

## Wissensrepräsentation

Christopher Habel, Carola Eschenbach  
Universität Hamburg, FB Informatik  
AB Wissens- und Sprachverarbeitung (WSV)

Sommersemester 2003

## Wissensrepräsentation

Christopher Habel, Carola Eschenbach  
Sommersemester 2003

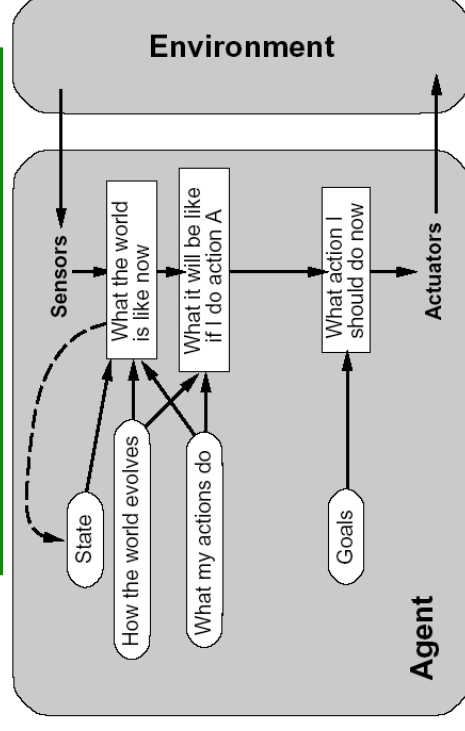
### Sitzung 24: Wissen über Aktionen

- Situationskalkül
- Spezifikationen von Aktionen
- Erweiterung: Golog: Programmiersprache für Agenten
- Instruktionsbasiertes Handeln

## Literatur

Levesque, Hector J., Raymond Reiter, Yves Lespérance, Fangzhen Lin & Richard B. Scherl (1997). GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming* 31. 59–84.  
Brachman, Ronald J. & Hector J. Levesque (to appear). *Knowledge Representation and Reasoning*. Chapter 14.

## Goal-based agent



## Interne Repräsentationen des Wissensbasierten Agenten

### Zeitunabhängig

- Hintergrundwissen

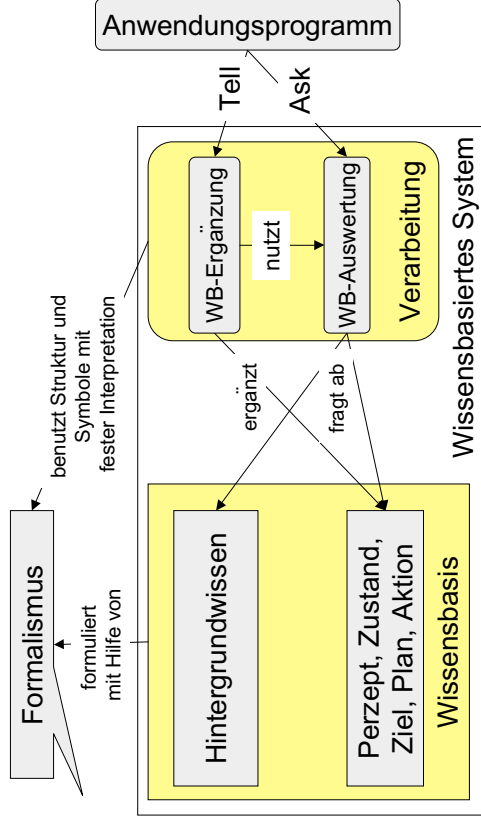
### Zeitabhängig

- Perzepte: Rep. den aktuellen Weltzustand (relativ zur Sensorik)
- (int.) Zustand: Rep. der durchlaufenen Historie
- Ziel: Rep. den aktuell angestrebten Weltzustands
- Plan: Rep. der intendierten Handlungsfolge
- Aktion: Rep. der mit der Aktorik (in der Welt) auszuführenden Handlung

### Veränderung des Weltzustandes

- aufgrund von Handlungen des Agenten

## Wissensbasiertes System



## Hintergrundwissen über Handlungen

### für Aktionsauswahl, Planung

#### Erforderlich

- Voraussetzungen der Handlung
- Effekte der Handlung

#### Nützlich

- Kosten (Zeit, ...)
- Verlässlichkeit

## Wissen über Aktionen, Handlungen

### Legalitätsfeststellung

- Ist eine Aktion in einer gegebenen Situation ausführbar ?
- Ist eine Aktionsfolge ausführbar ? (Wenn ja: In welchen Situationen ?)

