

Kapitel 8: Wissensrepräsentation mit Semantischen Netzen und Frames

Quellen:

- Die meisten Folien stammen von Dr. Knut Hinkelmann, FH Solothurn
- Bücher: Gottlob, Rich & Knight

Wissensrepräsentation in XPS

- Ein System heißt wissensbasiert, wenn es Weltwissen deklarativ in Form einer Wissensbasis speichert und verwendet.
- Eine Wissensbasis ist eine Datenstruktur, die als Beschreibung von Zuständen der Welt interpretiert werden kann.
- Deklaratives Wissen beschreibt, wie die Welt ist.
- Vorteile:
 - modular
 - veränderbar / flexibel
 - verstehbar
 - erklärbar
 - verwendungsunabhängig

Programmieren vs. Wissen Repräsentieren

Programmieren

- Entwerfe Algorithmus
- Wähle Programmiersprache
- Kodiere Algorithmus
- Laß Programm laufen

Wissensrepräsentation

- Identifiziere relevantes Wissen
- Wähle Repräsentationssprache
- Repräsentiere das Wissen
- Inferiere Problemlösung

Was wenn:

- inspiziere Programm
- kodiere neu

Fehler auftreten ?

- eliminiere falsche Information
- füge fehlende Information hinzu

Was wenn:

- Programm Trace?

Erklärung verlangt wird ?

- Prämissen der Inferenz

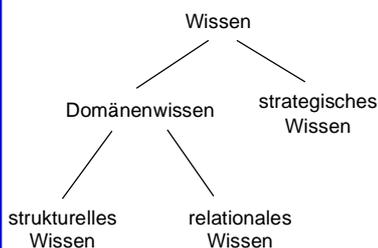
Was wenn:

- kodiere neu

Problem sich ändert ?

- füge neues Wissen hinzu

Wissensarten: Unterscheidung zwischen Domänenwissen und strategischem Wissen



– Domänenwissen: Wissen über das Anwendungsgebiet

- **strukturelles Wissen:** Entitäten des Anwendungsgebiets und strukturelle Beziehungen zwischen ihnen. Die wichtigsten strukturellen Beziehungen sind:

- *Klassifikation* (instance-of): Objekt A ist vom Typ bzw. der Klasse K_A
- *Subsumtion* (is-a): Klasse K_A ist allgemeiner als Klasse K_B
- *Aggregation* (part-of): Objekt A besteht aus den Teilen A_1, A_2, \dots

- **relationales Wissen:** nicht-strukturelle Beziehungen und Eigenschaften

– strategisches Wissen: Wissen darüber, wie man Anwendungswissen einsetzt, um ein Problem zu lösen

Für die verschiedenen Typen von Wissen gibt es unterschiedlich geeignete Repräsentationsformalismen

Beispiel: was weiß ein erfahrener Mechaniker?

Strukturell / relational / strategisch ? Wie modellieren?

- Es gibt Diesel- und Benzinmotoren.
- Wenn sich der Anlasser nicht muckst, schaut man zuerst mal nach ...
- Opel stellt Kadetts, Corsa, Tigras, ... her.
- Den Corsa gibt's in den Varianten
- Eine Einspritzpumpe besteht aus den Teilen
- Beim Manta ist ... eine typische Schwachstelle.
- Vor zwei Jahren, am Porsche vom Nicolin, da war dies und das zu beobachten und erstaunlicherweise war die Ursache, daß ...
- Die Blinker und der Rest der elektrischen Anlage sind so und so verkabelt.
- Der Motor im ... ist eine Variante des ...
-



Andreas Abecker



Mörsbach, Sammersemesler 2000

Wissensrepräsentation durch Logik: Diskussion

- Die Logik ist eine sehr allgemeine Sprache zur Repräsentation von Wissen
- Auf der epistemologischen Ebene stehen nur Prädikate, Konstanten und Funktionen für die Wissensrepräsentation zur Verfügung
- Klassen, strukturelle Beziehungen (is-a, instance-of, part-of) und Eigenschaften müssen durch Prädikate repräsentiert werden (kognitiv adäquat?)
- Die Ausdrucksmächtigkeit der Sprache und die Allgemeinheit der Inferenzen erfordert Einschränkungen der Inferenzstrategien und der Ausdrucksmächtigkeit, um effiziente Verarbeitung zu ermöglichen

Wie können Wissensrepräsentationsformalismen aussehen, die kognitiv adäquater und / oder effizienter verarbeitbar sind?



