

## Kapitel 3

# Ontologiebasiertes Wissensmanagement

- 3.1 Anforderungen des Wissensmanagements
- 3.2 Ontologien und verwandte Ansätze als Lösungsansätze
- 3.3 Ontologien
  - 3.3.1 Ontobroker
  - 3.3.2 SEAL - Semantisches Portal
- 3.4 Zusammenfassung

## 3.1 Wissensmanagement

- Ziel von Wissensmanagement:
    - benötigtes Wissen
      - zur richtigen Zeit,
      - am richtigen Ort,
      - in passender Form und Granularität
- zur Verfügung zu stellen

# Wissensverteilung und Zugriff

## ■ Pull-Ansatz

- Browsing (Wissenslandschaften)
- Suche
  - im Web oder Intranet
  - in Dokumenten-Repositories
  - in Archiven ...
- Anfrage an Organizational Memory

## ■ Push-Ansatz

- intelligente (persönliche) Agenten
- EMail-Verteilerlisten
- Newsgroups

## ■ Filter-Techniken

- einsetzbar bei Push und Pull
- fokussieren auf relevante Informationen

## *"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 (aufgabenspezifisches) Vokabular

## 3.2 Ontologien und verwandte Ansätze

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

## 3.2.1 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](#).

## 3.2.2 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

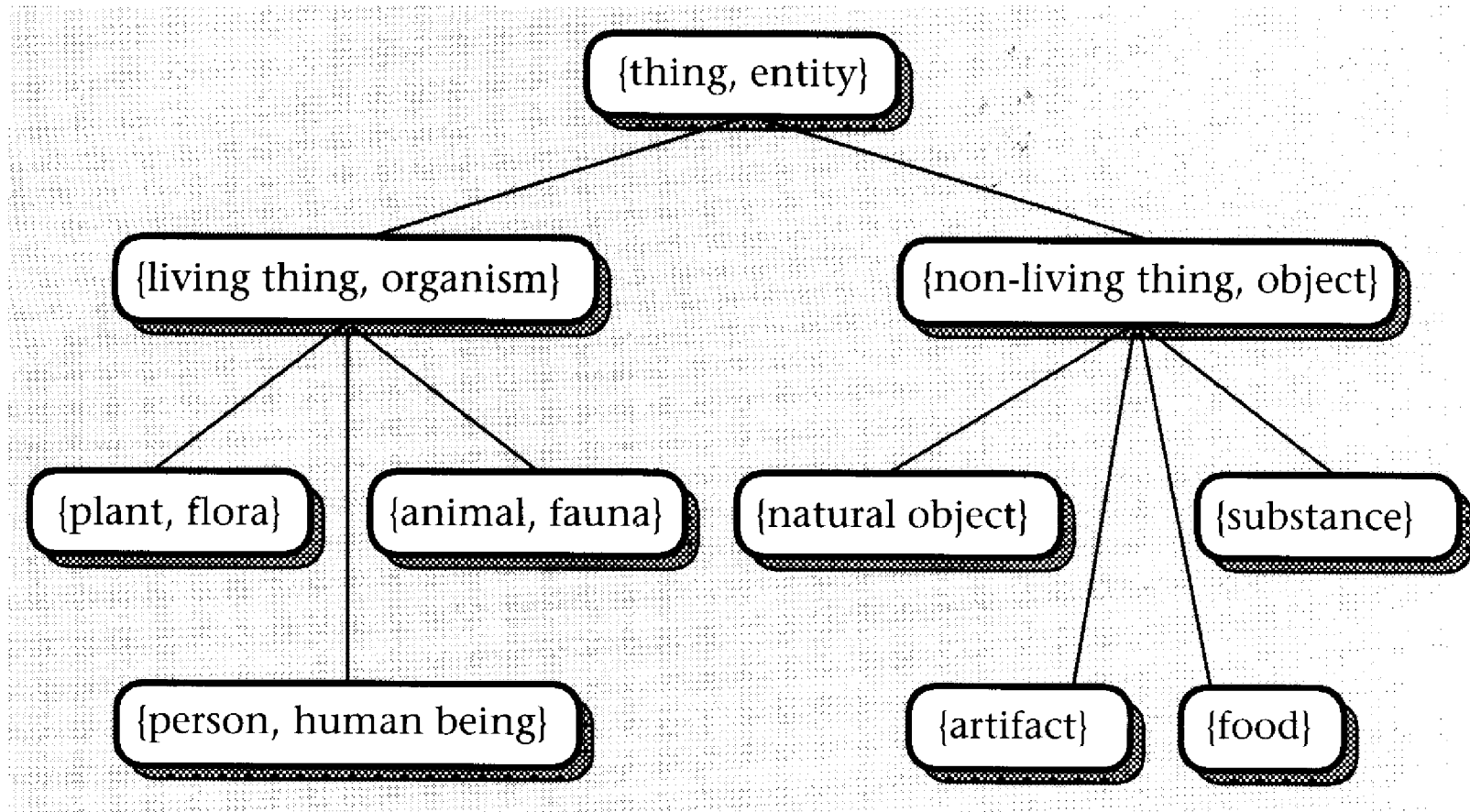
semantische Relation	syntaktische Kategorien	Beispiele
Synonymie (gleichbedeutend)	N, V, Adj, Adv	Ast, Zweig lesen, schmökern dunkel, finster schnell, flink
Antonymie (gegenteilig)	Adj, Adv, (N, V)	nass, trocken schnell, langsam freundlich, unfreundlich
Hyperonymie (Ober-/Unterbegriff, <i>is-a</i> )	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
  - **Meronomie** 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

