
Wissensrepräsentation

—
Christopher Habel, Özgür Özçep
Sommersemester 2005

Sitzung 6: Frames

- Objektzentrierte Repräsentation: Die Idee
- Basic Frame Language

Literatur:

Brachman, R. J. & Levesque, H. J. (2004). Knowledge Representation and Reasoning. San Francisco, CA: Morgan Kaufmann. (Kapitel 8)

Objektzentrierte Repräsentationen

Historie:

Minsky, M. (1974). A Framework for Representing Knowledge. Report. MIT: AI-Lab. Memo 306.

Teilweise abgedruckt in:

- P.H. Winston (ed.) (1975). *The psychology of computer vision*. McGraw Hill: New York. (pp. 211–77)
- J. Haugeland (ed.) (1981). *Mind Design*. Cambridge, MA: MIT-Press. (pp. 95–128)

Objekt-orientierte Programmierung:

Goldberg, A. & Robson, D. (1983). Smalltalk-80: The language and its implementation. Addison-Wesley: Reading, MA.

Objektzentrierung: Organization von Fakten- und Regelmengen

Fakten und Regeln

- satzartige (flache) Repräsentationen
- Information über Entitäten (Objekte) ist meist über mehrere Sätze (Fakten / Regeln) verteilt

Organisation von Fakten und Regeln

- Gruppierung in Mengen
- möglich Gruppierungskriterien
 - Anwendungskontexte
 - beteiligte Entitäten

➤ Objektzentrierte Repräsentationen

Objektzentrierte Repräsentationen: Die Idee

Konzepte:

Menschliches Schliessen & Problemlösen ist konzept-geleitet.

Most obvious, natural organizational technique depends on our ability to see the world in terms of objects.

[B&L, 2004]

- **(physikalische) Objekte:** z.B. Schreibtische, Autos, Menschen, Berge, Räume,...
- **Situationen:** Vorlesungen, Reisen, Konzertbesuche, Diplomarbeit_schreiben, ...

➤ Objektzentrierte Repräsentation

- Prozeduren, zur Bestimmung von Eigenschaften, Identifikation von Teilen, Interaktion von Teilen, ...
 - Bedingungen für die Kombination & Interaktion von Objekten und Teilen von Objekten.
-

Basic Frame Language

Frame

- ist eine – durch einen Namen individuierte – Liste von *Slots*
 - Slots sind durch einen Namen individuiert
 - Slots können durch Werte (*filler*) gefüllt werden
- Frame schema:
(Frame-name
 <slot-name1 value1>
 <slot-name2 value1 > ...)
- Slots können Verweise auf andere Frames enthalten.
(tripLeg123-1
 <:Destination toronto>...)
(toronto
 <:Province ontario>
 <:Population 4.5M>...)

Generic frames – individual frames

Generic frames

≈ Kategorien / Klassen von Objekten

Individual frames

≈ Instanzen von Objekten

```
(tripLeg123-1  
 <:INSTANCE-OF TripLeg>  
 <:Destination toronto>...)  
(toronto  
 <:INSTANCE-OF CanadianCity>  
 <:Province ontario>  
 <:Population 4.5M>...)  
(CanadianCity  
 <:IS-A City>  
 <:Province CanadianProvince >  
 <:Country canada>...)
```

GenFr-Namen

beginnen mit
Grossbuchstaben

IndFr-Namen

beginnen mit
Kleinbuchstaben

Slot-Namen

beginnen mit „:“

INSTANCE-OF Verweis
von IndFr zu GenFr

IS-A Verweis von GenFr
zu GenFr

Procedural attachment

• Berechnung von Werten, falls benötigt

```
(Person  
 <:Age  
 [:IF-NEEDED ComputeAgeFromDateOfBirth] >...)
```

• Propagierung von Bedingungen, wenn Werte zugewiesen werden

```
(Lecture  
 <:DayOfWeek WeekDay>  
 <:Date [:IF-ADDED ComputeDayOfWeek] >...)
```

• Vererbung

Weiterleitung von „Default-Werten“ mittels der
INSTANCE-OF und **IS-A** Verweise.

