

## Kapitel 2

# Ontologiebasiertes Wissensmanagement

- 2.1 Anforderungen des Wissensmanagements
- 2.2 Ontologien und verwandte (Modellierungs-) Ansätze
- 2.3 Ontologien
- 2.4 Zusammenfassung

## 2.1 Anforderungen des Wissensmanagements

### „Gemeinsame Sprache“

*"People can't share knowledge if they don't speak a common language"*

[T. Davenport]

- ➔ „Gemeinsame Sprache“ notwendig für funktionierendes Wissensmanagement
  - wohldefiniertes Vokabular an **Lexemen** (lexical entries)
  - einheitliches **Verständnis** welche **Begriffe** (concepts) und **Beziehungen** (relations) durch die Lexeme referenziert werden

### Beispiel für „gemeinsame Sprache“

- Lexeme:
  - "employee", "Angestellter", "Angestellte"
  - "company", "enterprise", "Firma", "Fabrik", "Unternehmen"
  - "member", "membership", "participate", "work"
- Begriffe:
  - Person, Employee, Manager, Consultant, Project
  - Company, Manufacturer, FinanceComp, Insurer, Bank
- Beziehungen:
  - memberOf, participantOf

### Vorteile der „gemeinsamen Sprache“

- ermöglicht Zugriff auf Wissen
  - integriert und vereinheitlicht
    - verschiedene Quellen
    - unterschiedliche Repräsentationen
    - verschiedene Granularitätsstufen
- bietet verschiedene Sichten auf Wissen
  - unter Berücksichtigung von
    - Benutzer
    - Benutzungskontext
  - Fokus auf relevante Aspekte
  - Abstraktionsniveau adäquat
  - Spezielles (aufgaben-spezifisches) Vokabular

## 2.2 Ontologien und verwandte (Modellierungs-) Ansätze

- Beschreibung einer Domäne kann auf verschiedenen Ebenen und Komplexitätsstufen angegangen werden:
  - Glossare
  - Thesauri
  - Datenbankschemata
  - Objektmodelle
  - Ontologien

### 2.2.1 Glossar

- zentrale Sammlung aller Begriffe
- jeder Begriff wird **natürlich-sprachlich** erläutert
- Begriffe stehen **unabhängig** nebeneinander, d.h. sie werden nicht in eine hierarchische Struktur eingebettet (z.B. *is-a*-Hierarchie)
- in den Begriffsbeschreibungen kann es Verweise auf weitere Einträge des Glossars geben

### Beispiel eines Glossars - UML

(Ausschnitt aus dem Glossar von UML - Unified Modeling Language, <http://www.omg.org/technology/uml/index.htm>)

#### aggregation

A special form of association that specifies a whole-part relationship between the aggregate (whole) and a component part.

Contrast: [composition](#).

#### association

A relationship that describes a set of links.

#### attribute

A named property of a type.

#### class

A description of a set of objects that share the same attributes, operations, methods, relationships, and semantics. A class is an implementation of type.

Synonym: [class \[OMA\]](#). See: [type](#), [implementation](#).

#### client

A type, class, or component that requests a service from another type, class, or component.

Synonym: [client object \[OMA\]](#). Contrast: [supplier](#).

### 2.2.2 Thesaurus

- Wörterbuch mit **semantischen Relationen** zwischen den Wortbedeutungen
  - wir betrachten nachfolgend **WordNet** [Miller 95]
  - Vokabular einer Sprache ist gegeben durch
    - $W = \{ (f, s) \mid f \text{ ist Wortform, } s \text{ ist Wortbedeutung} \}$
  - (f, s) wird Wort der Sprache genannt
  - Eine Wortform kann mehrere Bedeutungen haben (**Polysemie**), z.B. Bank = Sitzgelegenheit oder Geldinstitut
  - **Morphologie** definiert Abhängigkeiten zwischen verschiedenen, verwandten Wortformen, z.B.
 