aus [Friedman-Noy, Hafner 97]



### 3.2.3 Konzeptuelles Datenbankschema

- ein konzeptuelles Datenbankschema stellt eine Beschreibung von Begriffen (Konzepten) und Beziehungen zur Verfügung
  - siehe z.B. erweitertes Entity-Relationship-Modell (EER)
- im Vergleich zu Ontologien sind konzeptuelle Datenbankschemata
  - beschränkter in Bezug auf verwendete Modellierungsprimitive
  - nicht zwingend auf die gemeinsame Nutzung durch verschiedene Nutzergruppen / Anwendungen ausgerichtet
  - nicht Bestandteil der operativ eingesetzten Datenbank-Anwendung

## 3.2.4 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



## 3.3 Ontologie

### 3.3.1 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 **Konzepte** und **Beziehungen** eines Anwendungsbereichs
    - Beziehungen führen zu **weiteren** relevanten Konzepten
    - Attribute innerhalb einer Konzeptdefinition kennzeichnen **zusätzliche relevante Merkmale** für domänenspezifisches Konzept
  - **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“)
- ***DAML+OIL*** (vgl. Vorlesung „Intelligente Systeme im World Wide Web I“)

## Beispiel (ein Teil der Ontolce Ontologie in F-Logic)

Konzepte	Beziehungen	Regeln
Object [ ]. Person :: Object. Employee :: Person. Manager :: Employee. Consultant :: Employee.  Project :: Object.  Company :: Object. Manufacturer :: Company. FinanceComp :: Company. Insurer :: FinanceComp. LifeInsurer :: Insurer. Bank :: FinanceCompany.  Location :: Object.	Person [   firstName ==>> String;   lastName ==>> String;   email ==>> String;   phone ==>> String;   participantOf ==>> Project;   hasCompExperience ==>>     Company;   address ==>> Location]  Project [   projectname ==>> String;   projectgoal ==>> String;   client ==>> Company;   member ==>> Person;   leader ==>> Person].	FORALL Proj1, Pers1 Proj1 : Project [member ->> Pers1]  ↔ Pers1 : Person [participantOf ->> Proj1].  FORALL Pers1, Proj1, Comp1 Proj1 : Project [member ->> Pers1, client ->> Comp1]  → Pers1 : Person [hasCompExperience ->>     Comp1].

## Beispiel Ontolce

### 1. Zustand - Unternehmen XX:

In natural language:

PersonX is Person,

PersonY is Person,

PersonZ is Person,

ProjectX is Project,

ProjectY is Project,

ProjectZ is Project,

PersonX is participant of ProjectX,

PersonY is participant of ProjectY,

PersonZ is participant of ProjectZ,

In logical language:

PersonX:Person[

firstName->>John

lastName->>Taylor

e-mail->>john@free.com

participantOf->>ProjectX]

ProjectX:Project[

projectName->>Onto

projectGoal->>XX

member->> PersonX]

## Beispiel Ontolce

### 2. Zustand: - Unternehmen YY

In natural language:

PersonA is Employee,  
PersonB is Employee,  
PersonC is Employee,  
ProjectA is Project,  
ProjectB is Project,  
ProjectC is Project,  
  
PersonA is participant of ProjectA,  
PersonB is participant of ProjectA,  
PersonC is participant of ProjectA,

In logical language:

PersonA: Employee[  
  
firstName->>Robert  
lastName->>Carter  
hasCompExperience->>Telecom  
participantOf->>ProjectA]  
  
ProjectA:Project[  
  
projectName->>Web  
client->>Telecom  
leader->>PersonA  
member->>PersonB  
member->>PersonC]

## What is an Ontology? A Semiotic View

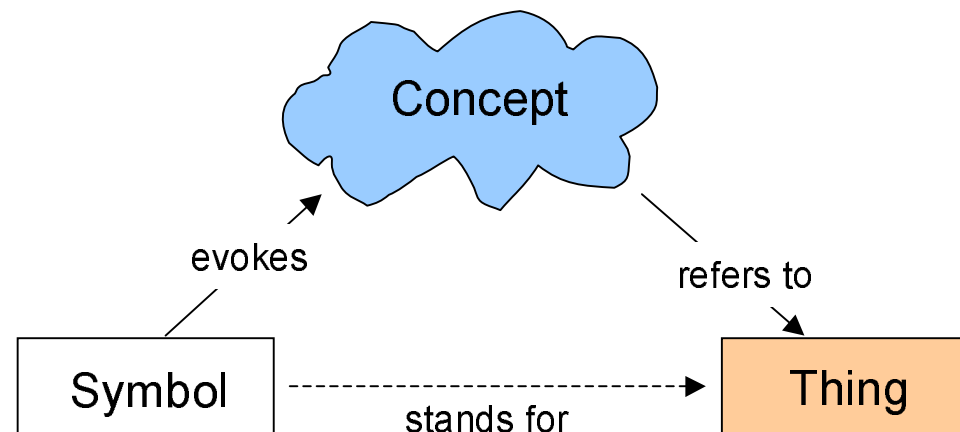
- **Syntax:** a set of signs (symbols, lexical entries) that convey meaning for humans (not for machines)
- **Semantics:** Relations between signs and things of the real world
- **Pragmatics:** Which signs are used for which purpose?
- **Social:** Who uses which signs?

## 3.3.2 Ontologies for Communication

### Context of communication

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

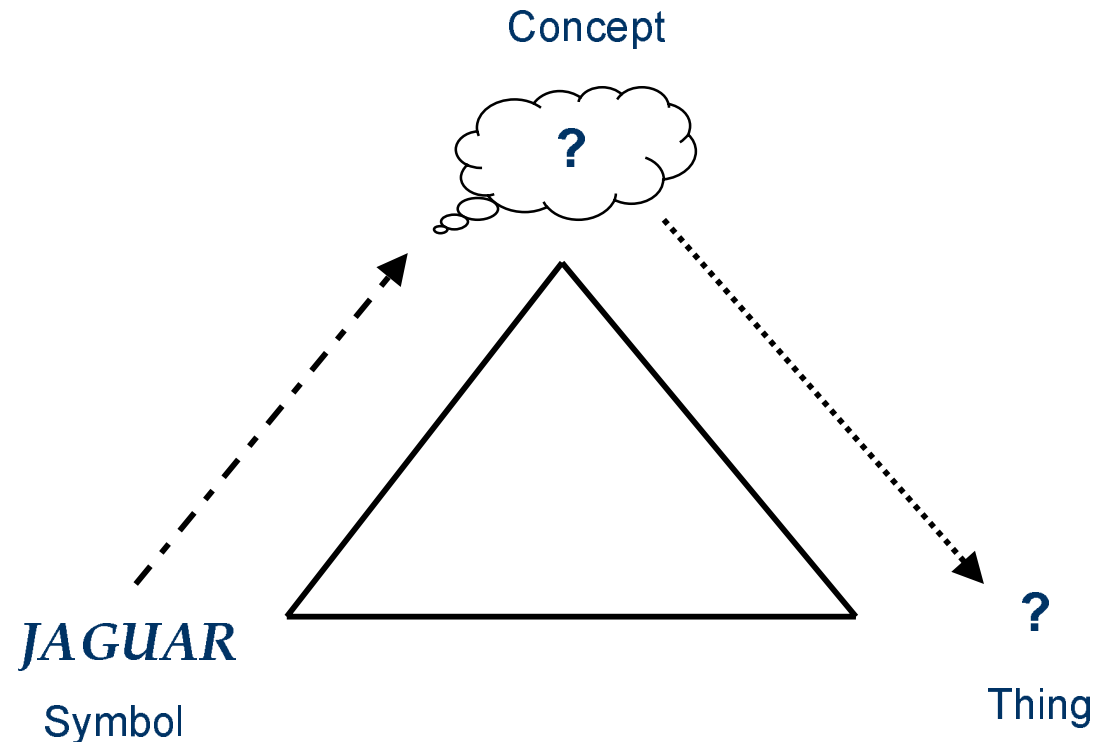
- symbols,
- concepts and
- things of the world:





## Mapping from Symbols to Things in World

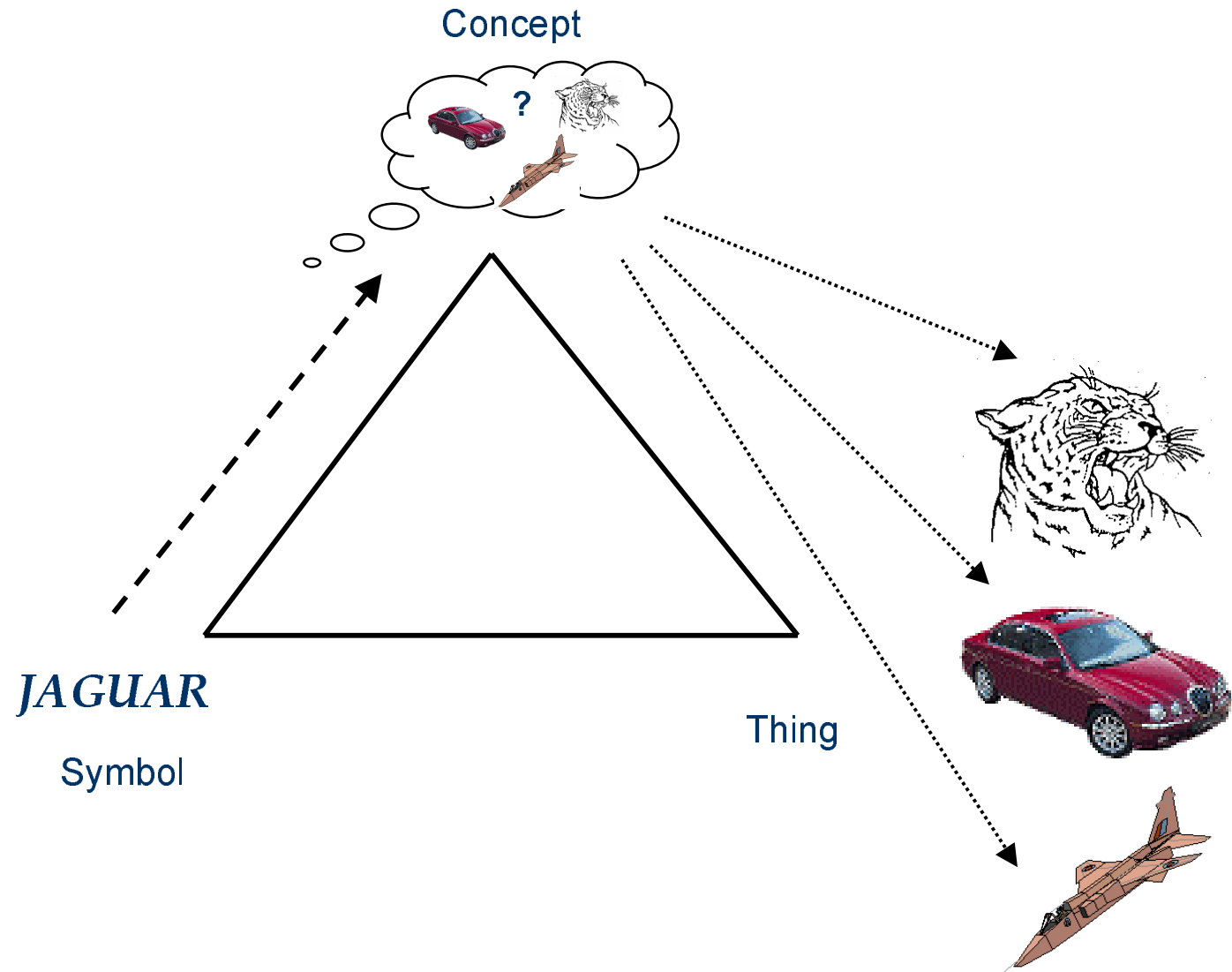
The relationship between a **symbol** and a **thing** is indirect.



