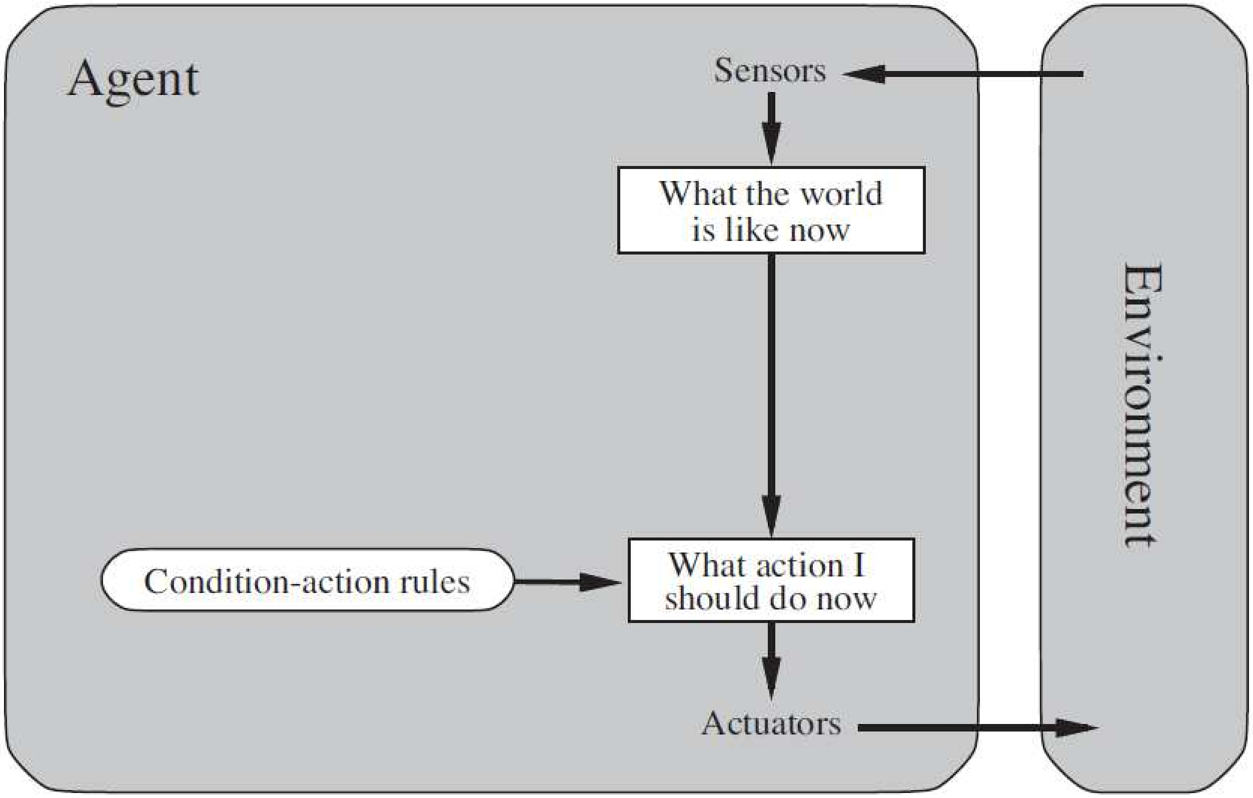
State space for Chess: 10^{150}

Estimated number of atoms in the universe: 10^{80}

… practically infeasible

Taxonomy of agents

- Simple reflex agent



  – Direct use of sensory data often intractable

Taxonomy of agents

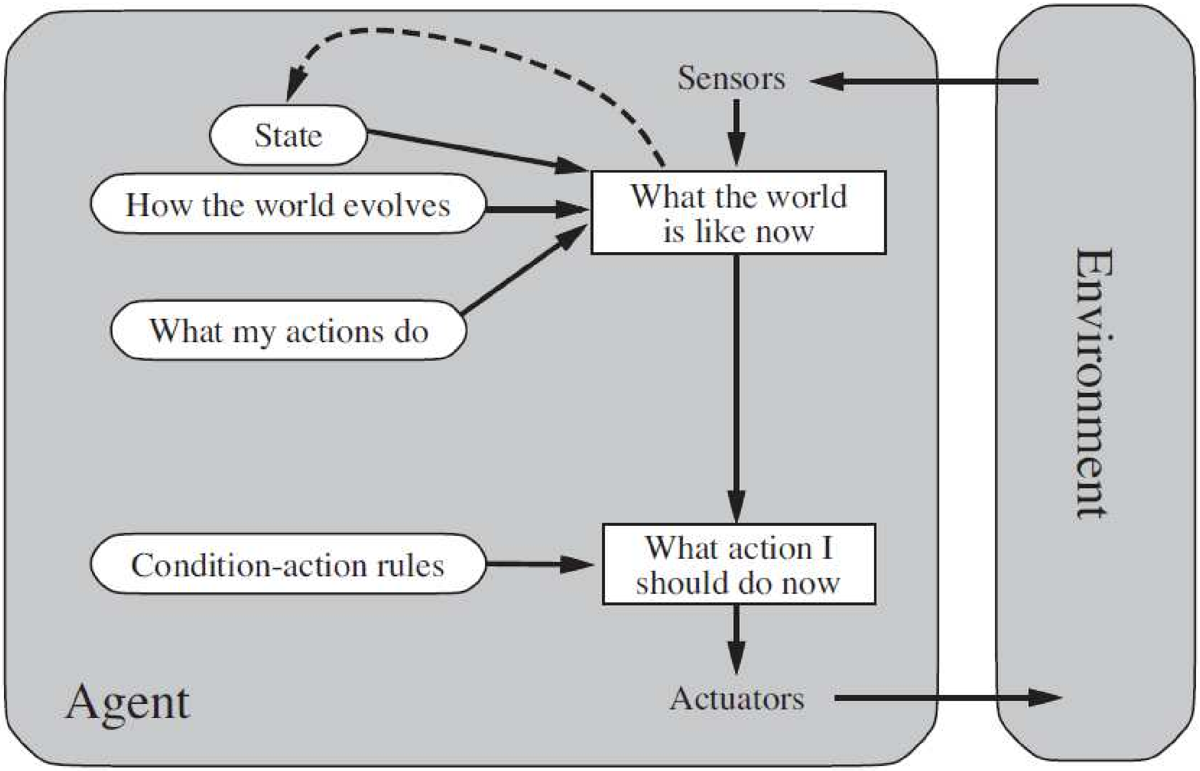
- Simple reflex agent

```
function SIMPLE-REFLEX-AGENT(percept)
returns  an action

rules: a set of condition-action rules

state ← interpret-input (percepts)
rule ← rule-match(state, rules)
action ← rule-action (rule)
return action
```

  – Interpretation of percepts

Taxonomy of agents

- Agents with internal world model



  - If history of percepts is needed to select an action, it must be represented in appropriate way