gegangen	→	gehen
geht	→	gehen

## WordNet

- WordNet enthält ca.
  - 118.000 Wortformen
  - 90.000 Wortbedeutungen
  - 166.000 Worte (d.h. *(f, s)*-Paare)
- WordNet unterteilt die Worte in die **syntaktische** Kategorien
  - Substantiv (N, noun)
  - Verb (V, verb)
  - Adjektiv (Adj, adjective) und
  - Adverb (Adv, adverb).
  - Präpositionen, Pronomen und Artikel werden nicht betrachtet.
- WordNet beinhaltet weitverbreitete, leicht verständliche **semantische** Relationen zwischen den Wortbedeutungen.

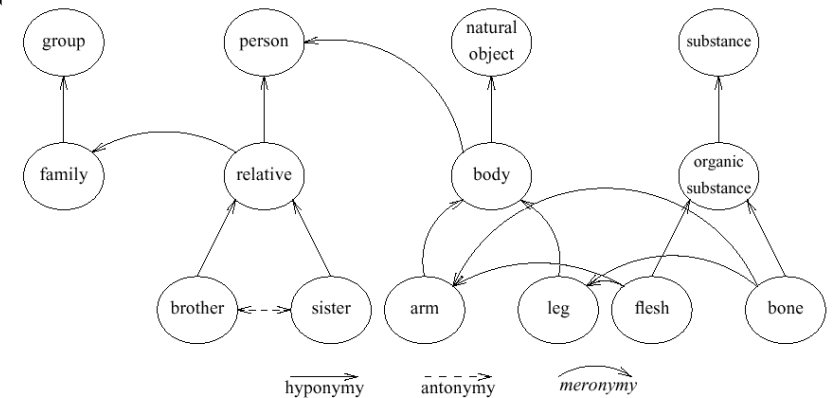
## Semantische Relationen

semantische Relation	syntaktische Kategorien	Beispiele
Synonymie (gleichbedeutend)	N, V, Adj, Adv	Ast, Zweig lesen, schmökern dunkel, finster schnell, flink
Antonymie (gegenteilig)	Adj, Adv, (N, V)	nass, trocken schnell, langsam freundlich, unfreundlich
Hyperonymie (Ober-/Unterbegriff, <i>is-a</i> )	N	Tier, Lebewesen Säugetier, Tier Hund, Säugetier
Meronymie (Teil-von)	N	Henkel, Tasse Schiff, Flotte Mehl, Kuchen
Troponomie	V	flüstern, sprechen

## WordNet

- Semantische Relationen haben teilweise **direkte** Entsprechungen in Modellierungsprimitiven von (objektorientierten) Repräsentationsformalismen
  - **Hyperonymie / Hyponymie** entspricht *is-a*-Beziehung (Ober-/ Unterbegriff)  
siehe Terminologische Logik, UML, Vererbung im ODMG-Objektmodell
  - **Meronymie** entspricht *part-of*-Beziehung
    - Bemerkung: *part-of* kann verschiedene semantische Beziehungen ausdrücken, z.B.  
Komponente-von: *Henkel - Tasse*  
Element-von: *Schiff - Flotte*
  - *WordNet* (und vergleichbare Systeme) erlauben **fließenden Übergang** zwischen linguistischen Informationen und konzeptuellen Schemabeschreibungen bzw. Domänenmodellen

## Auszug aus WordNet



## WordNet online

Web WordNet 2.0 - Microsoft Internet Explorer

WordNet 2.0 Search

Search word: car Find senses

Overview for "car"

The noun "car" has 5 senses in WordNet.

1. **car**, auto, automobile, machine, motorcar -- (4-wheeled motor vehicle; usually propelled by an internal combustion engine; "he needs a car to get to work")
2. **car**, railcar, railway car, railroad car -- (a wheeled vehicle adapted to the rails of railroad; "three cars had jumped the rails")
3. cable car, **car** -- (a conveyance for passengers or freight on a cable railway; "they took a cable car to the top of the mountain")
4. **car**, gondola -- (car suspended from an airship and carrying personnel and cargo and power plant)
5. **car**, elevator car -- (where passengers ride up and down; "the car was on the top floor")

Search for: Meronyms (parts of car), regular of senses:

Show Synonyms, ordered by estimated frequency

Show Coordinate Terms

Show Hypernyms (car is a kind of...)

