Seminar Ontology-Management

Repräsentationssprachen für Ontologien

Gliederung

- Einführung
- Common Logic
- IDEF5
- Einordnung der Sprachen

Einführung

Einführung

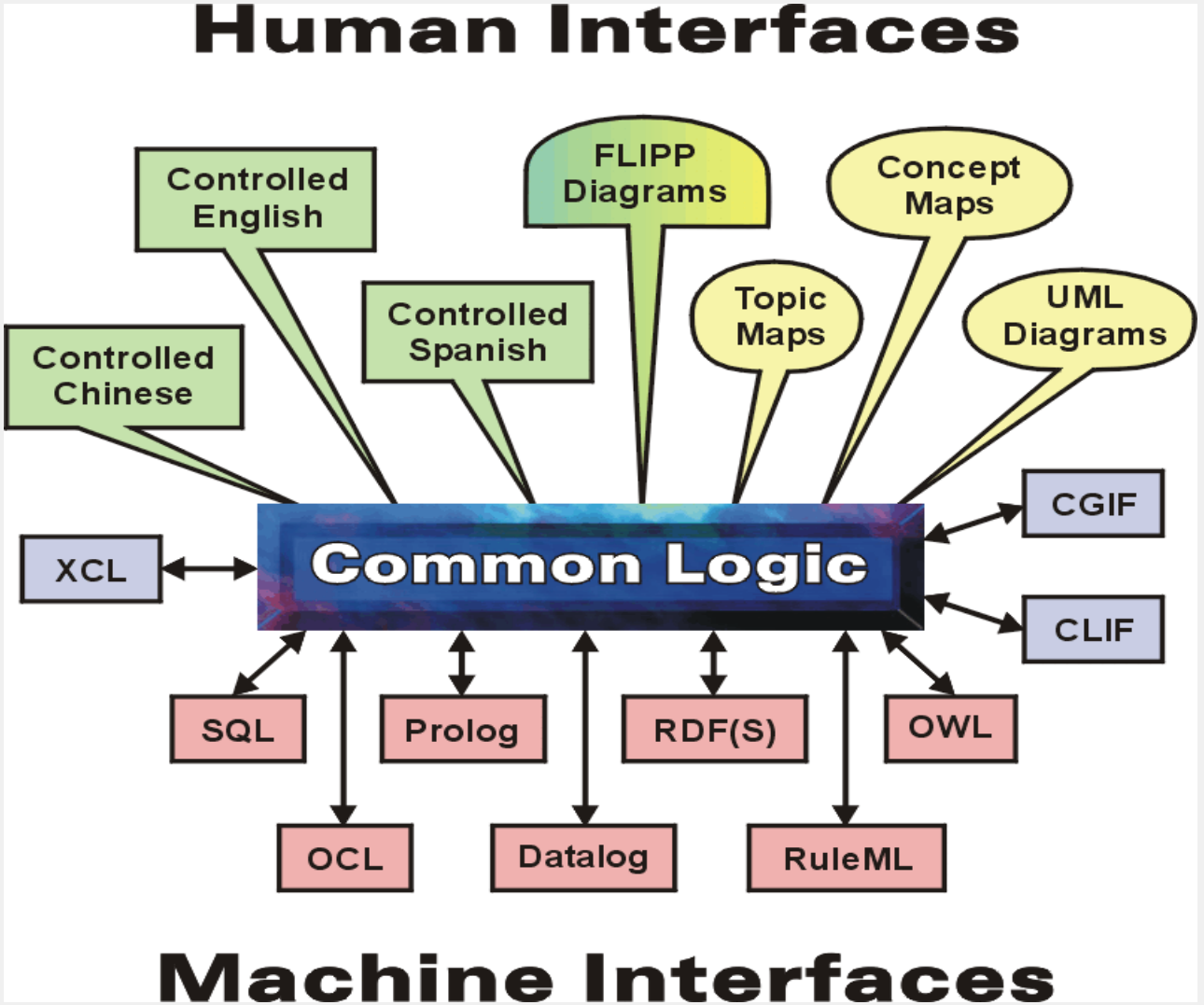
- Wozu Repräsentationssprachen?
  - standartisierte Darstellung von Ontologien
  - Maschinenlesbarkeit
- erlauben eindeutige, formale Beschreibung von Ontologien
- Anforderungen:
  - formale Semantik
  - hinreichende Ausdrucksstärke

Common Logic

Common Logic (CL)

- beschrieben im ISO Standart ISO/IEC 24797:2007
- keine formale Sprache, sondern ein Framework für eine Familie von formalen Sprachen
- beschreibt abstrakten Syntax und Semantik
- eine Sprache dieser Familie heißt CL Dialekt
- eine Sprache ist ein Dialekt, wenn es ein Mapping gibt, welche den Syntax auf den abstrakten CL Syntax abbildet; Semantik folgt daraus
- Dialekte sind i.A. in einander übersetzbar

Common Logic (CL)



- Quelle: http://www.jfsowa.com/talks/cl_sowa.pdf

Common Logic (CL)

