

Kapitel 9: Frames

(Dieser Foliensatz basiert auf Material von Mirjam Minor, Humboldt-Universität Berlin, WS 2000/01, sowie von Andreas Abecker/Knut Hinkelmann)

Erwartungsgesteuerte Verarbeitung

- Die Idee der Frames basiert auf der Annahme, dass unsere Informationsverarbeitung wesentlich von unseren Erwartungen abhängt [Minsky, 1981].
- Minsky geht davon aus, dass immer dann, wenn eine neue Situation auftritt oder wenn sich die Sicht auf ein Problem gravierend ändert, man im Gedächtnis nach passenden Situationen sucht.
- Die gespeicherte Situation wird auf die neue Situation angepasst.

- Aus der bekannten Situation ergeben sich Erwartungen über die neue Situation.
- Beispiele: Restaurantbesuch, Geburtstagsparty, Wohnzimmer
- Frames sind eine Datenstruktur zur Repräsentation stereotyper Situationen.

Vererbung und Defaults

- Zu den **Rollen** eines Konzeptes gehören auch solche, die Verweise zu Ober- und Unter-Konzepten widerspiegeln, d.h. Vererbung ist möglich.
- Es gibt die Möglichkeit zur Angabe von **Default-Werten**, die später überschrieben werden können (z.B. Anzahl der Sofas = 2).
- Auch **geerbte (generische) Werte** können überschrieben werden.

```

[Mensch
  superclass: lebewesen
  geschlecht:
    type: männlich ODER weiblich
    maxKardinalität: 1
  vorname:
    type: string
  name:
    type: string
  alter
    type: integer
  eltern
    type: mensch max Kardinalität: 2
]

[Mann
  superclass: Mensch
  geschlecht: männlich ]

[Vater
  superclass: Mann
  kinder:
    type: mensch minKardinalität: 1
]

[Reise
  reisender:
    type: mensch
  ermäßigung:
    type: [0..100]
    proc: if reisender.alter < 18
      then 50
      else if reisender.alter >
        then 25
        else 0
    exec: if-needed
]

[GS
  member-of: Vater
  vorname: "Gerd"
  alter: 42
]

[r123
  member-of: Reise
  reisender: GS
]

```

```

[rechteck
  seite1
    type: real
  seite2
    type: real
  fläche
    type: real
    proc: (seite1 * seite2)
    exec: if-needed
]

[quadrat
  superclass: rechteck
  seite2
    type: real
    proc: seite1
    exec: if-added
]

[q1
  superclass: quadrat
  seite1: 5
]

```

Begriffe zur Darstellung eines Frames

Frame: Liste von Slots.

Slots (Schlitze) sind gegliedert in Facetten.

Eine **Facette** nimmt Füllsel (fillers) auf.

Ein Slot hat mindestens zwei Facetten: Slot-Name und Slot-Wert, ggf. weitere, z.B. für Defaultwert.

Gebräuchliche Facetten (I)

Value: eigentlicher Slotwert, speziell auch als Verweis(e) auf andere Frames:

- instance-of, is-a
- ako
- part-of bzw. has-part
- supersets (instance-of, is-a, ako)
- subsets (Nachfolger in der Hierarchie)
- Parallelverweise (Verheiratet mit ...)

Defaultwerte (des Frames)

Generische Werte (für Vererbung)

Gebräuchliche Facetten (II)

Slotbedingungen: zulässige Wertebereiche

Prozeduren (Dämonen):

- if-needed (Berechnungen, „Lese-Prozeduren“, z.B. Alter aus Geburtsdatum und aktueller Zeit)
- if-added (Aktionen, „Schreib-Prozeduren“) in Verbindung mit belegen/ändern eines Slotwertes, z.B. bei Änderung des Familienstandes auch Steuerklasse ändern („aktive Datenbank“)

Vererbungshinweise (Vererbungsrolle)

Problemlose Behandlung von **Ausnahmen:**

Reguläre Werte:

- Standardwerte/Defaults,
- geerbte (generische) Werte

Bei Ausnahme:

- Überschreiben des regulären Wertes

Faktorisierung von Wissen

Die Idee framebasierter Systeme geht davon aus, daß man Wissen innerhalb vorgegebener Strukturen sukzessive erweitern und aktualisieren kann.

Es gibt „generische“ und konkrete Frames, die zwar die gleiche Struktur, aber verschiedene Werte haben:

- was gehört dazu → Schema
- was ist normal → defaults
- wie berechnen → prozedurales Wissen
- allgemeine Eigenschaften → Vererbung.

Inferenzen mit Frames 1

Beantwortung von Fragen: Was ist der Wert von Slot x von Frame y ? Beispiel: Der Slot reisender von r123 hat den Wert GS.

