Kapitel 2. Intelligente Agenten

Übersicht

2.1 Definitionen von Agenten

2.2 Beispiele von Agenten

Zeitplan:

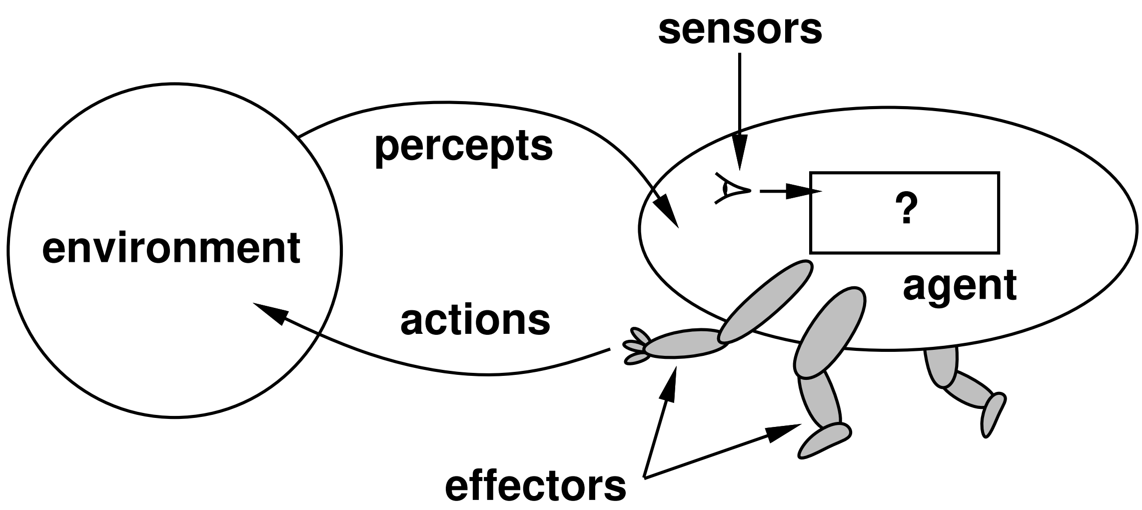
- 15 Minuten

1 Einleitung

2 Intelligente Agenten

2.1 Definitionen von Agenten

Definition 2.1 (Agent)
Ein Agent ist etwas, das seine Umgebung mittels Sensoren wahrnimmt und in dieser Umgebung mittels Werkzeugen agiert.



Definition 2.2 (Rationaler, allwissender Agent)
Rationale Agenten sind solche, die das **Richtige** *tun. (Ein Maß für die Performanz wird benötigt.)*

Allwissende Agenten sind solche, die die Resultate ihrer Aktionen im voraus wissen.

Rationale Agenten sind i.a. nicht allwissend!

Aphorismus von Karl Kraus: Im Zweifelsfall entscheide man sich für das richtige.

Wie ist das **Richtige** definiert und wovon hängt es ab?

1. Performanz-Maß (möglichst objektiv),
2. Percept-Sequenz: was bisher wahrgenommen wurde,
3. Wissen des Agenten über die Umgebung,
4. wie der Agent agieren kann.

Definition 2.3 (Idealer rationaler Agent)
Ein idealer rationaler Agent wählt für jede Percept-Sequenz genau die Aktion aus, die sein Performanz-Maß maximiert (bei gegebenem Wissen über die Umgebung).

Agenten können durch Abbildungen

$$\text{Menge der Percept-Sequenzen} \mapsto \text{Menge der Aktionen}$$

mathematisch beschrieben werden.