- Common Logic basiert auf der Prädikatenlogik erster Stufe
- mit Sequenzsymbolen kann jedoch mehr als mit Prädikatenlogik erster Stufe ausgesagt werden
- ISO Standart beschreibt 3 Syntaxen:
  - Common Logic Interchange Format (CLIF)
  - Conceptual Graph Interchange Format (CGIF)
  - eXtended Common Logic Markup Language (XCL)
- jeder Dialekt muss Mapping in eine dieser haben

Common Logic Syntax

- der Syntax lässt sich in Kategorien aufteilen
- Text
  - eine Menge von Ausdrücken und einem Namen
- Ausdruck
  - ein Modul, Satz oder Einbindung
- Kommentar:
  - Daten, welche an Ausdrücke oder andere Kommentare angehängt werden können
- Satz
  - ist ein boolscher oder quantifizierter Satz, oder ein Satz mit einem Kommentar

Common Logic Syntax

- Modul
  - besteht aus einem Namen, einer ausschließenden Menge und einem Text
- Einbindung
  - besteht aus einem Namen, der ein externes Stück CL Sprache identifiziert
- Quantifizierter Satz
  - hat einen Quantor, eine bindende Sequenz aus Namen und Sequenzsymbolen und einen Satz
  - jeder Dialekt sollte All- und Existenzquantor definieren

Common Logic Syntax

- Boolscher Satz
  - besteht aus Typ und einer Menge von Sätzen, deren Anzahl durch den Typ bestimmt ist
  - jeder Dialekt sollte mindesten die 5 Typen definieren: Konjunktion, Disjunktion, Implikation, Bidirektion, Negation
- Atom
  - ist entweder eine Gleichung mit zwei Termen als Argumenten
  - oder ein atomarer Satz mit einem Term, dem Prädikat und einer Termsequenz, den Argumenten

Common Logic Syntax

- Term
  - ist ein Name, Funktionsterm oder ein Term mit einem Kommentar
- Funktionsterm
  - besteht aus einem Term, dem Operator und einer Sequenz von Termen, den Argumenten
- Sequenz von Termen
  - ist eine endliche Sequenz von Termen oder Sequenzsymbolen

Common Logic Syntax

- Vokabular
  - eine Menge von Namen und Sequenzsymbolen
- Namen und Sequenzsymbole
  - zwei zu einander und zu allen anderen Kategorien disjunkte Mengen
- Namen und Sequenzsymbole unterliegen keinen anderen Bestimmungen
- manche Dialekte unterteilen Namen in Diskurs- und nicht-Diskursnamen

Common Logic Semantik

- Semantik
  - CL Text erfüllt Interpretationen
- Diskursuniversum (UD)
  - Menge aller Individuen einer Interpretation
- Referenzuniversum (UR)
  - Menge aller Dinge die benötigt werden, um die Bedeutung logischer Ausdrücke in einer Interpretation zu bestimmen

- Interpretation
  - Eine Interpretation I über dem Vokabular V ist die Menge UR_I mit der Teilmenge UD_I und 4 Abbildungen
  - rel_I: von UR_I auf Teilmengen von $UD_I^x = \langle \langle x_1, \ldots, x_n \rangle | x_1, \ldots, x_n \in UD_I \rangle$
  - fun_I: von UR_I in die Menge der Funktionen, die UD_I^x *nach* UD_I abbilden
  - int_I: von Namen in V nach UR_I, wobei $int_I(v)$ in UD_I, gdw. v ein Diskursname ist
  - seq_I: Sequenzsymbole in V nach UD_I^x

Common Logic Semantik

- zur Bestimmung des Wertes eines Ausdrucks A in einer Interpretation I gibt es 20 Regeln

	Hat E die Form	So ist I(E)
1	Name N	intI(N)
3	Sequenz von Termen T_1 ... T_n	I(T_1);I(<T_2 ... T_n>)
5	Term mit Operator O und Argumentensequenz S	funI(I(O))(I(S))
7	atomarer Satz mit Prädikat P und Argumentensequenz S	wahr, falls I(S) in relI(I(P)), ansonsten falsch
9	boolscher Satz vom Typ Konjunktion und Komponenten C1 ... Cn	wahr, falls I(C1) = ... = I(Cn) = wahr, ansonsten falsch

- 1 + 2
- x < y
- x = 1 ^ y = 2

Common Logic Übersetzen von Dialekten

- Übersetzung
  - Mapping von Ausdrücken aus Quelldialekt auf Ausdrücke im Zieldialekt
- einfach, wenn beide Dialekte nicht-Diskursnamen enthalten, oder beide nicht
- Nicht-Diskursnamen -> Diskursnamen
  - Modul mit nicht-Diskursnamen in ausschließender Menge
- Diskursname -> nicht-Diskursnamen
  - in atomaren Sätzen werden Prädikate und Operatoren durch 'Platzhalter' ersetzt