Show Hyponyms (... is a kind of car), brief

Show Hyponyms (... is a kind of car), full

Show Holonyms (car is a part of...), regular

Show Meronyms (parts of car), regular

Show Meronyms (parts of car), inherited

Return to:

Domain Terms

Familiarity

<http://www.cogsci.princeton.edu/cgi-bin/webwn>  
Informations- and Wissensmanagement WS 2005-2006

## 2.3 Ontologie

### 2.3.1 Überblick

- ursprünglich eine **philosophische** Disziplin
  - zur Untersuchung und Beschreibung der Realität
  - Wissenschaft vom Seienden (vgl. Aristoteles „Metaphysik“ IV, 1)
- In der **Informatik** wird der Begriff Ontologie häufig folgendermaßen definiert:

*"An ontology is an explicit specification of a shared conceptualisation."*

[Gruber 95]

## Ontologien zur Definition der gemeinsamen Sprache

*"An ontology is an explicit specification of a shared conceptualisation."*

- explizite Spezifikation
  - formal, mathematisch eindeutig
- gemeinsame Konzeptualisierung
  - gemeinsames Verständnis eines Anwendungsbereichs
  - getragen von einer Gruppe von Menschen, z.B. einer Abteilung
  - **intensionale** Charakterisierung der relevanten **Begriffe** und **Beziehungen** eines Anwendungsbereichs
    - Beziehungen führen zu **weiteren** relevanten Begriffen
    - Attribute innerhalb einer Begriffsdefinition kennzeichnen **zusätzliche relevante Merkmale** für domänenspezifischen Begriff
  - **intensionale Regeln** und **Constraints**

## Ontologien zur Definition der gemeinsamen Sprache

Mögliche Formalismen:

- **Terminologische Logik**  
(vgl. Vorlesung „Angewandte Informatik 1“)
- **Frame Logic**  
(vgl. Vorlesung „Intelligente Systeme im World Wide Web I“)
- **RDF Schema**  
(vgl. Vorlesung „Intelligente Systeme im World Wide Web I“)
- **OWL**  
(vgl. Vorlesung „Intelligente Systeme im World Wide Web I“)

## Beispiel (ein Teil der Human Resource Management Ontologie in F-Logic)

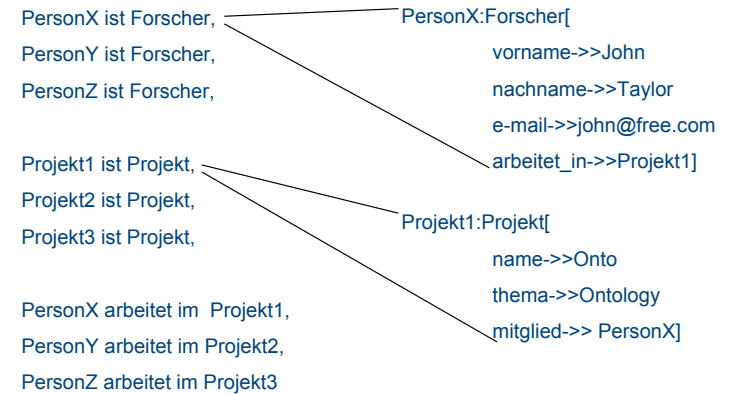
Begriffe	Beziehungen	Regeln
Objekt[]. Person :: Objekt. Mitarbeiter :: Person. Forscher :: Mitarbeiter. Wiss_Mitarbeiter :: Forscher. Wiss_Assistant :: Forscher. Professor :: Forscher. Administrativer_M :: Mitarbeiter. Sekretarin :: Administrativer_M. Techniker :: Administrativer_M.  Projekt :: Objekt.  Forschungsgruppe :: Objekt.  Forschungsgebiet :: Objekt.  Organisation :: Objekt.	Person[ nachname=>>STRING; vorname=>>STRING; telefon=>>STRING; email=>>STRING].  Forscher[ f_gruppe=>>Forschungsgruppe; arbeitet_in=>>Projekt; f_gebiet=>> Forschungsgebiet].  Projekt [ name =>> String; beschreibung =>> String; gruppe=>>Forschungsgruppe; leiter =>> Forscher; mitglied =>> Forscher; thema =>> Forschungsgebiet ].	FORALL Proj1, For1 Proj1 : Projekt [mitglied->> For1] <- For1 :Forscher [arbeitet_in ->> Proj1]  FORALL For1, Proj1, Fg1 For1 [f_gebiet ->> Fg1] <- For1 :Forscher [arbeitet_in ->>Proj1] And Proj1 : Projekt[thema->>Fg1] And Fg1:Forschungsgebiet.