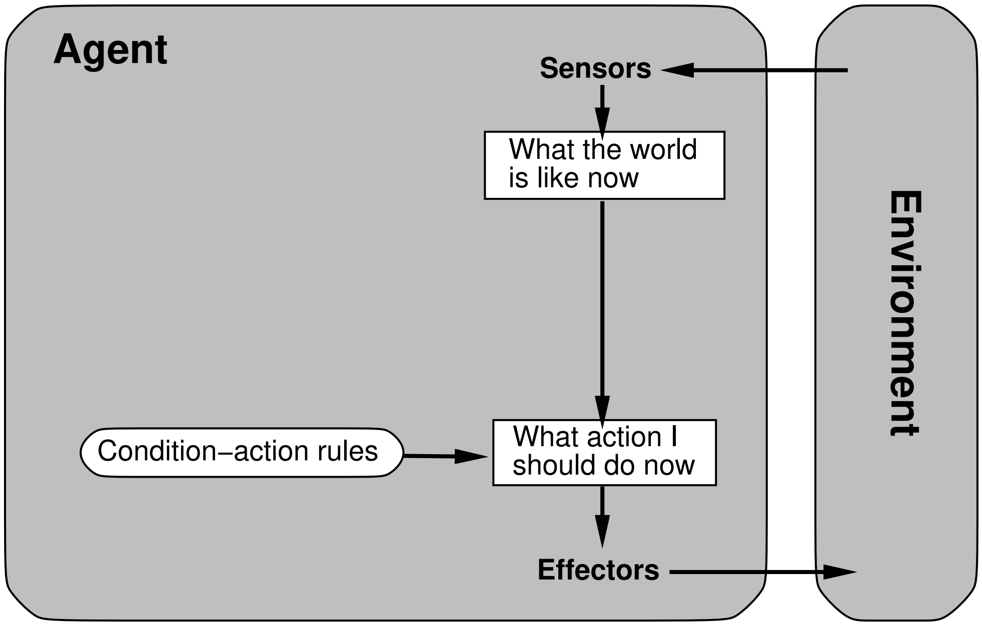
Die interne Struktur eines Agentenist

$$\textbf{Agent} = \textbf{Architektur} + \textbf{Programm}$$

Die KI beschäftigt sich damit, Agenten-Programme zu entwerfen.

Hier einige Beispiele:

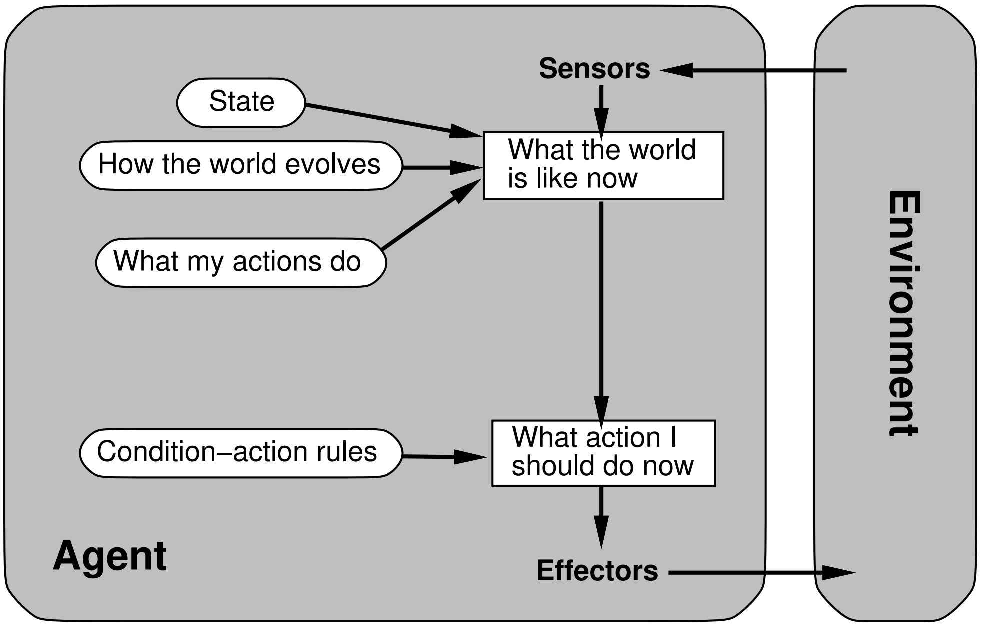
1. **Produktions-Regeln:** Falls der Vorgänger bremst, dann bremse auch.




```
function SIMPLE-REFLEX-AGENT(percept) returns action
  static: rules, a set of condition-action rules

  state ← INTERPRET-INPUT(percept)
  rule ← RULE-MATCH(state, rules)
  action ← RULE-ACTION[rule]
  return action
```