Common Logic CLIF

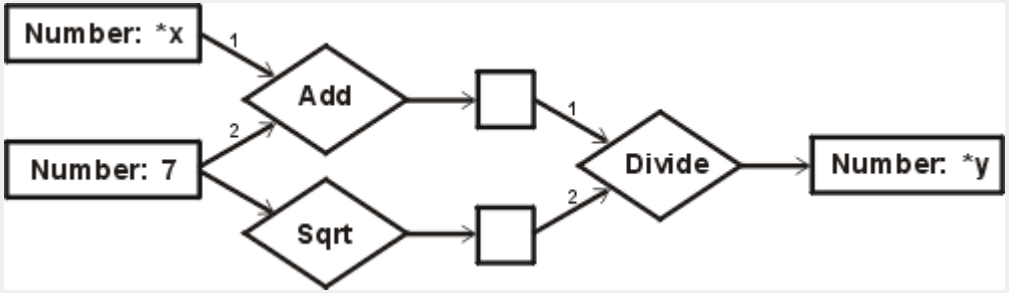
- Ähnlich KIF
- nutzt geklammerte Präfixschreibweise (bekannt aus LISP)
- Y = (x+7) / sqrt(7)
- (exists ((x Number) (y Number))
  (= y (Divide (Add x 7) (Sqrt 7))))

Common Logic CGIF

- Konzeptioneller Graph: Repräsentationsform für Logik in Form bipartiter Graphen
- [ ] beinhalten Konzepte, ( ) beinhalten Relationen
- Konzepten kann durch ': Wert' ein Wert zugewiesen werden, ': *x' beschreibt neuen Bezeichner
- in Relationen referenziert '?x' Bezeichner x
- Präfix '@every' ist der Allquantor, kein Präfix impliziert Existensquantor

Common Logic CGIF

- Y = (x+7) / sqrt(7)
- [Number: *x] [Number: *y]
  (Add ?x 7 | *u) (Sqrt 7 | *v) (Divide ?u ?v | ?y)



Quelle:

http://common-logic.org/SemTech2008/cl_sowa.pdf

Common Logic XCL

- für Übertragung von Common Logic Text gedachter Dialekt in XML-Form
- Beispiel: Es gibt Studenten über 30

```
<?xml version="1.0"?>
<text xmlns="http://purl.org/xcl/1.0/">
 <exists>
  <var name="alterStudent" />
  <and>
   <atomic>
    <relation>über_30</relation>
    <term name="alterStudent" />
   </atomic>
   <atomic>
    <relation>Student</relation>
    <term name="alterStudent" />
   </atomic>
  </and>
 </exists>
</text>
```

IDEF5

IDEF5

- IDEF5 beschreibt Methodik zum Erstellen von Ontologien sowie 2 Sprachen
- beinhaltet 3 Konzepte
  - Systematische Vorgehensweise und Richtlinien für die Ontologieerstellung
  - IDEF5 schematic Language
    - graphische Sprache
    - kann gebräuchlichste Formen von Informationen einfach darstellen
  - IDEF5 elaboration Language
    - Textsprache
    - sehr ausdrucksstark

IDEF5

- In IDEF5 sind folgende Kategorien auszumachen
- Art
  - Kategorie, welcher Individuen angehören können
  - Definierende Eigenschaften beschreiben die Art
  - Mitglieder müssen eine Eigenschaft erfüllen
- Individuum
  - Repräsentieren meist Objekte der realen Welt
  - nicht mehrfach instanziierbar

- Eigenschafen und Attribute
  - Attribut
    - eine Art Funktion, welche einem Objekt einen speziellen Wert zuordnet
  - Eigenschaft
    - Eigenschaft des Objektes
  - oft Korrelation: Attribut Größe von 1,80 m, Eigenschaft 1,80 m groß
  - aber: Eigenschaft 'hat Arme' nicht auf Attribut zurückzuführen

- Relationen
  - Verbindungen zwischen Objekten
  - mehrfach Instanziierbar
- Prozess
  - etwas, das in einen bestimmten Zeitraum passiert
  - führt bei beteiligten Individuen meist zu Änderung das Zustandes oder der Art
  - mehrfach inszanziierbar
- Zustand
  - Eigenschaft zu bestimmtem Zeitpunkt