### Vorhersage

- In welchen Weltzustand führt die Aktion / Aktionsfolge von einer gegebenen Situation ?

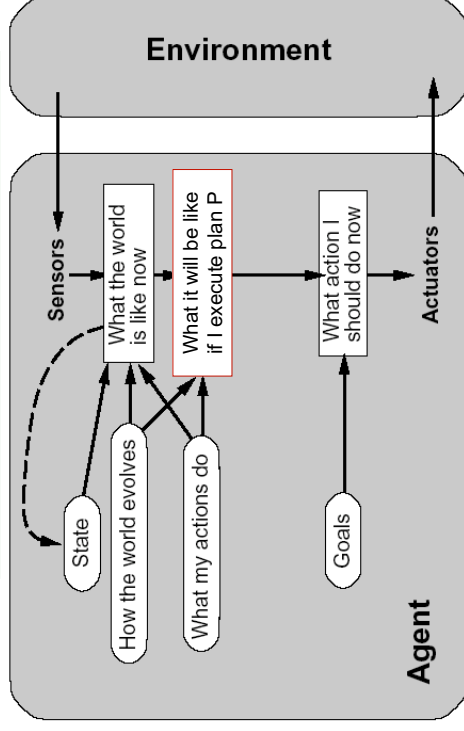
### Aktionswahl

- Welche Aktion bringt mich meinem Ziel näher?

### Planung

- Welche Aktionsfolge führt in einen Weltzustand, der meiner Zielbeschreibung entspricht ?

## Planning Goal-based agent



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

24 – 9

Russell & Norvig - ch.2, f.21

## Zeitabhängigkeit

### Repräsentationen von Wissen enthalten

- zeitunabhängige und
- zeitabhängige Anteile

### Unterscheidungsmöglichkeiten

- Trennung in der ‚Architektur‘ (STRIPS)
- Zeitbezug in den Formeln explizit machen

### Aktionen / Handlungen

- ‚treiben die Zeit voran‘
- ‚bewirken, welcher Zustand einem vorherigen folgt‘

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

24 – 10

## Handlungsbeschreibung in STRIPS (1971)

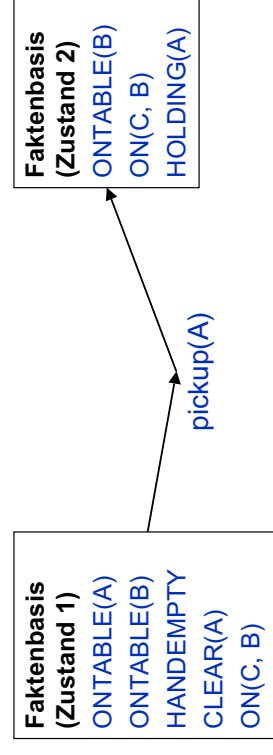
### Aktionen als Produktionen

- Name:  $\text{pickup}(x)$
- LHS: **preconditions**: Konjunktion von atomaren Formeln
  - $\text{ONTABLE}(x) \wedge \text{HANDEMPY} \wedge \text{CLEAR}(x)$
- RHS: **Delete** list: atomare Formeln:
  - $\text{ONTABLE}(x), \text{HANDEMPY}, \text{CLEAR}(x)$
- RHS: **Add** formula: atomare Formel:  $\text{HOLDING}(x)$

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

24 – 11

## Beispiel: Produktionsausführung in STRIPS

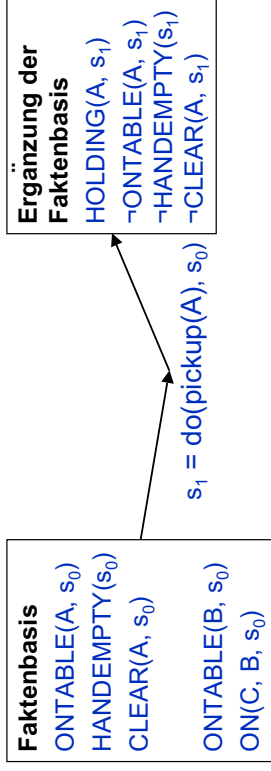


- Die Zustände 1 und 2 der Faktenbasis repräsentieren verschiedene Weltzustände
- Es besteht keine (einfache) logische Folgerungsbeziehung.