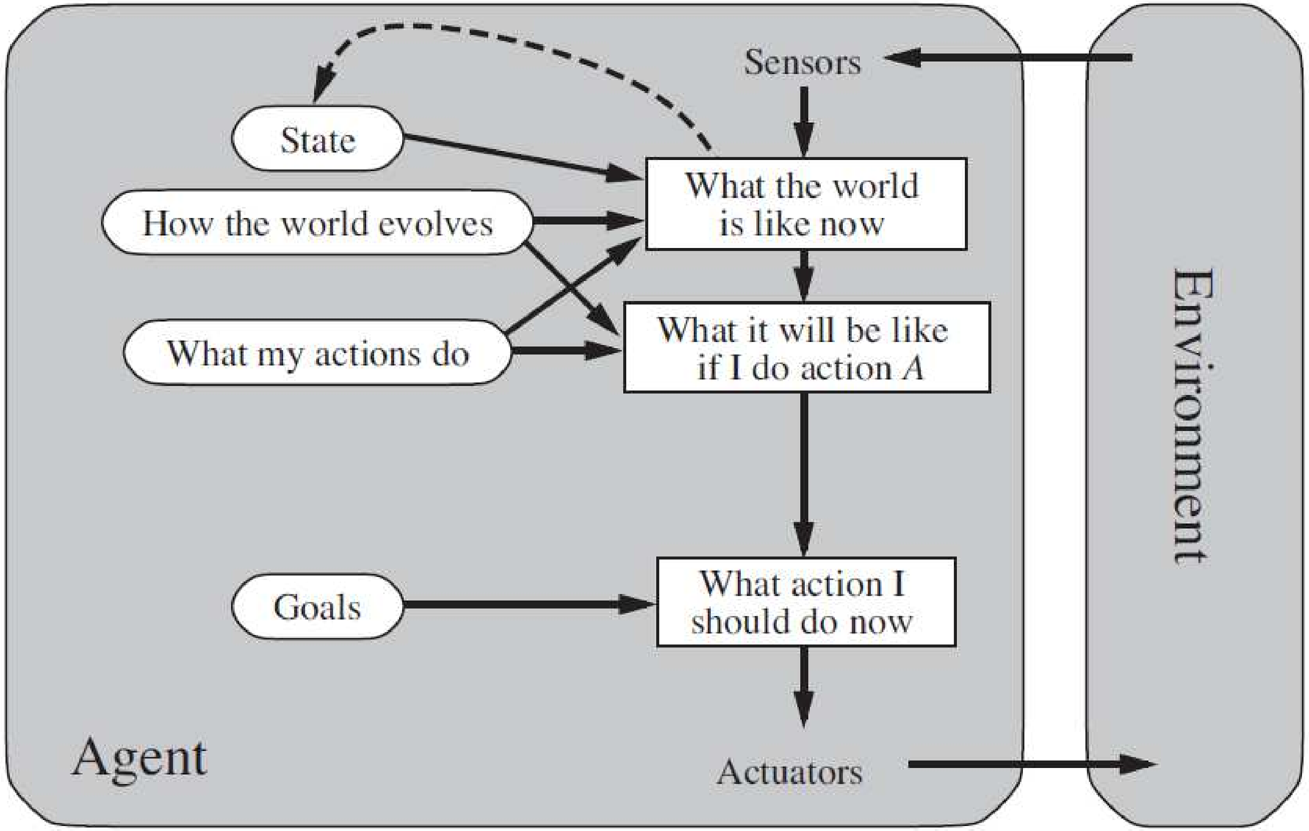
Taxonomy of agents

- Reflex agent with internal state

```
function REFLEX-AGENT-WITH-STATE(percept)
returns  an action

state: a description of the current world state
rules: a set of condition-action rules

state ← interpret-input (state, percepts)
rule ← rule-match(state, rules)
action ← rule-action (rule)
state ← update-state(state, action)

return action
```