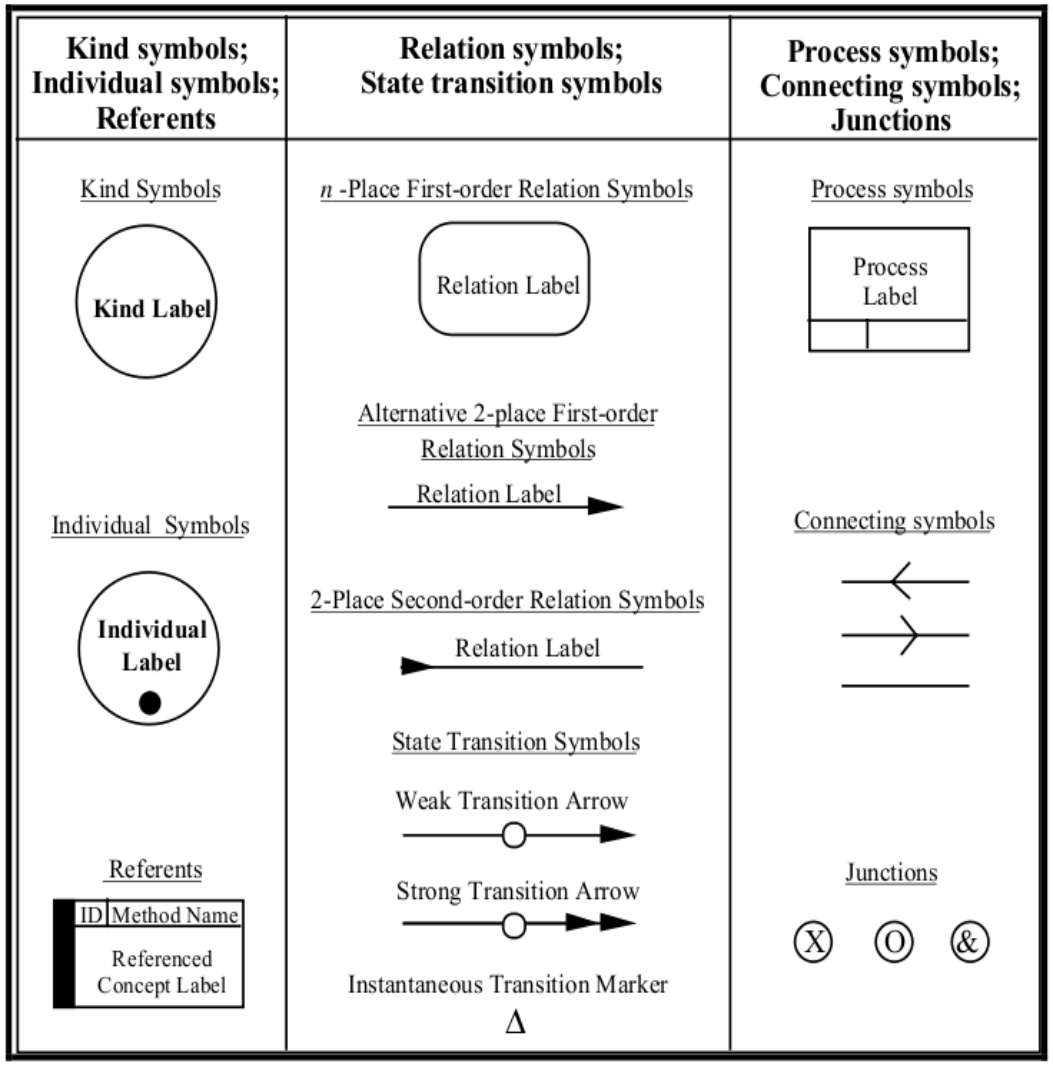
IDEF5 Entwicklungsprozess

- Es werden 5 Aktivitäten beschrieben
- Organisation und Aufgabenerfassung
  - Projekt definieren (Rahmen, Zweck, Standpunkt, ...)
  - Verteilung der Rollen
    - Projektleiter
    - Analyst
    - Experte
    - Teammitglied
    - Gutachter
- Datensammeln
  - Daten sammeln durch Befragung und aus Dokumenten

IDEF5 Entwicklungsprozess

- Daten analysieren
  - Identifizierung von Objekten
- vorläufige Ontologie erstellen
  - Prototypen für Arten, Eigenschaften usw. erstellen
- Ontologie präzisieren und validieren
  - Verfeinern und Präzisieren der Resultate vorangegangenen Phase
  - führt zur finalen Ontologie

