

Wissensrepräsentation

Christopher Habel, Carola Eschenbach
Sommersemester 2003

Vorlesung 3: Agenten

- Agentenarchitekturen
- Wissensbasierte Agenten

Agenten: Architekturen – Programmen

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

agent = architecture + program

- Entwurf von
 - Architekturen (generell)
 - Programmen (speziell, aufgaben- und umgebungsspezifisch)

Ch. Habel / C. Eschenbach: Wissensrepräsentation, SoSe 2003

3 – 2

Interne Struktur von Agenten: Architekturen

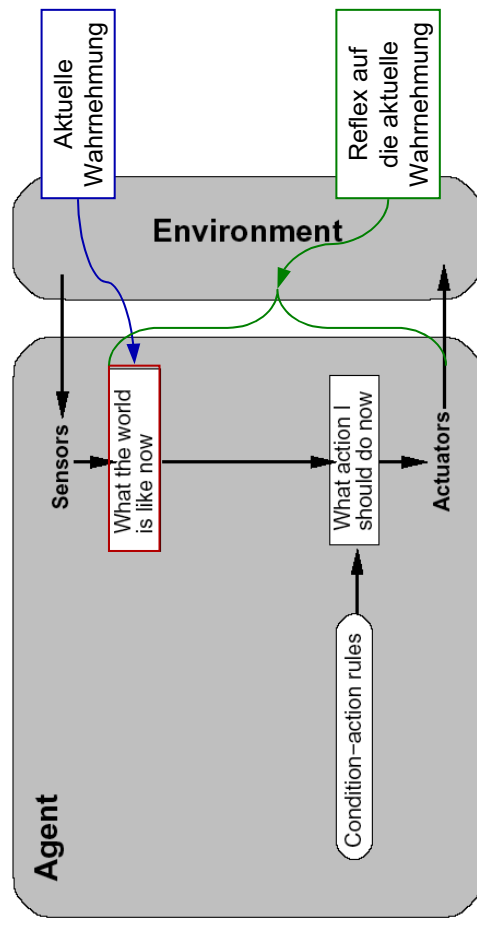
Vier grundlegende Architekturen + Wissenserwerb

- Simple reflex agents
 - Model-based reflex agents
 - Goal-based agents
 - Utility-based agents
- Learning agents

Ch. Habel / C. Eschenbach: Wissensrepräsentation, SoSe 2003

3 – 3

Simple reflex agent



Ch. Habel / C. Eschenbach: Wissensrepräsentation, SoSe 2003

3 – 4

Simple reflex agent: Struktur des Agenten-Programms

Function SIMPLE-REFLEX-AGENT(*percept*) returns an action

static: *rules*, a set of condition-action rules

state ← INTERPRET-INPUT(*percept*)

rule ← RULE-MATCH (*state*, *rules*)

action ← RULE-ACTION (*rule*)

return *action*

Leistung des Agenten

- ist abhängig von der vollständigen Beobachtbarkeit der Umgebung

- Interpretation von Rücklicht/Bremslicht (Taxi-Agent)
- Selbstlokalisierung (Staubsaug-Agent)

Simple reflex agent → model-based reflex agent

Simple reflex agent

- handelt ausschliesslich auf der Basis der aktuellen Wahrnehmung
- ≈ die Welt, wie sie der Agent momentan wahrnimmt

Desiderata

- Berücksichtigung
 - gewisser Teile / Aspekte der Umgebung, die gegenwärtig nicht wahrgenommen werden
 - der Umstände / der Historie, die dazu geführt haben, dass die Umgebung so ist, wie sie ist.