Schliessen mit Frames

Basic (local) reasoning:

1. user “instantiates” a frame, i.e., declares that an object or situation exists
2. slot values are inherited where possible
3. user repeats (1) or one of
 - a) adds a value to a slot
(– restrictions are checked)
– IF-ADDED’s are run
 - b) asks for a slot value
– if it’s there, return it
– if not, if there’s a DEFAULT,
return it (run a
DEFAULTCALC if
there is one)
– otherwise, run IF-NEEDED’s

Globally:

- make frames be major situations or object-types you need to flesh out
- express constraints between slots as IF-NEEDED and IF-ADDED procedures
- fill in default values when known

Beispiel: Reiseplanung

Reise ist eine Sequenz von Reise-Schritten:

Generic Frame

```
(Trip
  <:FirstStep TravelStep>
  <:Traveller Person>
  <:BeginDate Date>
  <:EndDate Date>
  <:TotalCost Price>
  ...)
```

Individual Frame

```
(trip17
  <:INSTANCE-OF Trip>
  <:FirstStep travelStep17a>
  <:Traveller ronB>
  <:BeginDate 11/13/98>
  <:EndDate 11/18/98 >
  <:TotalCost $1752.45>
  ...)
```

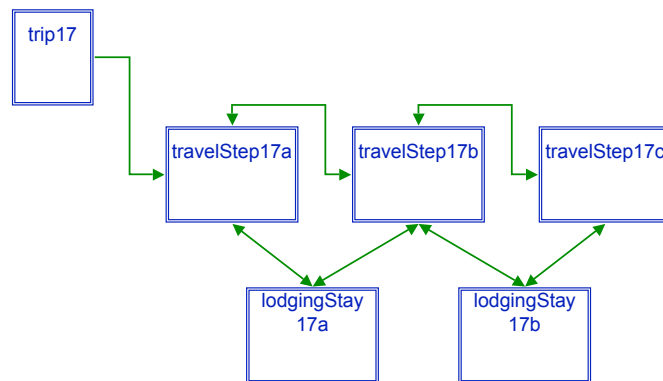
Reiseplanung (2): weitere Frames

```
(TripPart
  <:BeginDate Date>
  <:EndDate Date>
  <:Cost Price>
  ...)
```

```
TravelStep
  <:IS-A TripPart>
  <:Origin City>
  <:Destination City>
  <:OriginLodgingStay
    LodgingStay>
  <:DestinationLodgingStay
    LodgingStay>
  <:DepartureTime Time>
  <:ArrivalTime Time>
  <:NextStep TravelStep>
  <:PreviousStep TravelStep>
  <:Cost Price>
  ...)
```

```
(LodgingStay
  <:IS-A TripPart>
  <:Place City>
  <:ArrivingTravelStep TravelStep>
  <:DepartingTravelStep TravelStep>
  <:Cost Price>
  ...)
```

Reiseplanung (3): Abhängigkeiten zwischen Frames



Aufgabe: Reiseplanung mit Frames

Präsenz- bzw.
Nacharbeitsaufgabe

- Spezifizieren Sie die Abhängigkeit zwischen Reise-Abschnitten (TripSteps) mittels von **IF-ADDED** und **IF-NEEDED** Prozeduren.
- An welchen Stellen sind Defaults notwendig / sinnvoll?
- Testen Sie Ihre Spezifikation anhand der Planung einer Zweit-Tages-Ein-Übernachtungsreise.

(weitere) Literatur zu *Frames*

Bobrow, D. & Winograd, T. (1977). An overview of KRL, a knowledge representation language. *Cognitive Science*, 1. 3–46.

Hayes, P. (1980). The logic of frames. In D. Metzger (ed.), *Frame Conceptions and Text Understanding*. (pp. 46-61.) de Gruyter: Berlin.

Fikes, R. (1981). *Odyssey: A Knowledge-Based Assistant*. *Artificial Intelligence*, 16. 331–361.

Brachman, R.; Fikes, R.; & Levesque, H. (1983). KRYPTON: A functional approach to knowledge representation. *IEEE Computer*, 16. 67–73.

Wie es weiter geht

Produktionssysteme und Frame:

Probleme & Desiderata

- Strukturierung / Organization der Wissensbestände
- Wohldefinierte Semantik
bzw. systematische Beziehung zwischen
Schlussverfahren und Semantik
- „Handhabbare“ (effiziente) Wissensverarbeitung

➤ Beschreibungslogiken

➤ Constraint-Verfahren