

Intelligent Systems on the World Wide Web

3 Ontologien

Lecture Slides
Steffen Staab
Karlsruhe University
With Acknowledgements to Nenad Stojanovic

"Gemeinsame Sprache"

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

[T. Davenport]

Ontologie

- 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, formal specification of
a shared conceptualisation."*

[Gruber 95]

Ontologien zur Definition der gemeinsamen Sprache

Mögliche Formalismen:

- **Terminologische Logik** (vgl. Vorlesung „Angewandte Informatik 1")
- **Frame Logic**
- **RDF Schema**
- **DAML+OIL**

Ontology



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**

Slide 5

Ontology



"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

Slide 6

Ontology



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

Slide 7

Ontology



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

Slide 8

Ontology

Scharade

Was bedeutet „Vespern“?



Slide 9

Ontology

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].



Slide 10

Ontology

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?



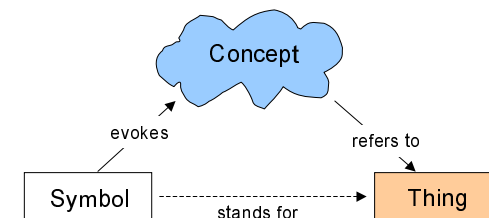
Slide 11

Ontology

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:



In the tradition of Saussure, Peirce und Frege



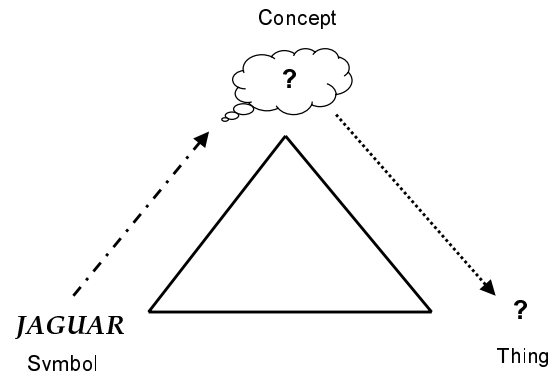
Slide 12

Ontology



Mapping from Symbols to Things in World

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



The link can only be completed when:

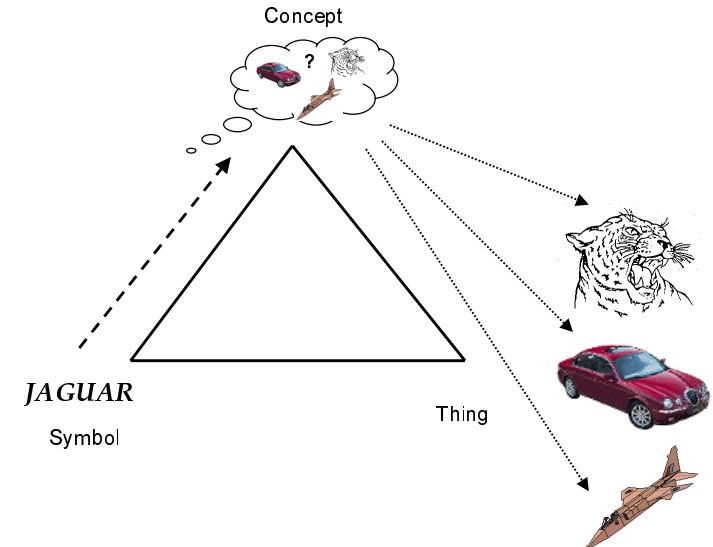
- an **interpreter processes the symbol**, which invokes a corresponding concept and then
- **links that concept to a thing** in the world.