Taxonomy of agents

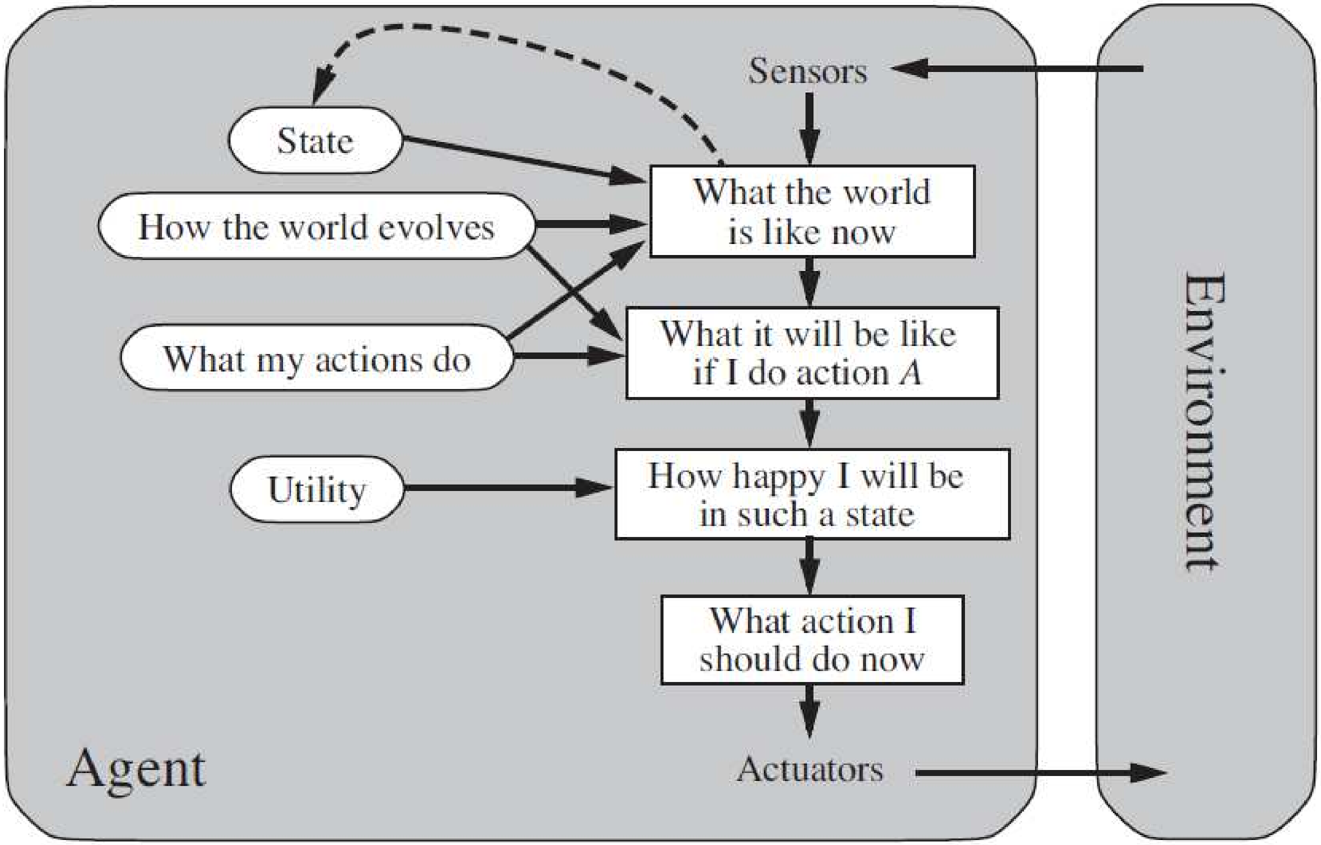
- Goal-based agents


Sensors
State
What the world is like now
How the world evolves
What my actions do
What it will be like if I do action *A*
Goals
What action I should do now
Agent
Actuators
Environment

  - Sole percepts are often not sufficient to choose an action