Andreas Abecker

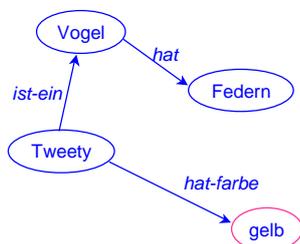


Quelle: Hinkelmann
Mörsbach, Sammersemesler 2000

Semantische Netze und Frames zur Repräsentation von Domänenwissen

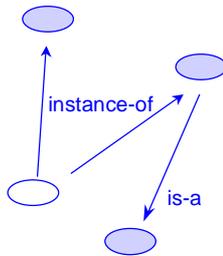
- Die Ideen semantischer Netze und Frames gehen auf zwei Erkenntnisse über das menschliche Gedächtnis zurück:
 - Das menschliche Gedächtnis zeichnet sich u.a. dadurch aus, daß es eine große Anzahl von Verbindungen oder Assoziationen zwischen Informationen gibt.
 - **Semantische Netze**: weisen einen hohen Verbindungsgrad auf
 - Wissensseinheiten sind relativ groß und strukturiert. Wissen sollte deshalb in Form von Konzepten mit assoziierten Beschreibungen organisiert werden.
 - **Frames**: organisieren Wissen in Form von Konzepten
- Beide Formalismen sind eng verwandt und haben sich im Laufe ihrer Entwicklung stark angenähert

Semantische Netze: Graphische Wissensrepräsentation



- Semantische Netze bestehen aus zwei Primitiven:
 - Knoten entsprechen Objekten oder Klassen von Objekten
 - Kanten entsprechen 2-stelligen Relationen zwischen Objekten
- In den Knoten selbst ist keine Information enthalten, alles Wissen über einen Knoten wird durch die Verbindungen repräsentiert, die von ihm ausgehen.

Semantische Netze: Knoten- und Kantenarten

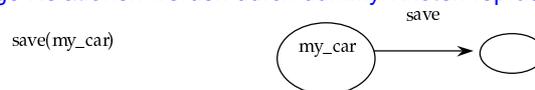


- Auf der epistemologischen Ebene hat sich die Unterscheidung von zwei primitiven Knoten durchgesetzt:
 - generisches Konzept (Klasse)
 - individuelles Konzept (Instanz)
- Daraus ergibt sich die Unterscheidung von mindestens drei Arten von Kanten
 - strukturelle Beziehungen:
 - **is-a**: Relation zwischen generischen Objekten
 - **instance-of**: Relation zwischen individuellem und generischem Objekt
 - nicht-strukturelle Beziehungen
 - beliebige 2-stellige Relationen

Repräsentation n-stelliger Relationen

In semantischen Netzen müssen mehrstellige Relationen auf 2-stellige reduziert werden:

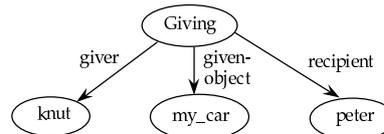
- einstellige Relationen werden durch dummy-Knoten repräsentiert:



- für n-stellige Relationen ($n > 2$) wird ein Knoten eingeführt, der die Relation selbst definiert:

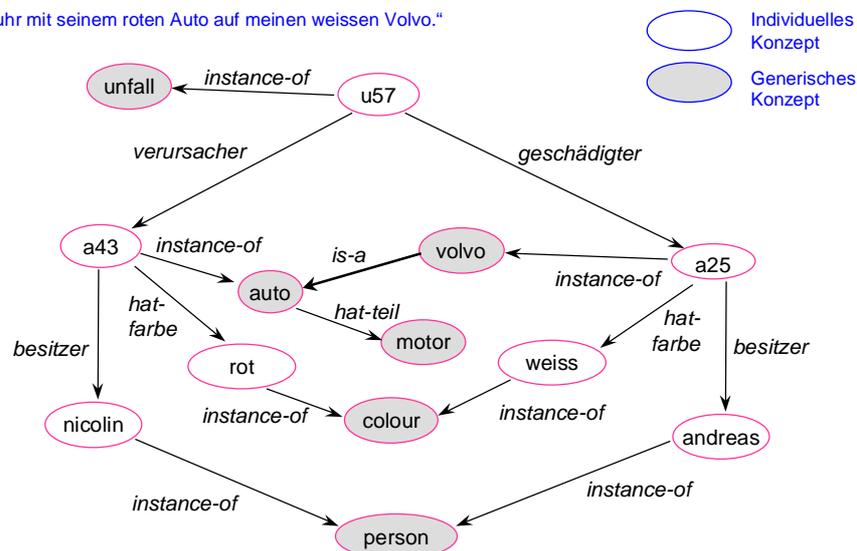
- $p(t_1, \dots, t_n)$ entspricht $p_1(t_1, PP) \wedge \dots \wedge p_n(t_n, PP)$ wobei
 $\forall i \in [1..n] \forall x: p_i(x, PP) \leftrightarrow \exists x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n: p(x_1, x_2, \dots, x_{i-1}, x, x_{i+1}, \dots, x_n)$