Model-based reflex agent

- mit Weltmodell (≈ Gedächtnis)

Simple reflex agent: Struktur des Agenten-Programms

Function SIMPLE-REFLEX-AGENT(*percept*) returns an action

static: *rules*, a set of condition-action rules

state ← INTERPRET-INPUT(*percept*)

rule ← RULE-MATCH (*state*, *rules*)

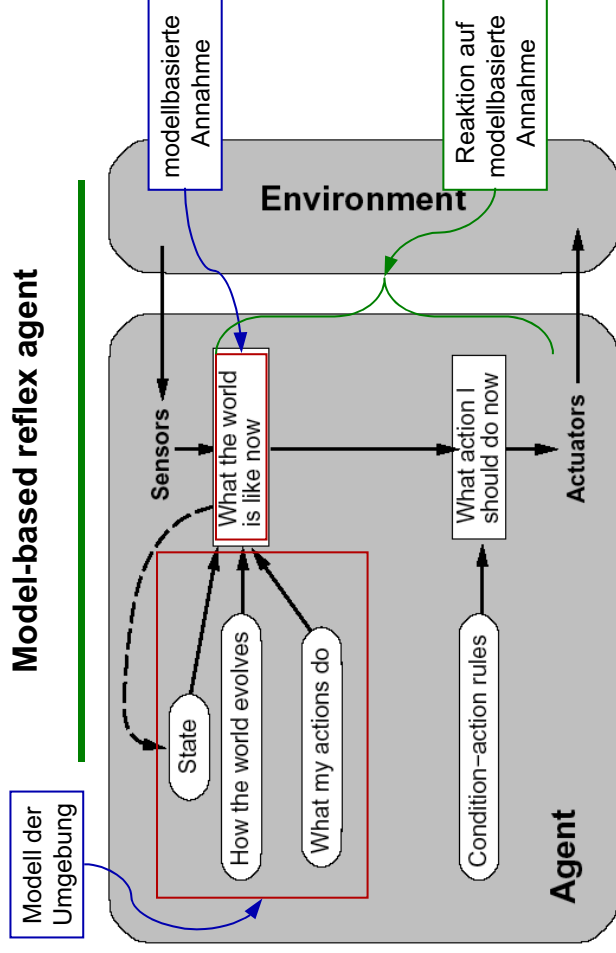
action ← RULE-ACTION (*rule*)

return *action*

Leistung des Agenten

- ist abhängig von der vollständigen Beobachtbarkeit der Umgebung

- Interpretation von Rücklicht/Bremslicht (Taxi-Agent)
- Selbstlokalisierung (Staubsaug-Agent)



Modell der Welt im model-based reflex agent

Weltmodell (betrifft Umgebung und ihre Historie)

- basiert auf Historie der Wahrnehmung
- berücksichtigt daher auch Teile der Umgebung, die gegenwärtig nicht wahrgenommen werden (UPDATE früherer Zustände des Modells)
- berücksichtigt die Wirkung der eigenen (vergangenen und aktuellen) Handlungen

Berechnung des Weltmodells

- beruht auf Wissensrepräsentation und Wissensverarbeitung
- kann u.U. sehr komplexe Teilprobleme beinhalten, z.B. Persistenzprobleme

Model-based reflex agent → Goal-based agent

Model-based reflex agent

- handelt auf der Basis eines Modells des Welt, „wie sie gegenwärtig ist“.

Desiderata

- Berücksichtigung
 - gewisser Teile / Aspekte der Umgebung, die in (naher) Zukunft der Fall sein werden, insbesondere der Veränderungen, die sich auf Grund der Handlungen des Agenten ergeben
 - Der Ziele, die der Agent hat / zu erreichen versucht.

➤ Goal-based agent

- mit Prognose- und Planungsfähigkeiten

Goal-based agent → Utility-based agent

Goal-based agent

- Handelt, um ein (vorgegebenes) Ziel zu erreichen.