## Beispiel

1. Zustand - Institute XX:

In natural language:

In logical language:

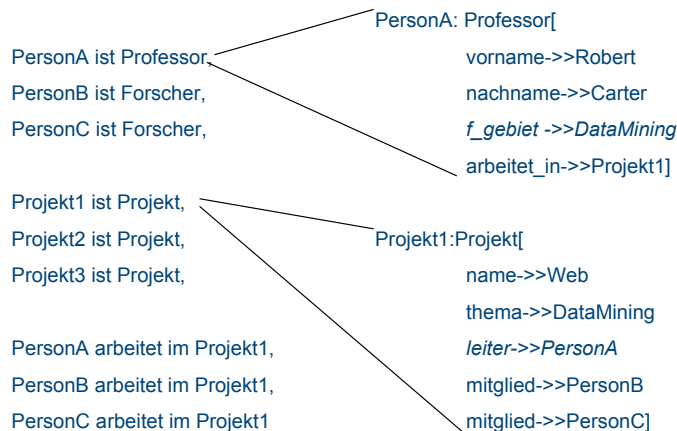


## Beispiel

2. Zustand - Institute YY

In natural language:

In logical language:



## What is an Ontology? A Semiotic View

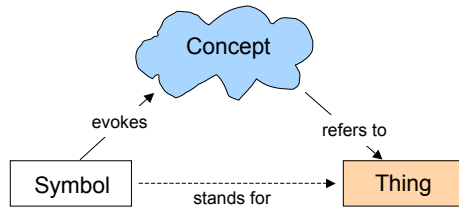
- **Syntax:**  
a set of signs (symbols, lexical entries) that convey meaning for humans (not for machines)
- **Semantics:**  
Relations between signs and things of the real world
- **Pragmatics:**  
Which signs are used for which purpose?
- **Social:**  
Who uses which signs?

### 2.3.2 Ontologies for Communication

#### Context of communication

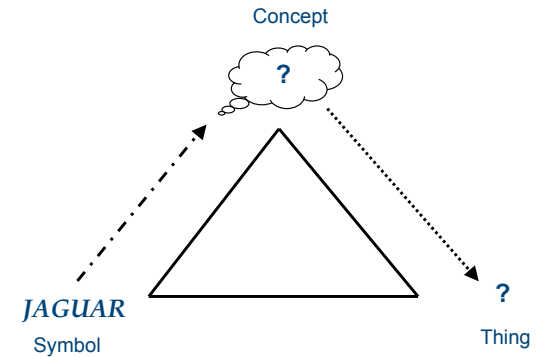
The general context of communication is described by the **meaning triangle**, that defines the interaction between

- symbols,
- concepts and
- things of the world:



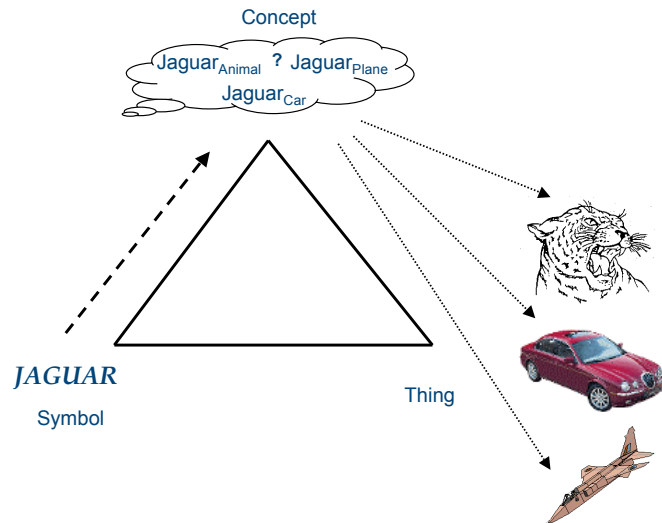
### Mapping from Symbols to Things in World

The relationship between a **symbol** and a **thing** is indirect.

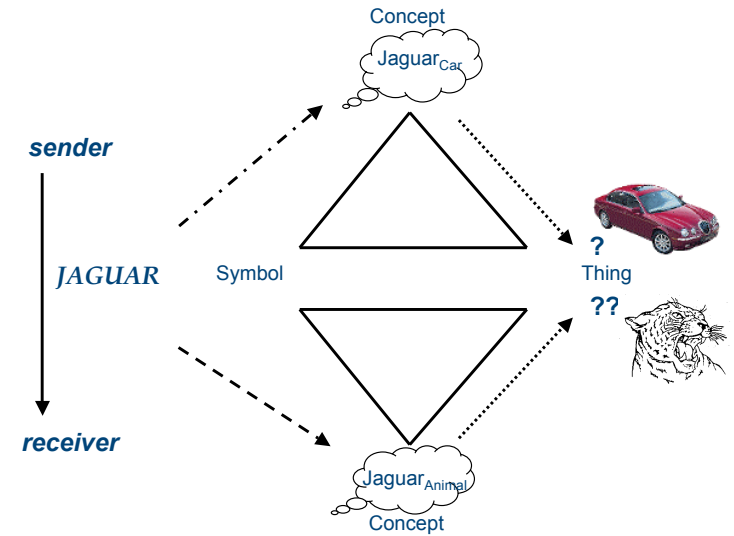


- The link can only be completed when:
- an **interpreter processes the symbol**, which invokes a corresponding concept and then
  - **links that concept to a thing** in the world.

### Mapping from Symbols to Things in World



### Shared Understanding in Communication



## Ontologies for Communication

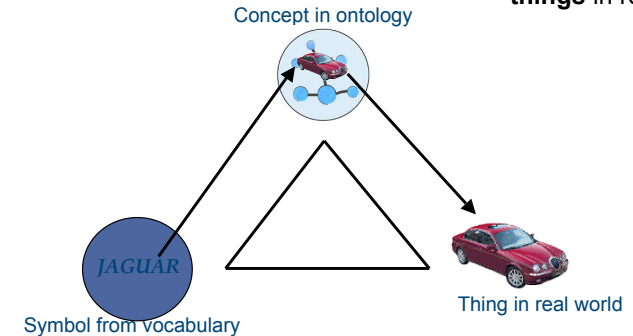
- Ontology refers to an engineering artifact, constituted by
  - a specific **vocabulary** used to describe a certain reality,
  - plus a set of explicit **assumptions** regarding the intended meaning of the vocabulary in some logical language - logical theory.
- Both, vocabulary and assumptions, serve human and software agents to reach a **common understanding** when communicating