IDEF5 Schematic Language



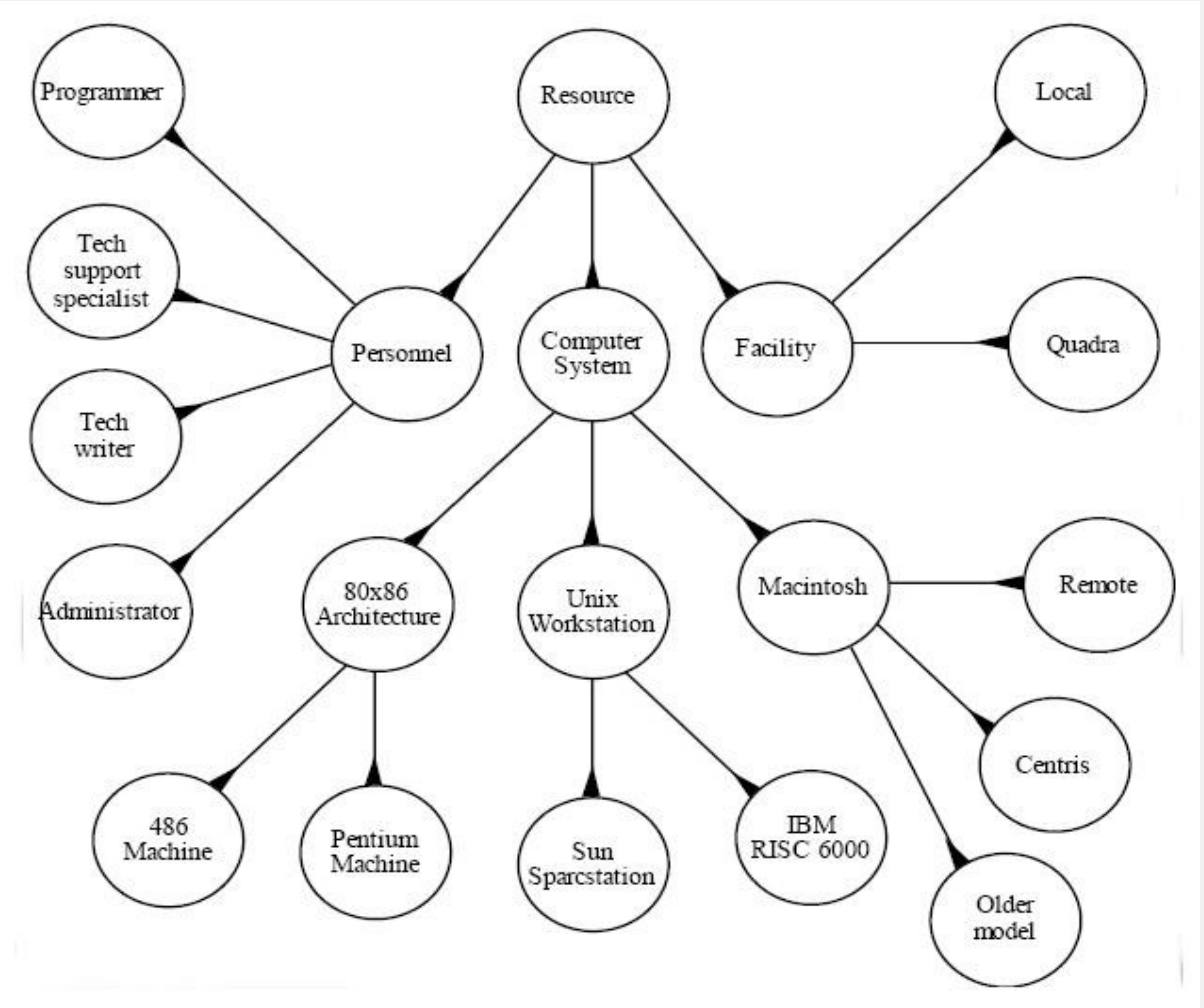
- Quelle: http://www.idef.com/pdf/Idef5.pdf

IDEF5 Schematic Language

- ausdrucksschwächer, einfache Visualisierung
- vier Typen (Schemas) von Diagrammen
  - Klassifikation
    - Ordnung von Wissen in logische Kategorien
  - Komposition
    - Visualisierung von “Teil von” - Beziehungen
  - Relation
    - Visualisierung von Relationen ersten und zweiten Grades
  - Objektzustand
    - Beschreiben von Zustandsänderungen durch Prozesse