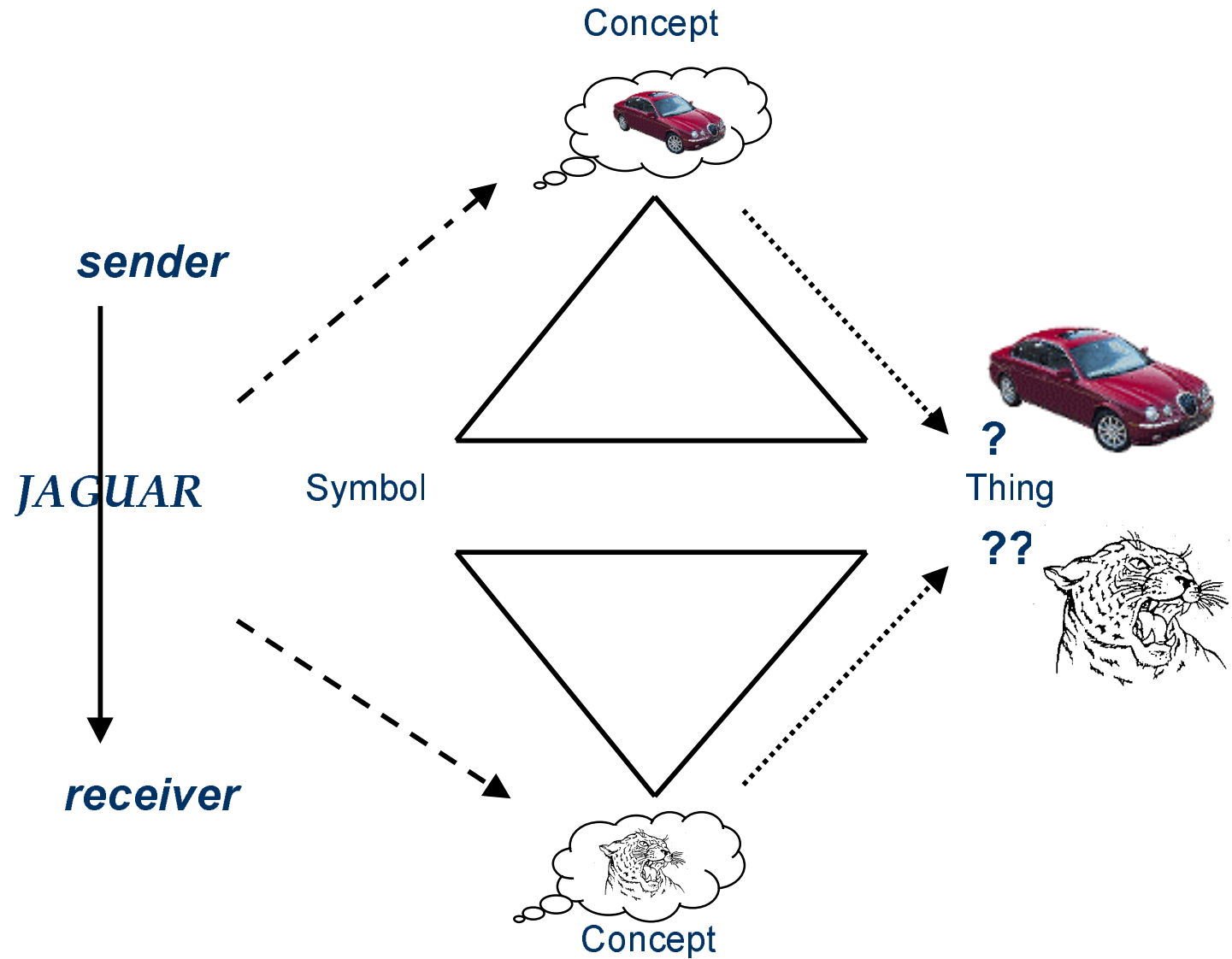
The link can only be completed when:

- 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

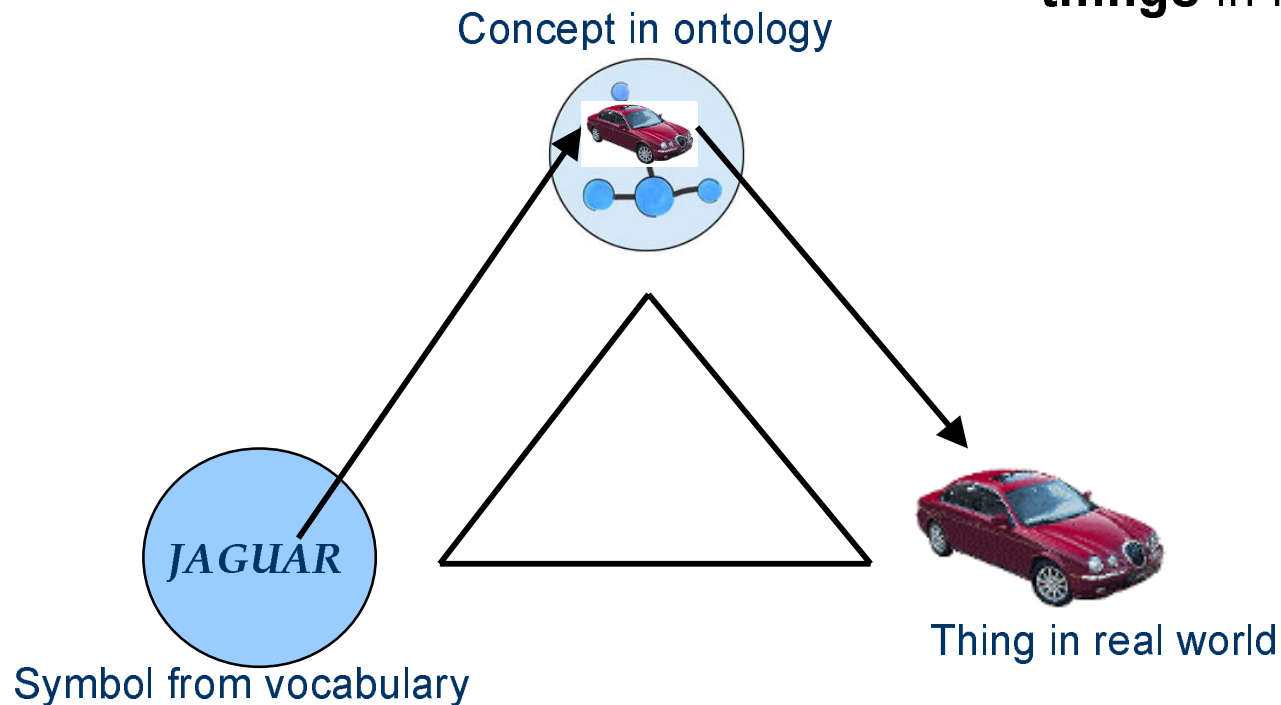
- Ontology refers to an engineering artifact, constituted by
  - a specific **vocabulary** used to describe a certain reality,
  - plus a set of explicit **assumptions** regarding the intended meaning of the vocabulary in some logical language - logical theory.
  
- Both, vocabulary and assumptions,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