give(knut, peter, my_car)



Beispiel für ein Semantisches Netz

„Nicolin fuhr mit seinem roten Auto auf meinen weissen Volvo.“



Andreas Abecker



Nach: Hinkelmann
Moxbach, Sammers emes ler 2000

Allgemeine Inferenzen in Semantischen Netzen

- **Beantwortung von Anfragen** durch Verfolgen von Kanten
 - Wie ist der Wert der Relation R von Knoten A?
 - Beispiel: Wie ist der Wert der Relation hat_farbe von a25?
Antwort: weiss
- **Spreading Activation** bzw. Intersection Search:
 - Gibt es eine Beziehung zwischen Knoten A und Knoten B?
 - Inferenz: Finde einen Knoten N, der sowohl von A als auch von B erreichbar ist.
 - Beispiel: Welche Beziehung haben andreas und weiss?
Antwort: weiss ist die Farbe von a25, dessen besitzer andreas ist
- **Vererbung:** Falls eine Information nicht an einem Knoten des Netzwerks gespeichert ist, folge den instance-of und is-a Pfaden um zu sehen, ob die Information an einer Superklasse gefunden werden kann.
 - Beispiel: Hat mein Auto einen Motor?
Antwort: ja

Andreas Abecker



Quelle: Hinkelmann
Moxbach, Sammers emes ler 2000

Aufgabe: Wie sieht das entsprechende Semantische Netz aus?

„Mary gab ihrer Lieblingscousine ihre grüne Vase mit Blümchenmuster.“



Andreas Abecker



Mörsch, Sommersemester 2000

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Modellierungshinweise: WAS modellieren?

- **Objekte** identifizieren. Dabei möglichst als Instanzen von **Klassen** modellieren. Gegebenenfalls weitergehend Klassenverbände aufbauen. Zusammenhänge zwischen auftretenden Klassen sehen und ggf. mitmodellieren (gemeinsame Oberklassen, Links etc.). Ggf. komplexe Begriffe zerlegen („Blümchenmuster“).
- **Eigenschaften** als Links / Kanten, **Werte** wiederum als Objekte, dazugehörige Klassen nicht vergessen.
- **Relationale Bezüge, komplexe Beziehungen, Aktivitäten** oder **zeitliche Verläufe** modellieren, z.B. durch n-stellige Prädikate, d.h. Objekte mit n Links (Beispiel bei Mary: „gab“ - Wer? Wem? Was? Wie?). Die Objekte wiederum in generische Klassen einordnen (allgemeines „geben“) und ggf. Verbände aufbauen („aktivität“).
- **Adverbiale** nicht vergessen, Bezüge klären, z.B. als ein Link der Aktivität modellieren.
- Ggf. etwas Mühe in Spezialeffekte verwenden („Lieblingscousine“).



Andreas Abecker



Mörsch, Sommersemester 2000

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Modellierungshinweise: WIE modellieren?

Vollständig, umfassend, detailliert

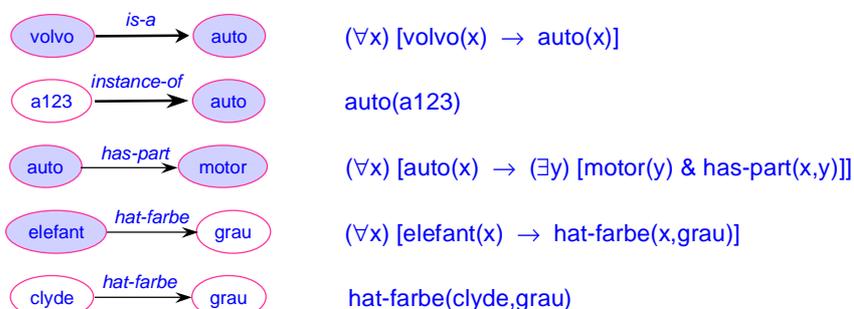
- keine allzu groben Konzepte ohne Verfeinerung („Ding“)
- keine zu speziellen Konzepte ohne Oberklassen („Donaudampfschiffahrtsgesellschafts....“)
- nichts vergessen (siehe vorhergehende Folie)
- auch versteckte Aussagen
- immer mal eine Oberklasse mit modellieren
- immer Objekte und Klassen unterscheiden