IDEF5 Schematic Language - Klassifikationsschema

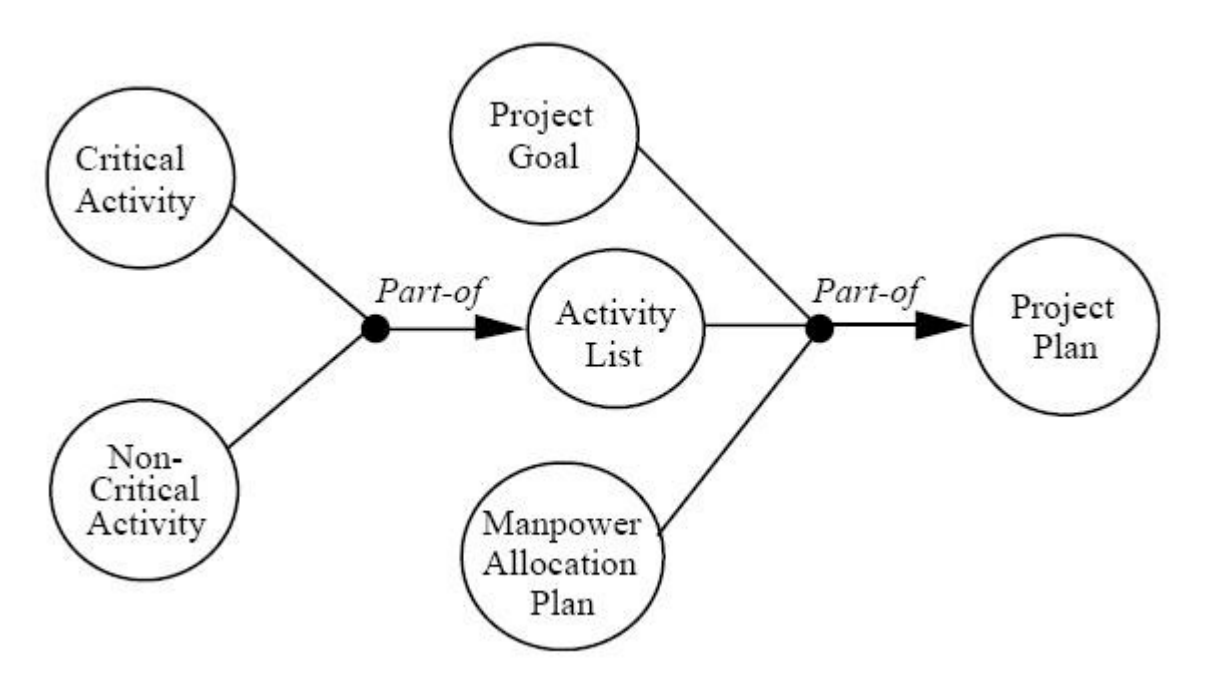
- is-a Relationen



Quelle: http://en.wikipedia.org/wiki/IDEF5

IDEF5 Schematic Language - Kompositionsschema

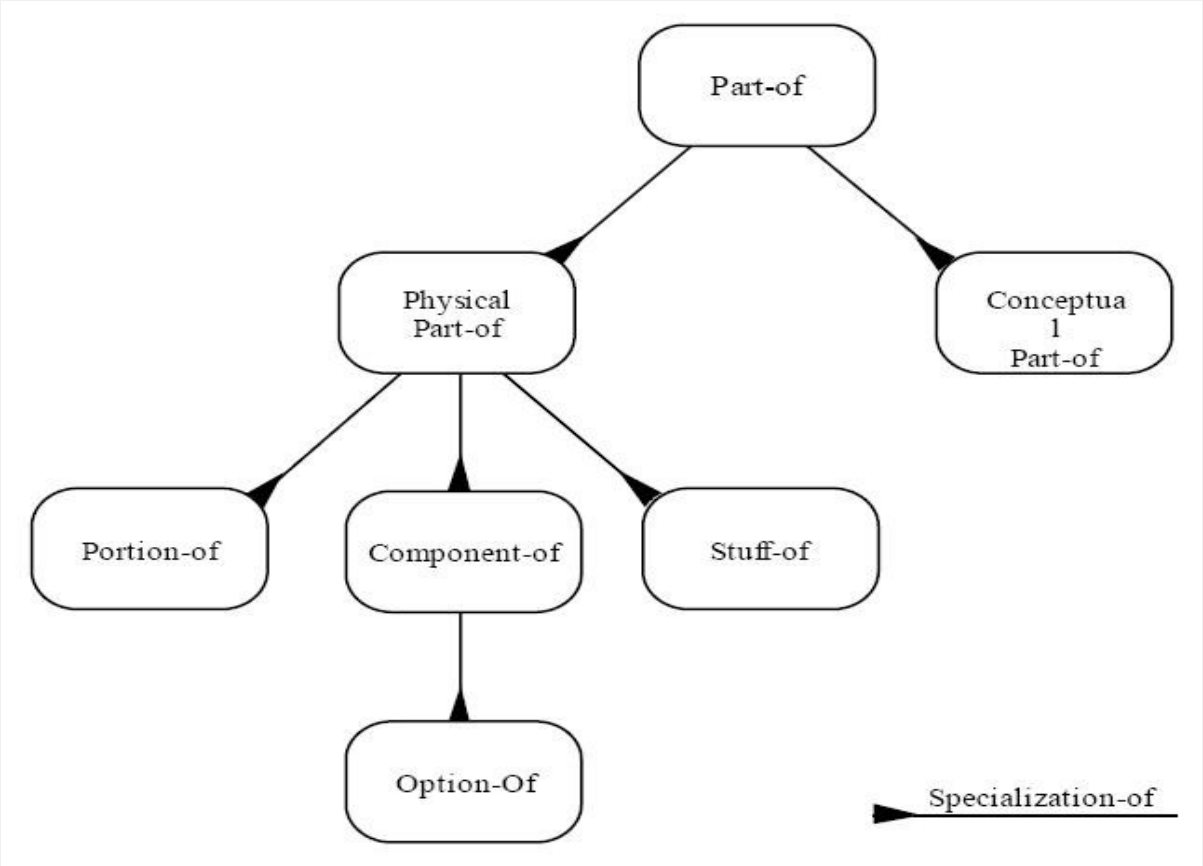
- part-of und Assotiationen sehr oft in Ontologien genutzt, welche sich auf Produktion u.ä. Beziehen
- X part-of Y, kann Teil sein, muss aber nicht