Slide 13

Ontology



Mapping from Symbols to Things in World

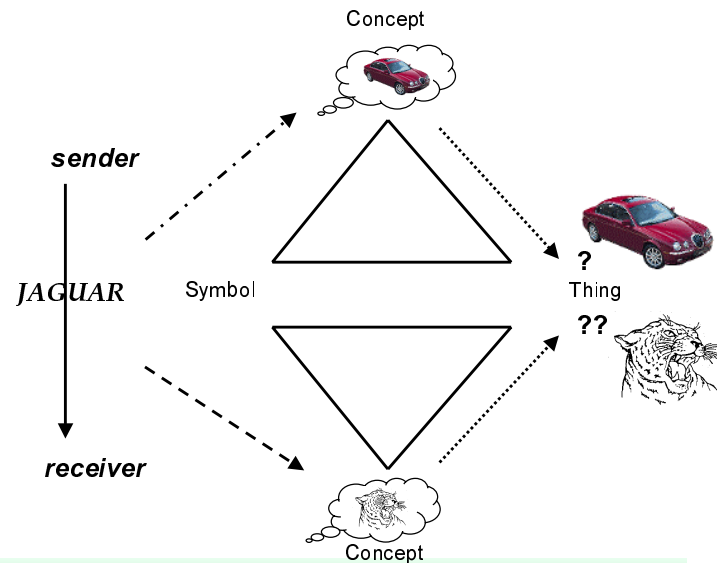


Slide 14

Ontology



Shared Understanding in Communication



Slide 15

Ontology



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

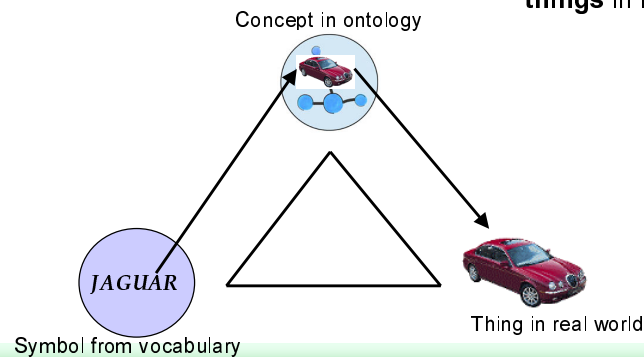
Slide 16

Ontologies for communication

The logical theory specifies:

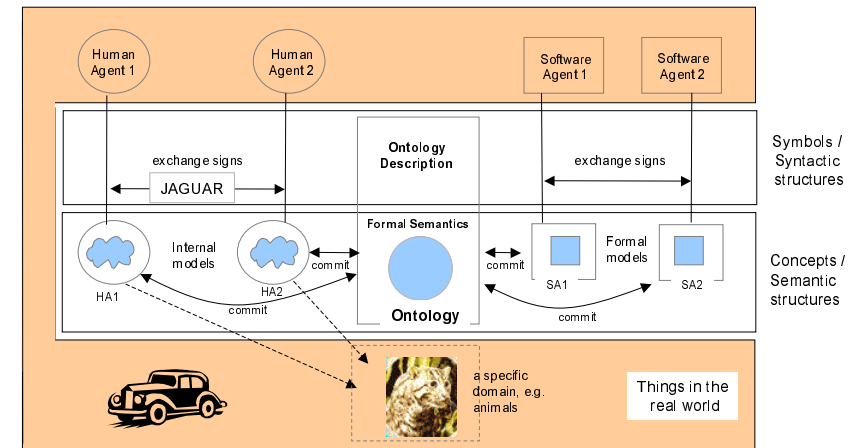
- relations between symbols and
- it **apprehends** relations with a semantics that restricts the set of possible interpretations of the symbol.

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



Slide 17

Ontologies for Communication between Human and/or Software Agents



Slide 18

Ontologie

„im weiteren Sinne“ umfasst:

- Stakeholder
- Shareholder
- Situation
- Etc.