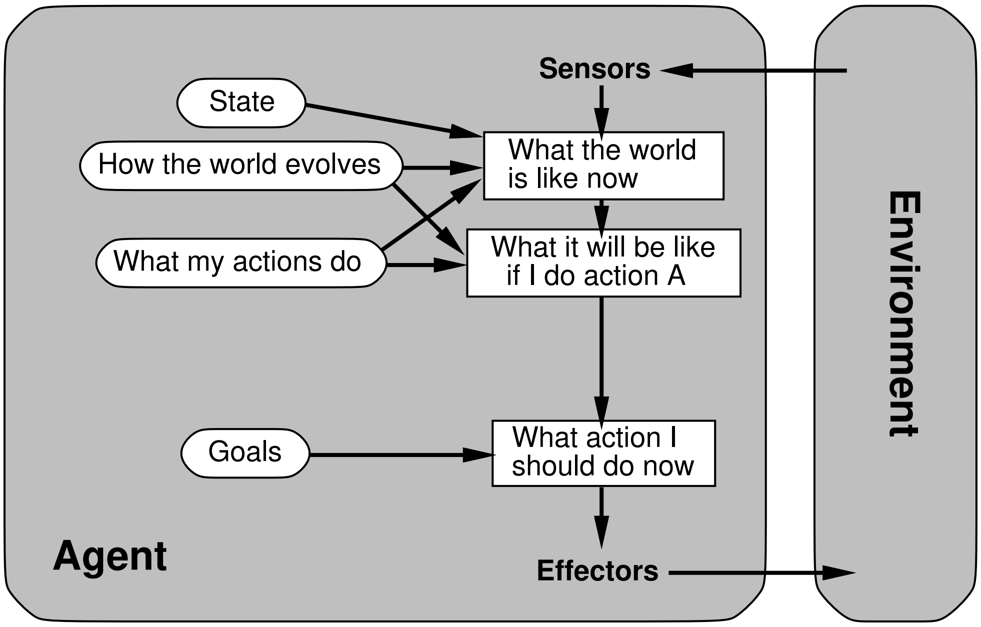
2. **zusätzlich interne Zustände:**



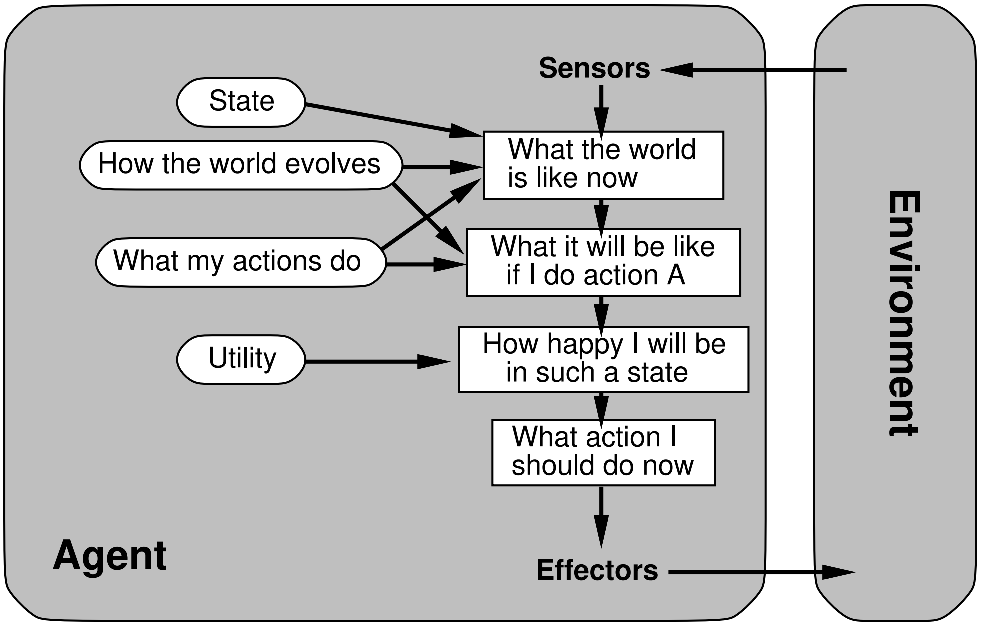

```
function REFLEX-AGENT-WITH-STATE(percept) returns action
  static: state, a description of the current world state
          rules, a set of condition-action rules

  state ← UPDATE-STATE(state, percept)
  rule ← RULE-MATCH(state, rules)
  action ← RULE-ACTION[rule]
  state ← UPDATE-STATE(state, action)
  return action
```

3.


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

4.


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