Quelle: http://en.wikipedia.org/wiki/IDEF5

IDEF5 Schematic Language - Relationsschema

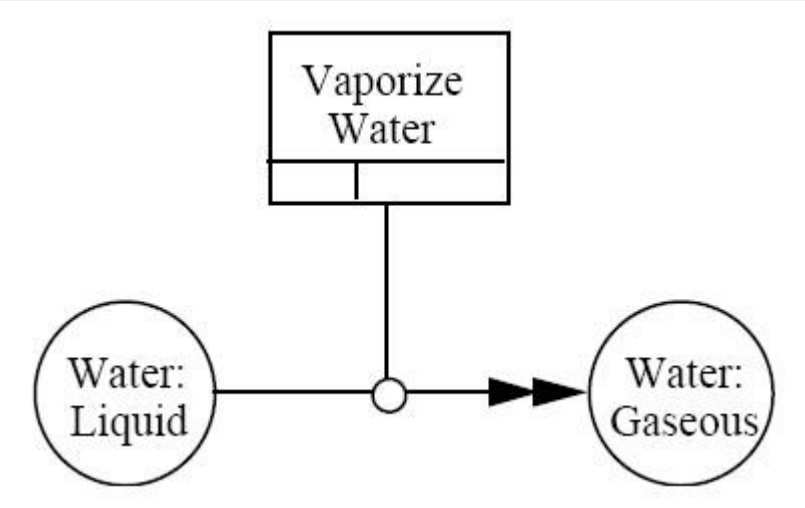
- beschreibt Relationen



Quelle: Quelle: http://en.wikipedia.org/wiki/IDEF5

IDEF5 Schematic Language - Objektzustandsschema

- Art:Zustand, z.B. Wasser:gefrohren
- transition link signalisiert zulässige Zustandsänderung, damit verknüpfter Prozess beschreibt diese



Quelle: http://en.wikipedia.org/wiki/IDEF5

IDEF5 Elaboration Language

- sehr ausdrucksstark, alle für Ontologie-repräsentation notwendigen Ausrücke enthalten
- nutzt geklammerte Präfixnotation

IDEF5 Elaboration Language

- Konstanten
  - Wörter, die i. A. Objekte der realen Welt bezeichnen
  - Ontologiekonstanten
  - Individuenkonstanten
  - Artkonstanten
  - Relationskonstanten
  - Funktionskonstanten
  - Attributskonstanten
  - u.v.m.

IDEF5 Elaboration Language

- Variablen
  - sind Platzhalter für Konstanten
  - ?Wort Individuenvariable
  - #Wort Prädikatvariable
- Operatoren
  - Definitionsoperatoren, z.B. Define-function
  - Termoperatoren, z.B. Listof, setof, if
  - Satzoperatoren
    - Logische Operatoren: =, forall, exists, ...
    - Modale Operatoren: nec, pos
    - weitere

IDEF5 Elaboration Language

- Terme
  - Konstanten und Variablen sind Terme
  - Attributterme, z.B. (age-of Hans)
  - Funktionsterme, z.B. (sqrt 2)
  - Listenterme, z.B. (listof x y z)
  - Mengenterme, z.B. (setof Hans Max Claus)
  - logische Terme, z.B. (if (> (Temperatur ?x) 100) heiß)
  - quantifizierte Terme, z.B. (exists (?x) (student ?x))

IDEF5 Elaboration Language

- Definitionen
  - definieren Individuen-, Relations- und Funktionskonstanten
  - komplette (eine) und partielle (mehrere) Definitionen
  - Individuum
    - komplett: Name und Term, partiell: name
    - optional beschreibender Satz
  - Relation
    - komplett: Name, Liste von Variablen (Argumenten), Term
      Partiell: Name, Liste von Variablen
    - optional: Argumenttypen, beschreibender Satz

IDEF5 Elaboration Language

- Funktion
  - komplett: Name, Liste von Variablen, Satz
  - partiell: Name, Liste von Variablen
  - optional: Argumenttypen; beschreibender Satz

	komplett	partiell
Individuum	(define-individual Koordinatenursprung := listof(0 0))	(define-individual Sonne (rotiert-um Erde Sonne))
Funktion	(define-function quadrat(?x) :argument-type((integer)) := (* (?x ?x)))	(define-function dauer(?x) :argument-type((Zeitintervall)) (sekunden(dauer ?x)))
Relation	(define-relation über(?x ?y) := unter(?y ?x))	(define-relation tochter-von(?x ?y) :argument-type((Frau) (Elternteil)) (=> (tochter-von ?x ?y) (kind-von ?x ?y))

IDEF5 Elaboration Language

- Satz
  - drücken Fakten über Konstanten der Ontologie aus
  - logische Konstanten (Konstanten mit Wahrheitswert)
  - (Un)Gleichheit, = bzw. /= und 2 Terme
  - relationale Sätze
  - logische Sätze
  - quantifizierte Sätze
  - IDEF5 spezifische Sätze -> nächsten Folien

IDEF5 Elaboration Language

```
Gleichung              : (= (age-of Hans) 25)
Ungleicheit            : (/= (preis Äpfel) (preis Birnen)
relationale Satz       : (verheiratet Claus Claudia)
logische Satz          : (and (= (age-of Hans) 25) (student Hans))
quantifizierter Satz : (forall (?x)
                           (=> (and ((mensch ?x) (< (alter ?x) 18))
                                    (kind ?x)))
```