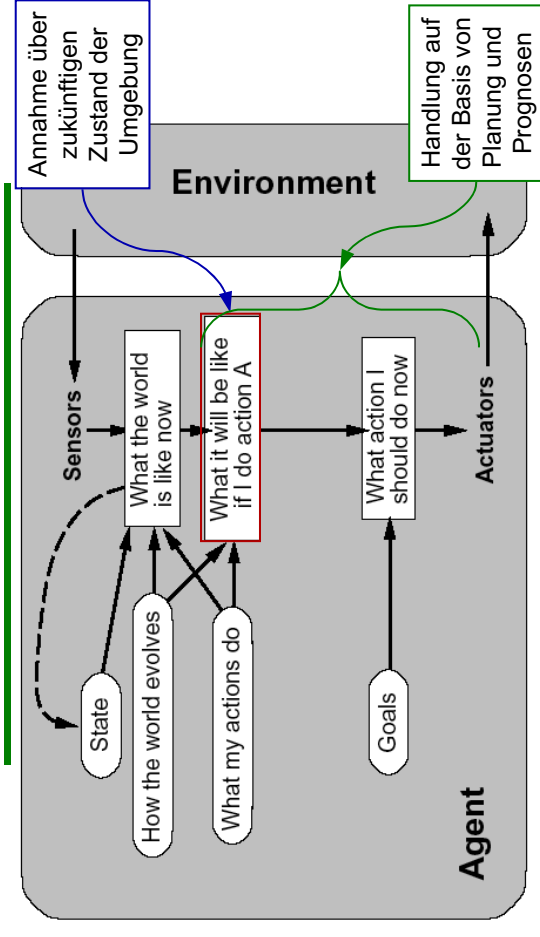
Desideratum

- Wie wird zwischen konkurrierenden Zielen entschieden, welches das vorrangige / wichtigste Ziel ist?
- Wie wird bei mehreren kompatiblen Zielen, ein „Gesamtziel“ gefunden.

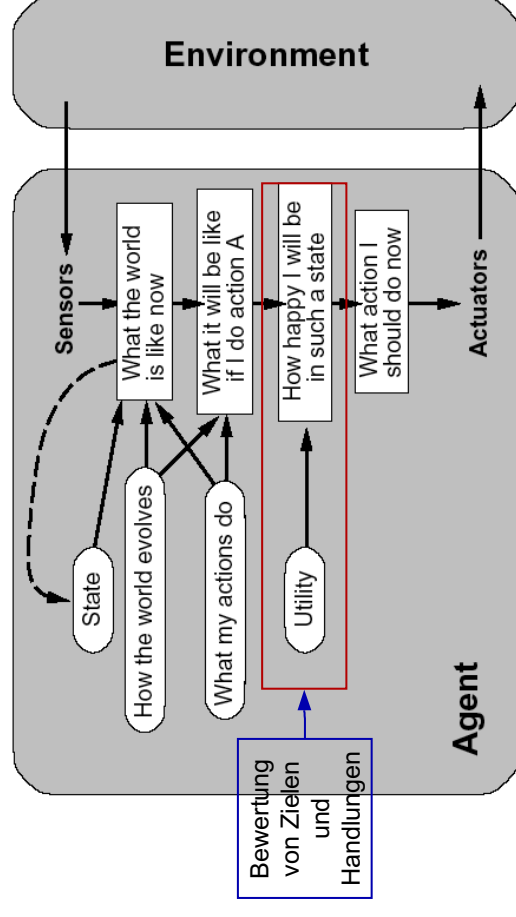
➤ Utility-based agent

- mit Fähigkeiten Ziele zu bewerten, zwischen Zielen auszuwerten, Zielkombinationen zu entwickeln.