Auswertung von Prozeduren

- if-added: Prozedur eines Slots wird ausgewertet, sobald ein Wert für den Slot eingetragen wird (entspricht datengetriebener Auswertung).
- if-needed: Prozedur wird ausgewertet, sobald auf den Slot zugegriffen wird (entspricht ziel-orientierter Auswertung).
- Beispiel: fläche von q1 ist 25.

Inferenzen mit Frames 2

Vererbung: Falls eine Information nicht im aktuellen Frame gespeichert ist, folge den member-of und superclass Slots, um zu sehen, ob die Information in einer Superklasse gefunden werden kann. Beispiel: Der Slot geschlecht von GS hat den Wert männlich.

Matching: Finde den Frame, der die aktuelle Situation am besten beschreibt. Beispiel: [s23 kinder:jens] passt zu dem Frame vater.

Default Reasoning

Default Reasoning ist das Ziehen plausibler Schlüsse, wenn es keine endgültige Evidenz gibt, daß das Gegenteil gilt

$\rho|\phi \models \phi$ "Wenn man herleiten kann, dass ρ gilt und es konsistent ist anzunehmen, dass ϕ gilt, dann leite ϕ her".

- Beispiel: Tweety ist ein Vogel, Vögel können fliegen. Kann Tweety fliegen?

Default-Regel: $\text{vogel}(x) \mid \text{fliegt}(x) \models \text{fliegt}(x)$

Wissensbasis: $\text{vogel}(\text{tweety})$

Inferenz: $\text{fliegt}(\text{tweety})$

Default Reasoning und Vererbung

Default-Werte (Standard-Werte) sind bei Klassen bzw. generischen Konzepten spezifiziert und können bei spezielleren Konzepten bzw. Instanzen überschrieben werden.

Nicht-monotones Schliessen

Default-Regel: $\text{vogel}(x) \mid \text{fliegt}(x) \models \text{fliegt}(x)$

Wissensbasis: $\text{vogel}(\text{tweety})$

$\forall x: [\text{pinguin}(x) \rightarrow \text{vogel}(x)]$

$\forall x: [\text{pinguin}(x) \rightarrow \neg \text{fliegt}(x)]$

- Da $\text{vogel}(\text{tweety})$ wahr ist und $\text{fliegt}(\text{tweety})$ konsistent ist mit der Wissensbasis, kann man mit der Default-Regel schließen: $\text{fliegt}(\text{tweety})$.

- Die Situation ändert sich dramatisch, wenn die Aussage $\text{pinguin}(\text{tweety})$ zur Wissensbasis hinzugefügt wird. Da nun $\neg \text{fliegt}(\text{tweety})$ hergeleitet werden kann, ist $\text{fliegt}(\text{tweety})$ nicht mehr konsistent mit der Wissensbasis, so daß die Default-Regel nicht mehr anwendbar ist.

Nicht-monotones Schliessen

- Default Reasoning ist nicht-monoton: Durch neue Informationen können Schlüsse falsch werden.
- Nicht-monotones Schließen verwaltet alle Inferenzschritte und überprüft beim Eintreffen neuer Informationen, welche Schlüsse ungültig geworden sind.
- Es gibt verschiedene nicht-monotone Logiken. Für eine konkrete Anwendung muss die jeweils passende gewählt werden.

Beispiele für framebasierte Systeme

Medizinische Diagnose

Die Diagnose zielt auf Ausfüllen der Slots, um eine Diagnose und Therapie zu bekommen. Dazu notwendig ist der Eintrag von Symptomwerten.

Lagerhaltung

Ein beständiger Prozeß der Aktualisierung löst Bestellungen aus etc.

Arbeitsablauf frame-basierter Systeme:

- Auswahl (Vorgabe) des zu bearbeitenden Frames
- Ausfüllen der Slots (Vererbung, Berechnung),
- dabei ggf. weitere Aktionen (Dämonen) aktivieren.

Analogie zu objektorientierter Programmierung

- dort aber mehr die dynamische, prozedurale Sicht der Interaktion von Objekten (messages),
- hier mehr die deklarative, statische Sicht der Wissens-Repräsentation

Analogie von Semantischen Netzen und Frames

Frames können als objektzentrierte Repräsentation semantischer Netze gesehen werden.

Frame-Slots, die Verweise auf andere Frames beinhalten, entsprechen den Kanten eines semantischen Netzes.

Anders als die meisten semantischen Netze erlauben alle realisierten Framesysteme Prozeduren.

Probleme der logischen Fundierung von Frames und Semantischen Netzen

Trotz der Bezeichnung 'Semantische Netze' wird die Semantik der Repräsentationen nicht klar definiert; was die Knoten und Kanten bedeuten, wird meist durch die Bezeichnungen suggeriert.

Inferenzen sind meist rein operational (i.a. durch die Implementierung) spezifiziert, bspw. die Semantik der is-a und instance-of Kanten durch einen Vererbungsmechanismus.

Konstrukte können mehrdeutig sein, z.B. bei multipler Vererbung; die tatsächlichen Werte sind deshalb von der Implementierung abhängig.