  - Explicit representatioon of goals and using it for action selection

Taxonomy of agents

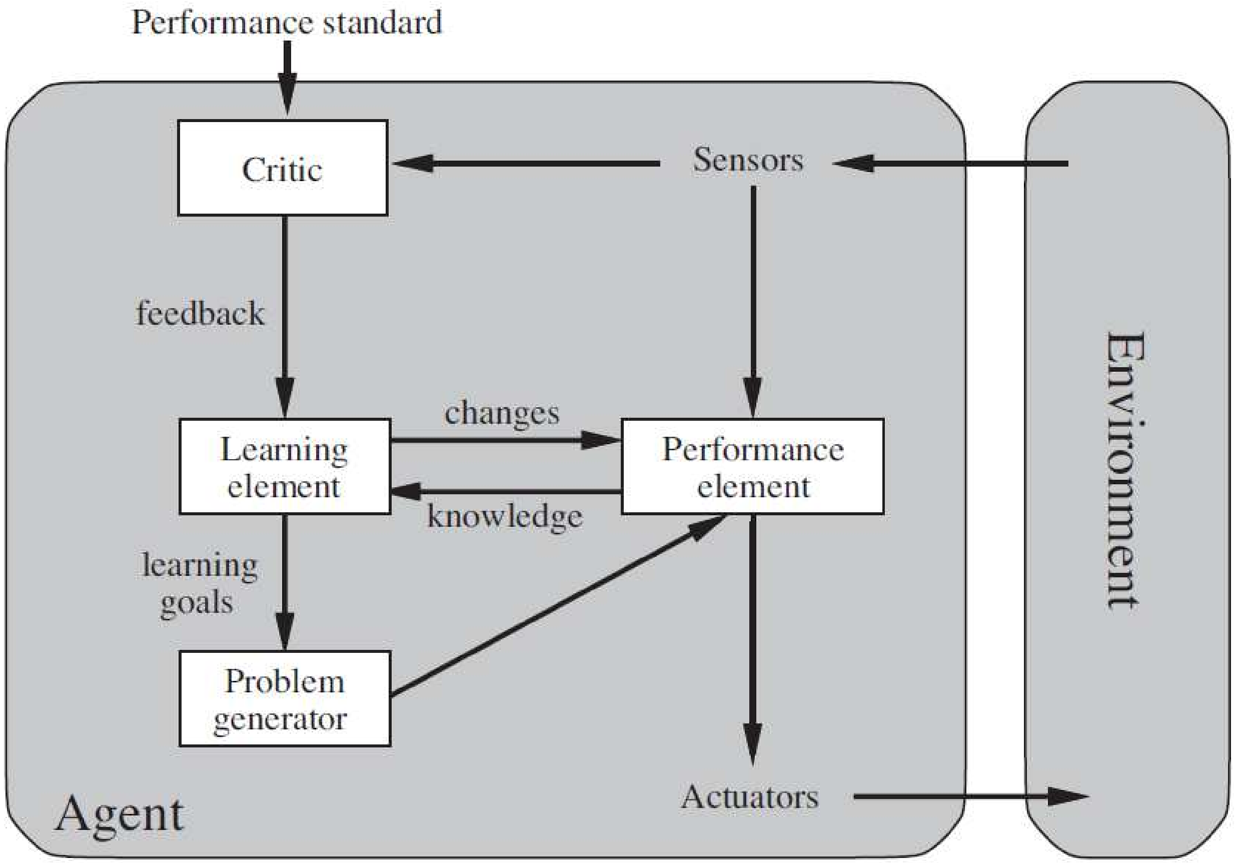
- Utility-based agents


Sensors
State
What the world is like now
How the world evolves
What it will be like if I do action *A*
What my actions do
Utility
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