## Ontologies for communication

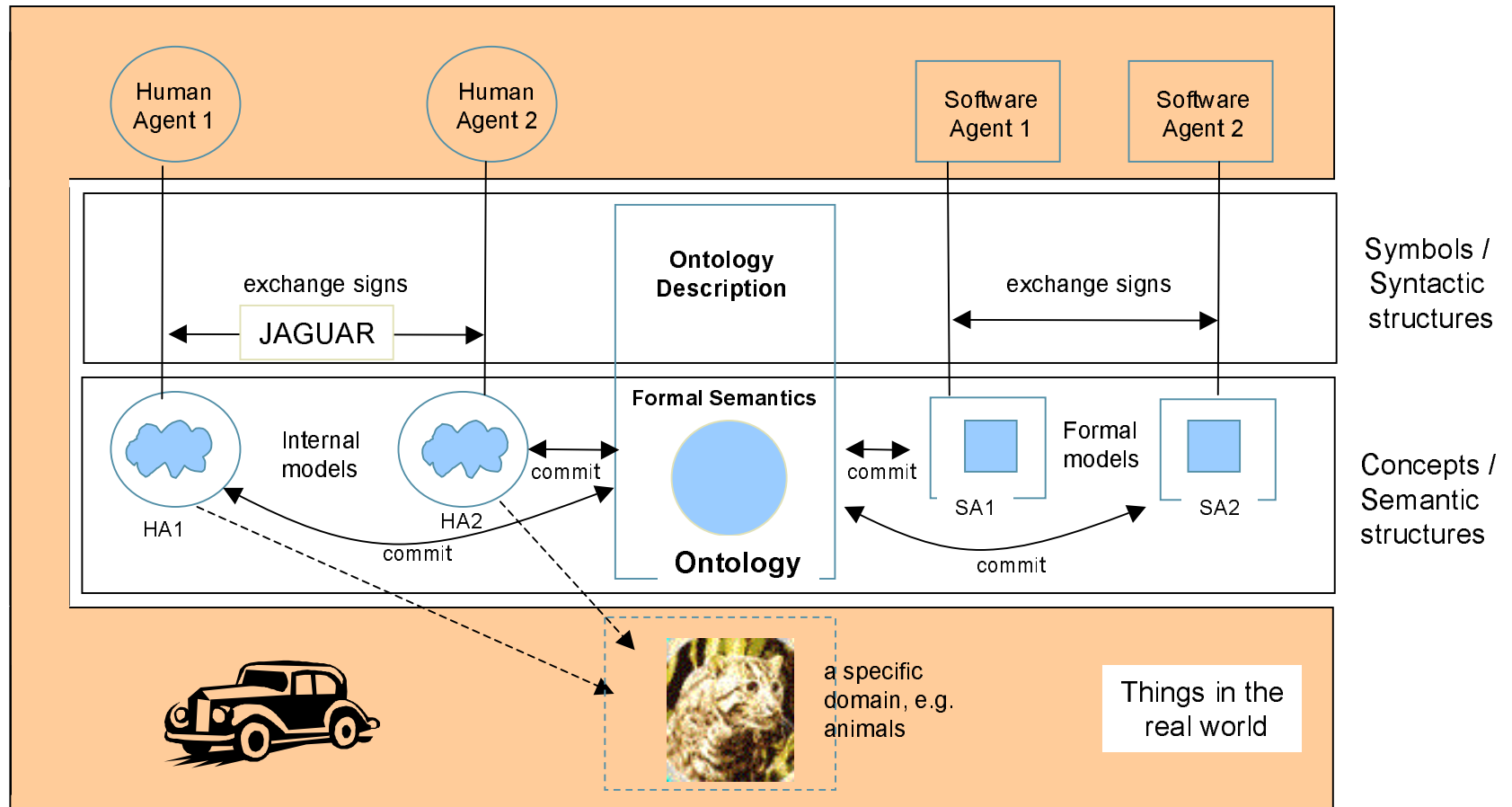
The logical theory specifies:

- relations between symbols and
- it **apprehends** 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,  $L_t$ . Their union is the lexicon  $L := L_c \cup L_t$
- 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  $(D, R)$  with  $D, R \in C$

The functions  $d$  and  $r$  applied to a binary relations  $R$  yield the corresponding domain and range concepts  $D$  and  $R$ , 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  $(F \subseteq Lc \times C)$  and  $(G \subseteq Lt \times R)$ .  $F$  and  $G$  link **lexical entries**  $\{Li\} \subset L$  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  $F$ :*  $\{("employee", Employee), ("Angestellter", Employee), ("Angestellte", Employee), ..., („organisation“, Company), ...\}$

*reference function  $G$ :*  $\{("member", member), ("participant", member), ("participate", participantOf), („client“, client), ...\}$

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

*axiom*:

in natural language:

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

## Distinguishing ontology and knowledge base

<b>Criteria:</b>	<b>Ontology</b>	<b>Knowledge base</b>
<b>Set of logical statements</b>	yes	yes
<b>Theory</b>	general theory	theory of particular circumstances
<b>Statements are mostly</b>	intensional	extensional
<b>Construction</b>	rather stable	continuous change
<b>Description logics</b>	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:

Sowa's top-level ontology

Cycorp

Mikrokosmos

Generalised Upper Model:

Sensus

WordNet (first two levels)

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

aus [Friedman-Noy, Hafner 97]