Goal-based agent



Utility-based agent

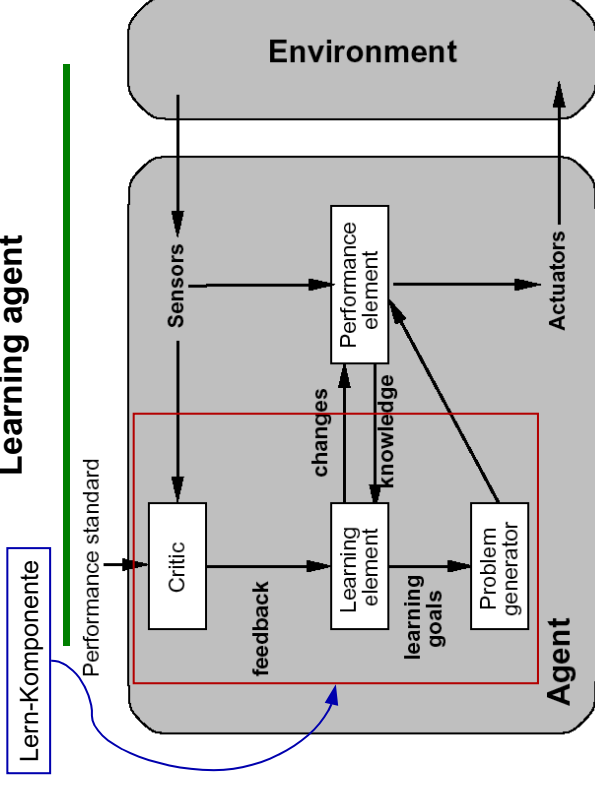


Typen von Agenten-Architekturen

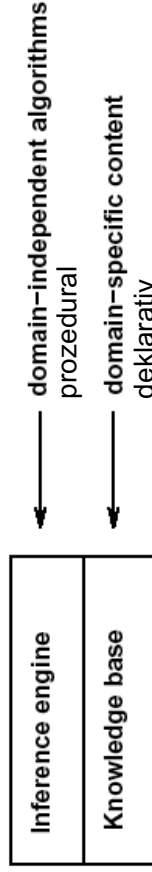
Vier grundlegende Architekturen

- Simple reflex agents
 - Model-based reflex agents
 - Goal-based agents
 - Utility-based agents
- Handlungsfähigkeit
- „Performanzkomponente“
- +
- „Lernkomponente“

Learning agent



Wissensbasierte Agenten (1): Wissensbasen & Inferenzmaschinen



- **Wissensbasis** Menge von **Sätzen** in einer **formalen Sprache**
- **Inferenzmaschine** Realisierung von **Ableitungsverfahren**
- **TELL deklarative Instruktion der Wissensbasis**
ASK Anfrage an die Wissensbasis
 - Beide interne Aktionen machen von Wissensbasis und Inferenzen Gebrauch

Wissensbasierte Agenten (knowledge-based agents)

function $KB-AGENT(percept)$ returns an *action*
 static: KB , a knowledge base

t , a counter, initially 0, indicating time

$TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))$

$action \leftarrow ASK(KB, MAKE-ACTION-QUERY(t))$

$TELL(KB, MAKE-ACTION-SENTENCE(action, t))$

$t \leftarrow t + 1$

return *action*

MAKE-PERCEPT-SENTENCE

→ Repräsentation, dass der Agent zur Zeit t *percept* wahrnimmt.

MAKE-ACTION-QUERY

→ Frage nach geeigneter Aktion zu t

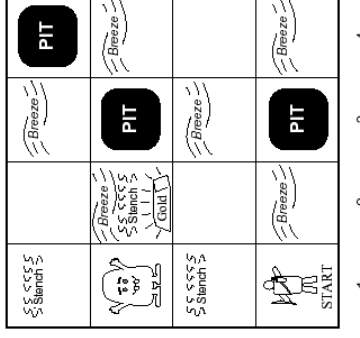
MAKE-ACTION-SENTENCE

→ Repräsentation, dass der Agent zur Zeit t *action* ausführen will

WUMPUS World

Eine *Höhlenwelt*, bestehend aus:

- Ein Agent, der Gold sucht. Er besitzt einen Bogen mit einem Pfeil.
- Ein Monster (Wumpus), der jeden frisst, der seine Kammer betritt.
- Fallgruben, in die der Agent fallen kann, aber nicht der Wumpus.



Eine WUMPUS
Start-Konfiguration

Ch. Habel / C. Eschenbach: Wissensrepräsentation, SoSe 2003

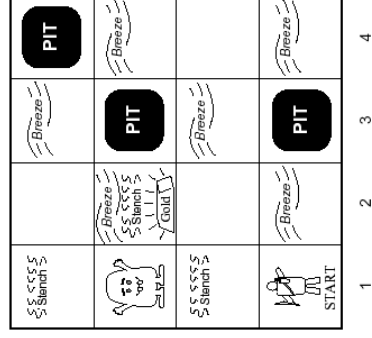
3 – 17

Russell & Norvig - ch.6, f.5

WUMPUS World - Die PEAS Beschreibung (2)

Aktuatoren:

- **Fortbewegung:**
 - vorwärts (hat keinen Effekt, wenn ein Wand vor dem Agenten ist.)
 - drehen um 90° (links / rechts)
- **Sterben**, beim betreten eines Raumes
 - in dem eine Fallgrube ist,
 - in dem der lebendige Wumpus ist.
- **Greifen** des Goldes
- **Schiessen** des einzigen Pfeils
 - vorwärts (bis es den Wumpus oder eine Wand trifft.)