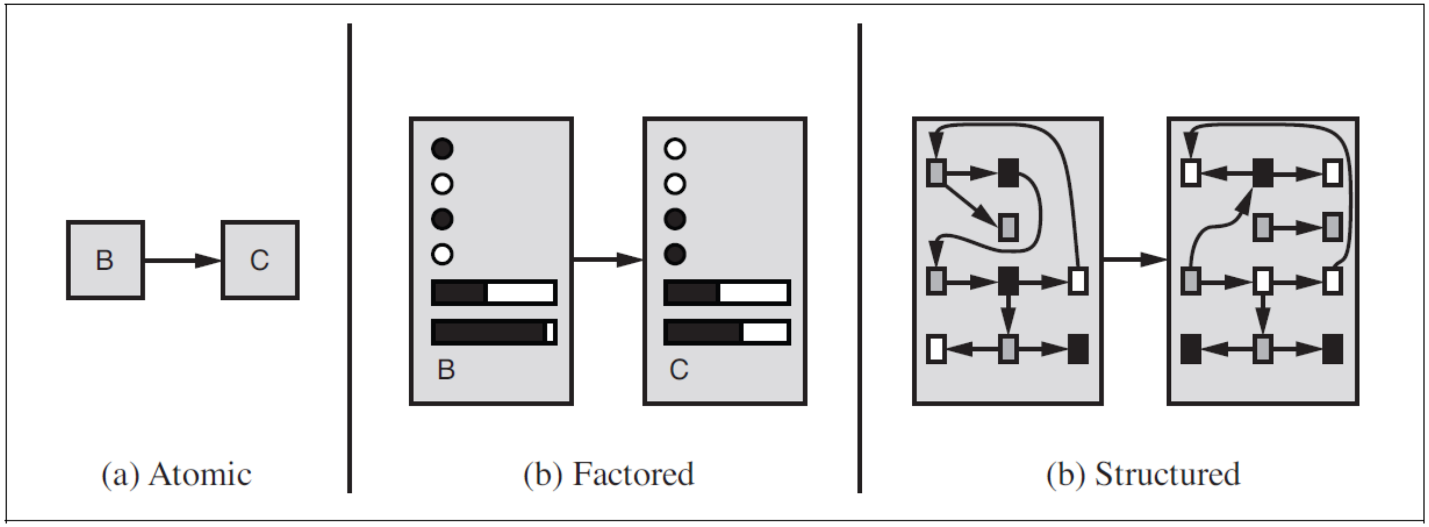
  - In most cases, several actions can be performed in a given state
  - Choose the action based on the utility of the expected next state

Taxonomy of agents

- Learning agents



Representation of states and transitions



- (a) Atomic: „blackbox“, no internal structure, e.g. a name
  - Used e.g. for search algorithms, games, HMM, MDP
- (b) Factored: vector of variables or attributes
  - Used e.g. for reasoning with propositional logic, constraint satisfaction
- (c) Structured: ocjects and relations
  - Bases for first-order logic, relational (or object-oriented) databases, probability models, knowledge-based learning, etc.

Discussion

Are the following statements true or false? Why?

a) An agent that senses only partial information about the state cannot be perfectly rational.

b) There exist task environments in which no pure reflex agent can behave rationally.

c) There exists a task environment in which every agent is rational.

d) The input to an agent program is the same as the input to the agent function.

e) Every agent function is implementable by some program/machine combination.

Discussion

f) Suppose an agent selects its action uniformly at random from the set of possible actions. There exists a deterministic task environment in which this agent is rational.

g) It is possible for a given agent to be perfectly rational in two distinct task environments.

h) Every agent is rational in an unobservable environment.

i) A perfectly rational poker-playing agent never loses.

Summary

- An agent is an entity that perceives and acts
- It consists of an architecture and a program
- A ideal rational agend is expected to maximize its performance measure, given the evidence provided by the percept sequence and the built-in knowledge
- There are different types of environments, some are more challenging than others
  - E.g. partially observable, stochastic, sequential, dynamic, continuous
- There are different types of agents
  - Reflex agents, only reacting to the perceipts
  - Goal-based agents, trying to achieve given goal(s)
  - Utility-based agents, maximizing their performance