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

24 – 12

## Beispiel: Situationskalkül



- Die  $s_0$  und  $s_1$  repräsentieren verschiedene Weltzustände
- Die Handlungsbeschreibungen können nicht explizit machen, was die Handlung alles unverändert lasst

## Handlungsbeschreibung im Situationskalkül

### Spezifikationen von Aktionen

- Name: pickup(x)
- **preconditions**  
 $\text{Poss}(a, s)$ : a kann in s ausgeführt werden
  - $\forall x, s [\text{Poss}(\text{pickup}(x), s) \Leftrightarrow \text{ONTABLE}(x, s) \wedge \text{HANDEMPY}(s) \wedge \text{CLEAR}(x, s)]$
- **effects**  
 $\text{do}(a, s)$ : Bezeichnung der Situation, die das Ausführen von a in s herbeiführt
  - $\forall x, s [\neg \text{ONTABLE}(x, \text{do}(\text{pickup}(x), s)) \wedge \neg \text{HANDEMPY}(\text{do}(\text{pickup}(x), s)) \wedge \neg \text{CLEAR}(x, \text{do}(\text{pickup}(x), s)) \wedge \text{HOLDING}(x, \text{do}(\text{pickup}(x), s))]$

## Situationskalkül

### Basisannahmen

- Nur der Agent verändert durch seine Aktionen die Welt
  - keine Berücksichtigung der Wahrnehmung
- Zeit: nur implizit über Aktionsfolgen
  - keine Dauer
  - keine kontinuierlichen Prozesse
  - keine Parallelität

### Ziel

- Aktionsauswahl
- Planung
- Vorhersage

### Aufhebung der Annahmen

- durch übergeordnete Mechanismen: z.B. Golog

## Situationskalkül: Prädikatenlogik mit

### neuen Sorten von Objekten

- (primitive) **Aktionen**: Anwendungsabhängig
  - **Situationen**: mögliche Historien (Initialzustand der Welt und Sequenzen von Aktionen davon ausgehend)
- ### Prädikaten und Relationssymbolen
- ohne Situationsargument
    - unveränderliche Eigenschaften, Beziehungen **heavy(x)**
    - mit Situationsargument (Konvention: an letzter Position)
      - veränderliche Eigenschaften Beziehungen **ontable(x, s)**
    - **Fluents**
  - Ausgezeichneter Fluent

- **Poss(a, s)**: In Situation s kann a ausgeführt werden
- zur Spezifikation von Vorbedingungen von Aktionen

## Situationskalkül: (syntaktische) Varianten

### Expliziter oder impliziter Agent

- Aktionsterme können den Agenten explizit benennen  
 $\text{Poss}(\text{pickup}(r, x), s) \Leftrightarrow \forall z [\neg \text{heavy}(x) \wedge \text{nextTo}(r, x, s)]$
- Aktionsterme können den Agenten implizit lassen (wenn es nur einen gibt)  
 $\text{Poss}(\text{pickup}(x), s) \Leftrightarrow \forall z [\neg \text{heavy}(x) \wedge \text{nextTo}(x, s)]$

### Fluents als Terme

- werden mit ausgezeichnete Relation Holds zu Situationen in Beziehung gesetzt  
 $\text{Poss}(\text{pickup}(x), s) \Leftrightarrow \forall z [\neg \text{heavy}(x) \wedge \text{Holds}(\text{nextTo}(x), s)]$
- erfordern die Duplizierung logischer Symbole für die Bildung komplexer Fluents  
 $\text{Holds}(\Phi \ \& \ \Psi, s) \Leftrightarrow \text{Holds}(\Phi, s) \wedge \text{Holds}(\Psi, s)$

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

24 – 17

## Wissen über Aktionen im Situationskalkül

### Legalitätsfeststellung

- Ist Aktion  $a$  in Situation  $s$  ausführbar ?

$$\mathcal{KB} \models \text{Poss}(a, s)$$

### Vorhersage

- Gilt Fluent  $\Phi$  nach Ausführung von Aktion  $a$  in Situation  $s$  ?  
 $\mathcal{KB} \models \Phi(\text{do}(a, s))$

### Verallgemeinerbar

- Für Aktionssequenzen

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

24 – 18

## Wissen über Aktionssequenzen im Situationskalkül

### Es sei

- $\langle a_1, \dots, a_n \rangle$  eine Sequenz von primitiven Aktionen
- Kurzschreibweisen
  - $\text{do}(\langle \rangle, s) = s$
  - $\text{do}(\langle a_1, \dots, a_n \rangle, s) = \text{do}(\langle a_2, \dots, a_n \rangle, \text{do}(a_1, s))$
  - $\text{Legal}(\langle \rangle, s) = \text{T}$
  - $\text{Legal}(\langle a_1, \dots, a_n \rangle, s) = \text{Legal}(\langle a_1, \dots, a_{n-1} \rangle, s) \wedge \text{Poss}(a_n, \text{do}(\langle a_1, \dots, a_{n-1} \rangle, s))$