Homogen

- gleiche Dinge gleich modellieren
- mehrfach auftauchende Werte und Klassen auch teilen
- Links standardisieren

Semantik Semantischer Netze

- Obwohl als Bezeichner für Knoten und Kanten sinntragende Begriffe verwendet werden, muß festgestellt werden, daß dies nur für den Benutzer von Bedeutung ist. Für das System selbst wird dadurch keine Semantik definiert. Insofern ist der Begriff semantisches Netz irreführend.
- Die Semantik semantischer Netze ergibt sich durch Übersetzung in Prädikatenlogik erster Stufe.
- Beispiele:



Probleme mit Semantischen Netzen

Darstellung komplexer Sachverhalte

- zeitliche Verläufe
- Quantifizierungen
- geschachtelte Strukturen, z.B. in Metaausagen, indirekter Rede, etc.
- Ambiguitäten (Uneindeutigkeiten) der natürlichen Sprache

Diese Dinge kann man weitgehend durch Erweiterungen des Formalismus in den Griff bekommen.

Homogenität als Nutzbarkeitsvoraussetzung

- Nutzung von Netzen nur sinnvoll möglich, wenn unterschiedliche Anwender (Modellierer & Fragesteller) ähnliche Konzeptualisierungen wählen (Begriffe, Klassen, Unterklassen, Linktypen)
 - Modellierungsrichtlinien, Bibliotheken, standardisierte Ontologien als Abhilfe
- Dies ist ein grundsätzlicheres Problem des Ansatzes.**

Motivation für Frames

- 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
- Inferenz in einem Frame ist zum großen Teil ein Erkennungsprozeß

Frames: Objektbasierte Wissensrepräsentation

Bezeichner		
Slot ₁	Slotbeschreibung ₁	Slotwert ₁
Slot ₂	Slotbeschreibung ₂	Slotwert ₂
Slot ₃	Slotbeschreibung ₃	Slotwert ₃
usw...		

Beispiel:

Knut		
Name	Typ: string, minKardinalität: 1	Hinkelmann
Vomame	Typ: String, minKardinalität: 1	Knut
Alter	Typ: integer, Kardinalität: 1	38

- Frames fassen alle Eigenschaften eines Objekts in einer Datenstruktur zusammen
- Frames bestehen aus:
 - Bezeichner des Frame
 - Slots mit
 - Slotbeschreibungen
 - Slotwerten
- Slotwerte können sein:
 - primitive Werte, z.B. Strings, Integer, ...
 - Verweise auf andere Frames
 - Prozeduren (procedural attachment)
- Slots können durch Slotbeschreibungen (Facetten) eingeschränkt werden, z.B.
 - Typ des Slotwertes
 - Mindest- oder Maximalzahl von Slotwerten
 - Hinweis auf Vererbung

Syntax: Allgemeine Primitive in Framesystemen

- Analog zu Semantischen Netzen hat sich bei Framesystemen auf der epistemologischen Ebene die Unterscheidung von zwei Arten von Frames durchgesetzt:
 - Klasse (entspricht generischem Konzept)
 - Instanz (entspricht individuellem Konzept)
- Daraus ergibt sich die Unterscheidung von mindestens drei Arten von Slots
 - strukturelle Beziehungen:
 - **superclass**: Relation zwischen Klassen (entspricht **is-a**)
 - **member-of**: Relation zwischen Instanz und Klasse (entspricht **instance-of**)
 - nicht-strukturelle Beziehungen
 - beliebige Slots für Attribute von Objekten

Beispiele für Frames (1)

```
[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 > 60
      then 25
    else 0
    exec: if-needed
]
[KH
  member-of: Vater
  vorname: „Knut“
  alter: 38
]
[r123
  member-of: Reise
  reisender: KH
]
```



Andreas Abecker



Quelle: Hinkelmann
Mörsch, Sammers emes ler 2000

Beispiele für Frames (2)