Ch. Habel / C. Eschenbach: Wissensrepräsentation, SoSe 2003

3 – 19

Russell & Norvig - ch.6, f.5

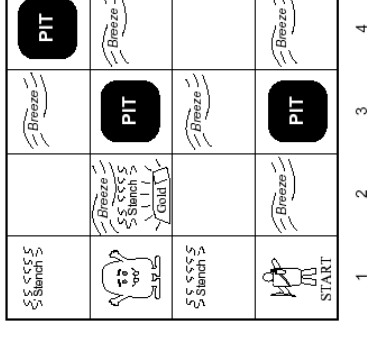
WUMPUS World - Die PEAS Beschreibung (1)

Umgebung:

- 4 x 4 Layout der Kammern, alle Nachbarzellen sind zugänglich.
- In Kammern, die
 - an den Wumpus angrenzen, oder in denen sich der Wumpus befindet, riecht es.
 - an eine Fallgrube angrenzen, zieht es.
- **Start immer im Feld [1,1]**

Design der Umgebung:

20% der Kammern (ausser [1,1]) besitzen eine Fallgrube.



Ch. Habel / C. Eschenbach: Wissensrepräsentation, SoSe 2003

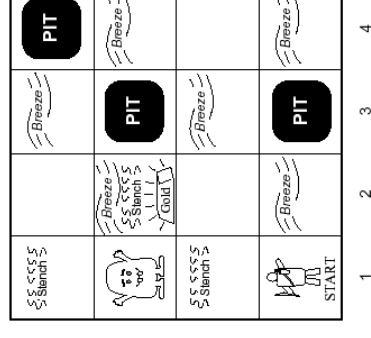
3 – 18

Russell & Norvig - ch.6, f.5

WUMPUS World - Die PEAS Beschreibung (3)

Sensoren:

- **Olfaktorisch/Geruch:** Wahrnehmung des Wumpusgeruchs
- **Luftbewegung:** Wahrnehmung des Luftzuges
- **Visuell:** Glitzern des Goldes
- **Taktil:** Wahrnehmung von Hindernissen (Wände)
- **Auditiv / Gehör:** Wahrnehmung des Aufschreis, wenn der Wumpus vom Pfeil getroffen wird (hörbar in der gesamten Umgebung).



Die Sensorik ist lokal, d.h. auf eine Kammer beschränkt.

Wahrnehmungsquintupel: [O,L,V,T,A]

Ch. Habel / C. Eschenbach: Wissensrepräsentation, SoSe 2003

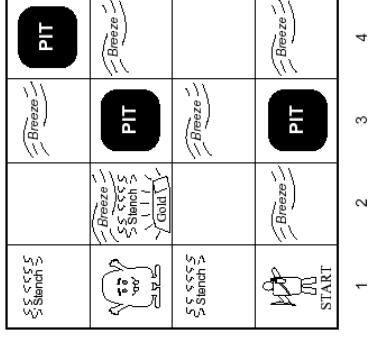
3 – 20

Russell & Norvig - ch.6, f.5

WUMPUS World - Die PEAS Beschreibung (4)

Performanzmaße:

- + 1000 Finden des Goldes
- 1000 in die Fallgrube fallen bzw. vom Wumpus gefressen werden
- 10 den einzigen Pfeil verwenden
- 1 eine andere Handlung ausführen

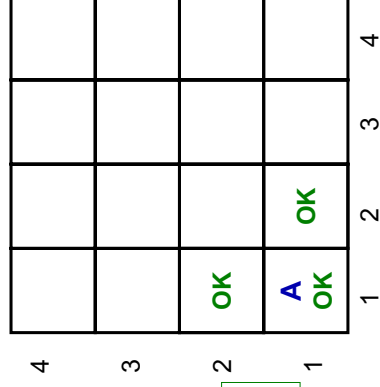


Exploration einer WUMPUS-Welt (1)