IDEF5 Elaboration Language

- Ontologiekonstrukte
  - drücken Informationen über Ontologie aus
  - 8 Konstrukte, z.B.
  - I5-ontology deklariert Ontologie
  - I5-ontology-context
  - I5-in-purpose
  - I5-ontology-analyst ...

```
(I5-ontology Computerontologie)
(I5-ontology-context Computerontologie “Die Ontologie befasst sich mit der
Beschreibung vom Computern und deren Teilen”)
(I5-ontology-purpose Computerontologie “Die Ontologie soll Personen,
welche sich kaum mit Computern beschäftig haben dieses Themengebiet näher
bringen”)
```

IDEF5 Elaboration Language

- Art Konstrukte
  - beschreibt eine Art
  - I5-kind
  - I5-kind-property
  - I5-kind-attribute
  - I5-kind-description
  - I5-referenced-relations
  - I5-subkind-of
  - I5-object-state
  - I5-process

IDEF5 Elaboration Language

```
(I5-kind Computer)
(I5-kind-property Computer hat_CPU defining)
(I5-kind-attribute Computer Speicherkapazität)
(I5-referenced-relations Computer (schneller_als))
(I5-subkind-of Computer Maschine)
```

IDEF5 Elaboration Language

- Eigenschaft Konstrukte
  - I5-property
  - I5-property-description
  - I5-has-property

```
(I5-property ist_Student)
(I5-has-property Hans ist_Student)
```

IDEF5 Elaboration Language

- Individuen Konstrukte
  - I5-Individual
  - I5-individual-description
  - I5-is-kind-of

```
(I5-individual Hans)
(I5-is-kind-of Hans Mensch)
```

- Attribut Konstrukte
  - I5-attribute
  - I5-attribute-description
  - I5-attribute-applies-to

```
(I5-attribute Vornamen (listof string))
(I5-attribute-applies-to Vornamen Hans)
```

IDEF5 Elaboration Language

- Relationen Konstrukte, deklarieren Relationen
  - I5-relation
  - I5-relation-arity
  - I5-rel-arg-type
  - I5-relation-description

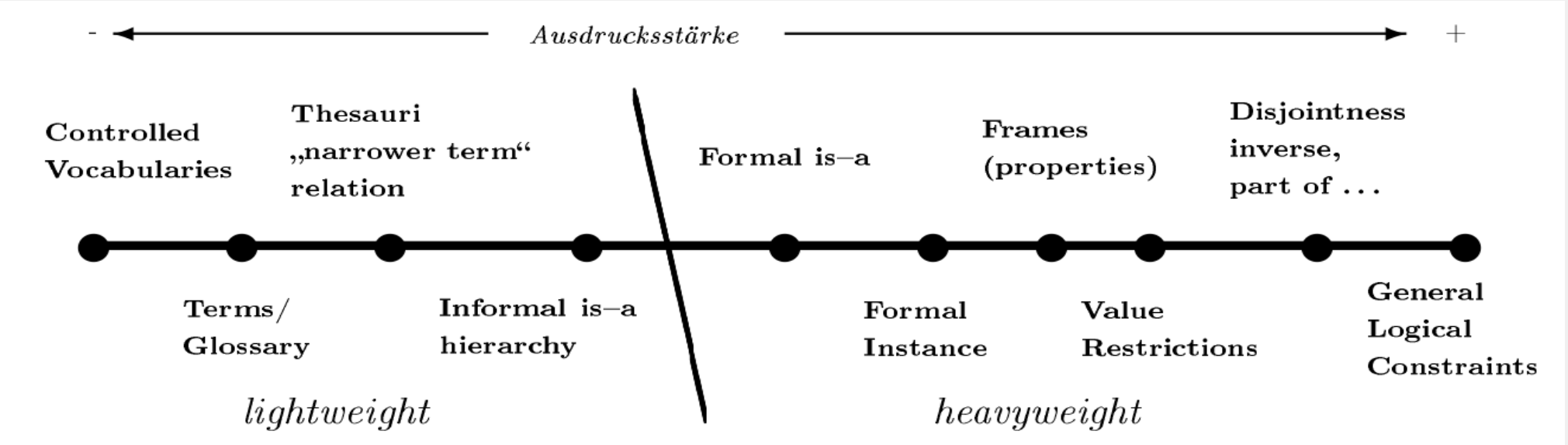
```
(I5-relation part-of)
(I5-relation-arity part-of 2)
(I5-rel-arg-type part-of ((Schraube Kabel)(Computer Auto))
```

IDEF5 Elaboration Language

- Funktionen Konstrukte, deklarieren Funktionen
  - I5-function
  - I5-function-arity
  - I5-fct-arg-type
  - I5-function-description

```
(I5-function Haltbarkeitsdatum)
(I5-function-arity Haltbarkeitsdatum 1)
(I5-function-description Haltbarkeitsdatum “Gibt
nach Eingabe des Herstellungsdatums des Produktes
das Haltbarkeitsdatum zurück”)
(I5-fyo-arg-type (date date))
```

Einordnung der Sprachen



- IDEF5 Elaboration Language ist ganz recht einzuordnen
- IDEF5 Schematic Language zumindest bei “Formal is-a”
- vollständige konforme CL Dialekte sind ganz rechts anzuordnen
- partiell konforme CL Dialekte je nach Grad der Konformität