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

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

[q1
  superclass: quadrat
  seitel: 5
]
```



Andreas Abecker



Quelle: Hinkelmann
Mörsch, Sammers emes ler 2000

Inferenzen mit Frames

- **Beantwortung von Fragen:** Was ist der Wert von Slot A von Frame O?
 - Beispiel: Der Slot `reisender` von `r123` hat den Wert `KH`
- **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
- **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 `KH` hat den Wert `männlich`
- **Matching:** Finde den Frame, der die aktuelle Situation am besten beschreibt
 - Beispiel: `[s23 kinder:jens]` paßt zu dem Frame `vater`



Andreas Abecker



Quelle: Hinkelmann
Mörsbach, Sammers emes ler 2000

Default Reasoning

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

$$p | \phi \quad | - \quad \phi$$

“Wenn man herleiten kann, daß p gilt und es ist konsistent anzunehmen, daß ϕ gilt, dann leite ϕ her”.

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

Default-Regel: `vogel(x) | fliegt(x)`

`| - fliegt(x)`

Wissensbasis: `vogel(tweety)`

Inferenz: `fliegt(tweety)`

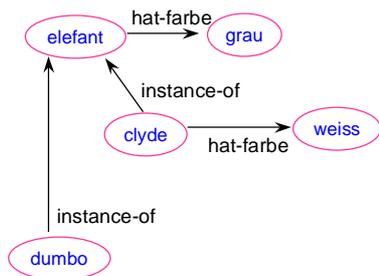


Andreas Abecker



Quelle: Hinkelmann
Mörsbach, Sammers emes ler 2000

Vererbung und Default Reasoning



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

Nicht-monotones Schließen

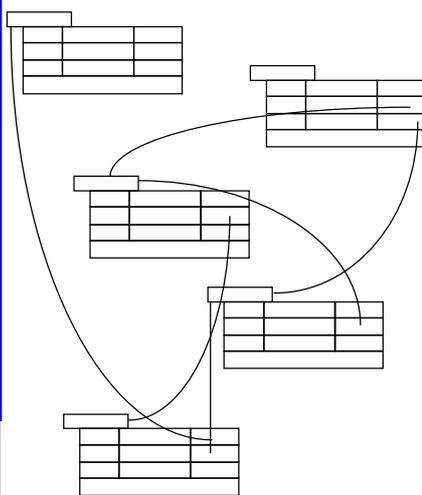
Default-Regel: $\text{vogel}(x) \mid \text{fliegt}(x) \vdash \text{fliegt}(x)$
 Wissensbasis: $\text{vogel}(\text{tweety})$
 $(\forall x) [\text{pinguin}(x) \rightarrow \text{vogel}(x)]$
 $(\forall x) [\text{pinguin}(x) \rightarrow \neg \text{fliegt}(x)]$

- 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.
- ➔ 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.

Aufgabe: wie sieht eine Framestruktur zur Modellierung der BA-Welt aus?

- Begriffe, die nicht vergessen werden sollten: Hiwi, Lehrbeauftragter, XPS-Vorlesung-im-SS2000
- Wo passen if-added, if-needed, if-removed Dämonen?

Analogie von Semantischen Netzen und Framesystemen



- Frames können als objektzentrierte Repräsentation semantischer Netze gesehen werden
- Indem Slotwerte von Frames wiederum Verweise auf andere Frames sind, erhält man ein Netzwerk:
 - Slots entsprechen den Kanten in semantischen Netzen
- Anders als semantische Netze erlauben alle realisierten Framesysteme ein procedural attachment

Probleme der logischen Fundierung von Semantischen Netzen und Framesystemen

- Semantik: 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:
 - Die Semantik der is-a und instance-of Kanten ergibt sich operational durch den Vererbungsmechanismus, ohne formales Kriterium:
 - Eine neue Klasse wird definiert durch eine virtuelle Kopie ihrer Superklasse, bei der Relationen hinzugefügt, Werte lokal geändert und Defaultwerte überschrieben werden.
 - Es hängt nur von dem Ersteller der Wissensbasis ab, an welcher Stelle eine Klasse in der Hierarchie eingefügt wird.
 - Konstrukte können mehrdeutig sein, z.B. bei multipler Vererbung; die tatsächlichen Werte sind deshalb von der Implementierung abhängig.

Ausblick: Terminologische Systeme (Beschreibungslogiken) bieten eine logische Fundierung von Semantischen Netzen bzw. Frames.



Andreas Abecker



Quelle: Hinkelmann

Mörsch, Sammers 2000