Legende:

- A Agent
- B Breeze
- G Gold, glitter
- OK Safe square
- P Pit
- S Stench
- V Visited
- W Wumpus

Startsituation: t0



Erste
Schlüsse

Wahrnehmung: [none, none, none, none]

WUMPUS World : Eigenschaftsanalyse der Umgebung

Beobachtbarkeit der Umgebung	Nein nur lokale Wahrnehmung
Vorhersehbarkeit der Umgebung	Deterministisch: Resultat ist exakt spezifiziert
Abhängigkeit der Handlungen des Agenten	Sequentiell
Dynamik der Umgebung	Statisch: Wumpus und Fallgruben sind statisch.
Interne Struktur	Diskret
Anzahl der Agenten	Single - agent: Wumpus ist eher Teil der Umgebung.

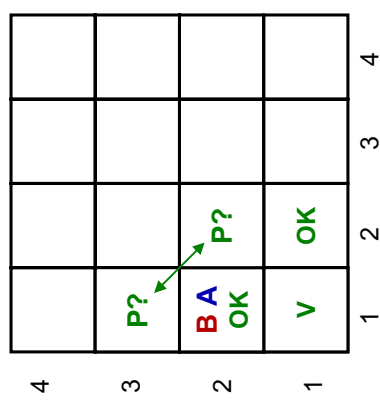
Exploration einer WUMPUS-Welt (2)

Aktion: zu [1,2] gehen

Wahrnehmung:
[none, B, none, none]

Schlüsse

Situation: t1



Exploration einer WUMPUS-Welt (3)

Aktionen: über [1,1] zu [2,1] gehen

Wahrnehmung: [S, none, none, none, none]

Schlüsse

Situation: t3

4				
3	P			
2	B V	OK		
1	V → ← S	A OK	W	
	1	2	3	4

Exploration einer WUMPUS-Welt (5)

Aktionen: zu [2,3] gehen

Wahrnehmung: [S, B, G, none, none]

Schluss: Gold ist hier

Aktion: Gold greifen

Situation: t5

4				
3	P	OK		
2	B V	V	SBG A OK	
1	V	S V	W	
	4	3	2	1

Exploration einer WUMPUS-Welt (4)

Aktion: zu [2,2] gehen

Wahrnehmung: [none, none, none, none, none]

Schlüsse

Situation: t4

4				
3	P	OK		
2	B V	A OK	OK	
1	V	S V	W	
	4	3	2	1

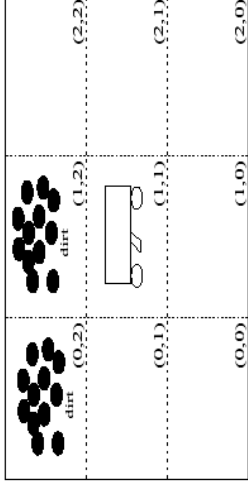
Weitere Literatur zu Agenten

Michael Wooldridge (2002).
An Introduction to MultiAgent Systems.
 Chichester: John Wiley & Sons

Folien zur MultiAgenten-Vorlesung von M. Wooldridge:
<http://www.csc.liv.ac.uk/~mjw/pubs/imas/>

Aufgabe: Staubsauger-Welt (1)

Goal is for the robot to clear up all dirt.



- **Wie ist diese Welt zu spezifizieren, um einen Staubsauger-Agenten zu realisieren?**

Aufgabe: Staubsauger-Welt (2)

- Use 3 domain predicates in this exercise:

$In(x, y)$ agent is at (x, y)
 $Dirt(x, y)$ there is dirt at (x, y)
 $Facing(d)$ the agent is facing direction d

- Possible actions:

$$Ac = \{turn, forward, suck\}$$

NB: *turn* means “turn right”.

Rules of the following type:

$$In(0, 0) \wedge Facing(north) \wedge \neg Dirt(0, 0) \longrightarrow Do(forward)$$