## Ontologies for communication

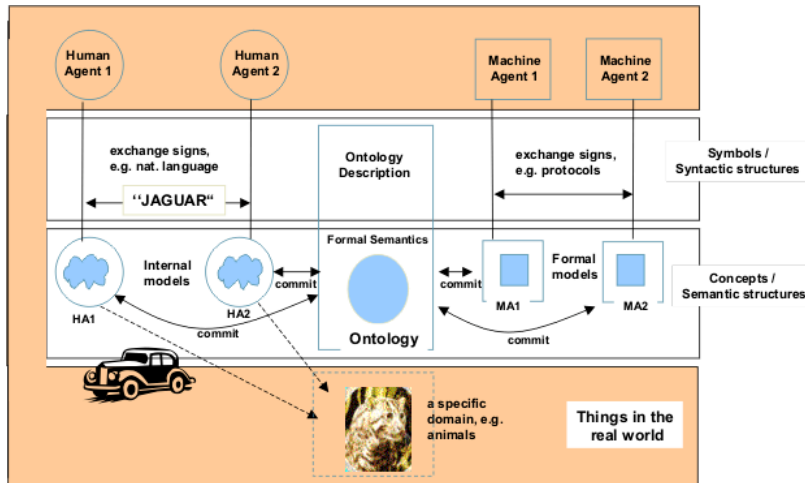
The logical theory specifies:

- relations between symbols and
- it **associates** relations with a semantics that restricts the set of possible interpretations of the symbol.

=> The ontology reduces the number of mapping from **symbols** to **things** in real world



## Ontologies for Communication between Human and/or Software Agents



## Ontologies for communication - Conclusion

*Human and/or software agents can't share knowledge if they don't speak a common language*

*Framework for this common understanding are Ontologies*

## Abstract Model of an Ontology (I)

**Definition:** An ontology is a sign system  $O := (L, F, G, C, H, R, A)$ , which consist of:

- A **lexicon**: The lexicon contains a set of signs (lexical entries) for concepts,  $L_C$ , and a set of signs for relations,  $L_R$ . Their union is the lexicon  $L := L_C \cup L_R$
  - A set  $C$  of **concepts**: About each  $c \in C$  exists at least one statement in the ontology, viz. its embedding in the taxonomy
  - A set of binary **relations**  $R$ :  $R$  denotes a set of binary relations. They specify pairs of domain and ranges  $(C_D, C_R)$  with  $C_D, C_R \in C$
- The functions  $d$  and  $r$  applied to a binary relation  $R$  yield the corresponding domain and range concepts  $C_D$  and  $C_R$ , respectively

## Abstract Model of an Ontology (Ia)

Example:

**lexicon**:  $L = \{ \text{"employee", "Angestellter", "Angestellte", "Organisation", "Projekt", "member", "participant", "client", "participate", ...} \}$

**concepts**:  $C = \{ \text{Person, Employee, Manager, Project, Company, FinanceComp, ...} \}$

**relations**:  $R = \{ \text{participantOf, member, client, ...} \}$

$d = \{ (\text{participantOf, Person}), (\text{member, Project}), (\text{client, Project}) \}$ ,

$r = \{ (\text{participantOf, Project}), (\text{member, Person}), (\text{client, Company}) \}$

## Abstract Model of an Ontology (II)

- Two **reference functions**  $F, G$ , with  $F: 2^{L_C} \rightarrow 2^C$  and  $G: 2^{L_R} \rightarrow 2^R$ .  $F$  and  $G$  link **lexical entries**  $\{l_1, l_2, \dots, l_n\} \subset L$  to the **concepts** and **relations** they refer to, respectively, in the given ontology.

In general, one lexical entry may refer to several concepts or relations and one concept or relation may be referred to by several lexical entries.

**Remark:** In order to map easily back and forth and because there is a **n** to **m** mapping between lexicon and concepts/relations,  $F$  and  $G$  are defined on sets rather than on single objects.