### Legalitätsfeststellung

- Ist  $\langle a_1, \dots, a_n \rangle$  in Situation  $s$  ausführbar ?  
 $\mathcal{KB} \models \text{Legal}(\langle a_1, \dots, a_n \rangle, s)$

### Vorhersage

- Gilt Fluent  $\Phi$  nach Ausführung von  $\langle a_1, \dots, a_n \rangle$  in  $s$  ?  
 $\mathcal{KB} \models \Phi(\text{do}(\langle a_1, \dots, a_n \rangle, s))$

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

24 – 19

## Spezifikation primitiver Aktionen: Aufzug

(Levesque et al. 1997)

### Primitive actions

- $\text{turnoff}(n)$  % Turn off call button  $n$ .
- $\text{open}$  % Open elevator door.
- $\text{close}$  % Close elevator door.
- $\text{up}(n)$  % Move elevator up to floor  $n$ .
- $\text{down}(n)$  % Move elevator down to floor  $n$ .

### Primitive Fluents

- $\text{on}(n, s)$  % unanswered call on floor  $n$
- $\text{currentFloor}(n, s)$  % current floor has number  $n$

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

24 – 20

## Preconditions for Primitive Actions

(Levesque et al. 1997)

*Pre*<sub>EL</sub>

- $\forall s, n$  [  $\text{poss}(\text{turnoff}(n),s) \Leftrightarrow \text{on}(n, s)$  ] % nur bestehende %Anforderungen
- $\forall s$  [  $\text{poss}(\text{open},s)$  ] % keine Vorbedingungen
- $\forall s$  [  $\text{poss}(\text{close},s)$  ] % keine Vorbedingungen
- $\forall s, n, m$  [  $\text{poss}(\text{up}(n),s) \Leftrightarrow \text{currentFloor}(m, s) \wedge m < n$  ] % up nur in höhere Etagen
- $\forall s, n, m$  [  $\text{poss}(\text{down}(n),s) \Leftrightarrow \text{currentFloor}(m, s) \wedge m > n$  ] % down nur niedriger Etagen

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

24 – 21

## Effect Axioms for Primitive Fluents

• *Eff*<sub>EL</sub>

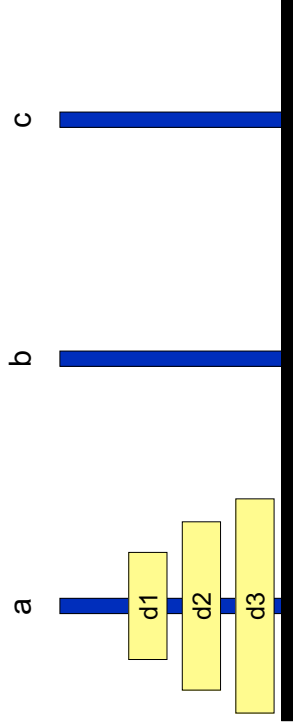
- **Positive Effekte (vgl. STRIPS Add-List)**  
 $\forall s, m$  [  $\text{currentFloor}(m, \text{do}(\text{up}(m), s))$  ] % Positions  
 $\forall s, m$  [  $\text{currentFloor}(m, \text{do}(\text{down}(m), s))$  ] % Positions
- **Negative Effekte (vgl. STRIPS Delete-List)**  
 $\forall s, m$  [  $\neg \text{on}(m, \text{do}(\text{turnoff}(m), s))$  ] % Anforderung  
 $\forall s, n, m$  [  $n \neq m \Rightarrow \neg \text{currentFloor}(m, \text{do}(\text{up}(n), s))$  ] % Position  
 $\forall s, n, m$  [  $n \neq m \Rightarrow \neg \text{currentFloor}(m, \text{do}(\text{down}(n), s))$  ] % Position
- open und close haben keine Effekte  
→ Revision

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

24 – 22

## Aktionen für den Turm von Hanoi

Gruppen-  
diskussion



Welche Relationen, Fluents, Aktionen werden benötigt?  
Welche Vorbedingungen und Effekte haben die Aktionen?

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

24 – 23