Example:

**reference function**  $F$ :  $\{ (\{ \text{"employee", "Angestellter", "Angestellte"} \}, \{ \text{Employee} \}), (\{ \text{"organisation", "Unternehmen"} \}, \{ \text{Company} \}), (\{ \text{"Bank"} \}, \{ \text{Bank}_{\text{Geldinstitut}}, \text{Bank}_{\text{Sitzgelegenheit}} \}), \dots \}$

**reference function**  $G$ :  $\{ (\{ \text{"member", "participant"} \}, \{ \text{member} \}), (\{ \text{"participate"} \}, \{ \text{participantOf} \}), (\{ \text{"client"} \}, \{ \text{client} \}), \dots \}$

## Abstract Model of an Ontology (III)

- A **taxonomy**  $H$ : Concepts are taxonomically related by the irreflexive, acyclic, transitive relation  $H$ , ( $H \subset C \times C$ ).

$H(C_i, C_j)$  means that  $C_i$  is a subconcept of  $C_j$

- A set of ontology **axioms**,  $A$

Example:

**taxonomy**:  $H = \{ (\text{Manager, Employee}), (\text{Employee, Person}), (\text{FinanceComp, Company}) \}$

**axiom**:

in natural language:

IF Person X is participantOf Project Y THEN Project Y has as member Person X

### Distinguishing ontology and knowledge base

Criteria:	Ontology	Knowledge base
Set of logical statements	yes	yes
Theory	general theory	theory of particular circumstances
Statements are mostly	intensional	extensional
Construction	rather stable	continuous change
Description logics	T-Box	A-Box

**Note:**

Overall the decision to model some relevant part of the domain in the ontology vs. in the knowledge base is often based on gradual distinctions and driven by the needs of the application.

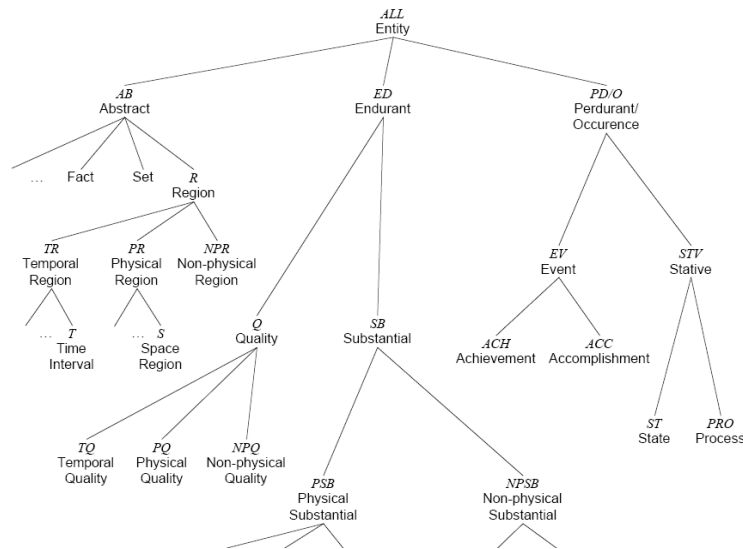
### Top level ontologies

- Top (higher, upper) level ontologies represent **commonsense sorts** of things that exist in the world and relations between them.
- In the last several years a number of top level ontologies have become generally available to the knowledge representation, knowledge engineering and natural language processing research communities.

**Examples:**

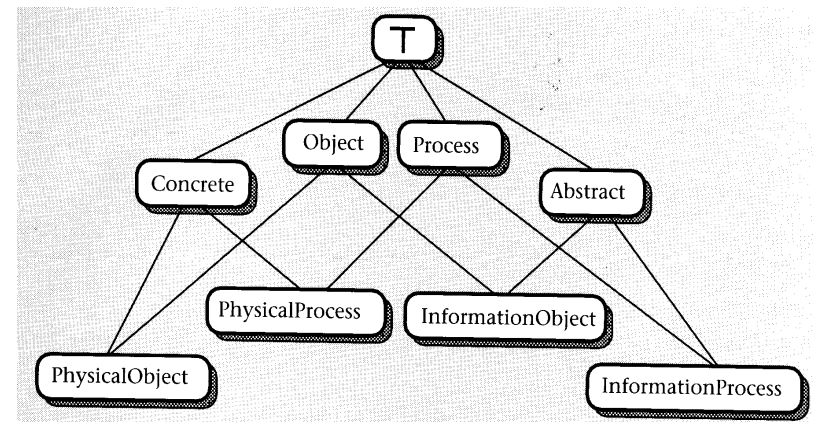
- DOLCE
- Sowa's top-level ontology
- Cycorp
- Mikrokosmos
- WordNet (first two levels)

### Top level ontologies - DOLCE



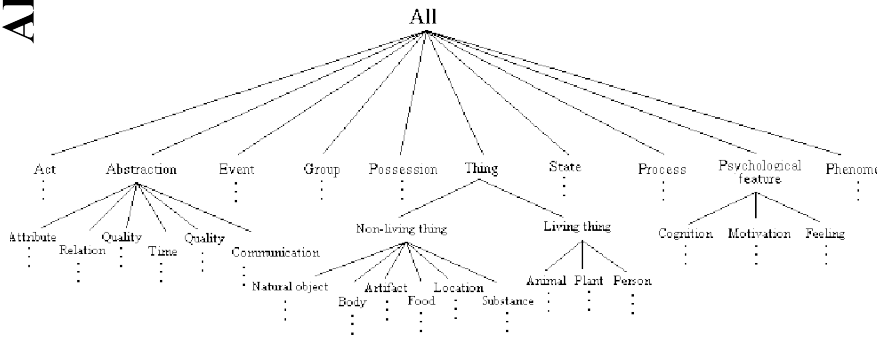
aus [Gangemi, Guarino, et al. 2002]

### Top level ontologies - John Sowa's Top-Level-Ontologie



aus [Friedman-Noy, Hafner 97]

## Top level hierarchies in WordNet



## 2.3.3 Konzeptuelles Datenbankschema vs. Ontologie

from [N. F. Noy & M. Klein, 2002]

1. Ontologies themselves are data to the extent to which database schemas have never been
  - Ontologies are used
    - as controlled vocabularies,
    - to drive search,
    - to provide navigation through large collections of documents,
    - to provide organization and configuration structure of Web sites.
  - result of an ontology query can include elements of the ontology itself
2. Classes and instances can be the same
  - Databases make a clear distinction between the schema and the instances
  - It is hard to distinguish where an ontology ends and instances begin

## Konzeptuelles Datenbankschema vs. Ontologie

3. Ontologies themselves incorporate (explicit) semantics
  - In database schemas the semantics
    - were specified explicitly at database-design time, but the specification has not become part of database specification,
  - Ontology definitions can be interpreted as a set of logical axioms
4. Ontologies are more often reused
  - A database schema does not usually directly reuse or extend existing schemas.
  - Ontologies often reuse and extend other ontologies,
    - they are not bound to a specific system

## Konzeptuelles Datenbankschema vs. Ontologie

5. Ontologies are de-centralized by nature
  - Database schema development and update is a centralized process
  - Ontology development is a much more de-centralized and collaborative process
    - no centralized control over who uses a particular ontology
6. Ontology data models are richer
  - The number of representation primitives in many ontologies is much larger than in a typical database schema:
    - extended cardinality constraints,
    - inverse properties,
    - transitive properties,
    - disjoint classes,...

## 2.3.4 Objektorientierte Modellierung (OO)

- zentrales Modellierungsprimitiv ist die **Klasse**
  - fasst ähnliche Objekte zusammen
  - definiert deren Attribute
  - definiert deren Verhalten über Methoden
- Relationen stellen Beziehungen zw. Klassen dar
  - Subklassen-Beziehung (mit (Mehrfach-)Vererbung)
  - Teil-Von-Beziehung
  - weitere, benutzerdefinierte
- Vergleich zu Ontologien
  - Modellierungsprimitive sehr ähnlich
  - OO-Modellierung und -Programmierung stark korreliert
  - OO kennt keine Regeln/Axiome
  - OO kodiert regelhafte Zusammenhänge in Methoden
  - Verhalten/Methoden in Ontologien unbekannt