„im engeren Sinne“ umfasst:

- Nur die formalen abstrakten Strukturen
- Nur Dinge, die man für die Anwendung auf dem Rechner modellieren möchte

Slide 19

Abstract ontology

Definition 1. Let \mathcal{L} be a logical language having a formal semantics in which inference rules can be expressed. An *abstract ontology* is a structure $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ consisting of

Slide 20

Abstract ontology



Definition 1. Let \mathcal{L} be a logical language having a formal semantics in which inference rules can be expressed. An *abstract ontology* is a structure $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ consisting of

- two disjoint sets C and R whose elements are called *concepts* and *relations*, resp.,

Slide 21

Abstract ontology



Definition 1. Let \mathcal{L} be a logical language having a formal semantics in which inference rules can be expressed. An *abstract ontology* is a structure $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ consisting of

- two disjoint sets C and R whose elements are called *concepts* and *relations*, resp.,
- a partial order \leq_C on C , called *concept hierarchy* or *taxonomy*,

Slide 22

Abstract ontology



Definition 1. Let \mathcal{L} be a logical language having a formal semantics in which inference rules can be expressed. An *abstract ontology* is a structure $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ consisting of

- two disjoint sets C and R whose elements are called *concepts* and *relations*, resp.,
- a partial order \leq_C on C , called *concept hierarchy* or *taxonomy*,
- a function $\sigma: R \rightarrow C \times C$ called *signature*,

Slide 23

Abstract ontology



Definition 1. Let \mathcal{L} be a logical language having a formal semantics in which inference rules can be expressed. An *abstract ontology* is a structure $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ consisting of

- two disjoint sets C and R whose elements are called *concepts* and *relations*, resp.,
- a partial order \leq_C on C , called *concept hierarchy* or *taxonomy*,
- a function $\sigma: R \rightarrow C \times C$ called *signature*,
- a partial order \leq_R on R where $r_1 \leq_R r_2$ implies $\sigma(r_1) \leq_{C \times C} \sigma(r_2)$, for $r_1, r_2 \in R$, called *relation hierarchy*.

Slide 24

Ontology

Abstract ontology



Definition 1. Let \mathcal{L} be a logical language having a formal semantics in which inference rules can be expressed. An *abstract ontology* is a structure $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ consisting of

- two disjoint sets C and R whose elements are called *concepts* and *relations*, resp.,
- a partial order \leq_C on C , called *concept hierarchy* or *taxonomy*,
- a function $\sigma: R \rightarrow C \times C$ called *signature*,
- a partial order \leq_R on R where $r_1 \leq_R r_2$ implies $\sigma(r_1) \leq_{C \times C} \sigma(r_2)$, for $r_1, r_2 \in R$, called *relation hierarchy*.
- and a set IR of inference rules expressed in the logical language \mathcal{L} .

Slide 25

Ontology

Abstract ontology



Definition 1. Let \mathcal{L} be a logical language having a formal semantics in which inference rules can be expressed. An *abstract ontology* is a structure $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ consisting of

- two disjoint sets C and R whose elements are called *concepts* and *relations*, resp.,
- a partial order \leq_C on C , called *concept hierarchy* or *taxonomy*,
- a function $\sigma: R \rightarrow C \times C$ called *signature*,
- a partial order \leq_R on R where $r_1 \leq_R r_2$ implies $\sigma(r_1) \leq_{C \times C} \sigma(r_2)$, for $r_1, r_2 \in R$, called *relation hierarchy*.
- and a set IR of inference rules expressed in the logical language \mathcal{L} .

The function $\text{dom}: R \rightarrow C$ with $\text{dom}(r) := \pi_1(\sigma(r))$ gives the *domain* of r , and the function $\text{range}: R \rightarrow C$ with $\text{range}(r) := \pi_2(\sigma(r))$ gives its *range*.

Slide 26

Ontology

Lexicon



Definition 2. A *lexicon* for an abstract ontology $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ is a structure $Lex := (S_C, S_R, Ref_C, Ref_R)$ consisting of

Ontology

Lexicon



Definition 2. A *lexicon* for an abstract ontology $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ is a structure $Lex := (S_C, S_R, Ref_C, Ref_R)$ consisting of

- two sets S_C and S_R whose elements are called *signs (lexical entries) for concepts* and *relations*, resp.,

Slide 27

Slide 28

Ontology

Lexicon



Definition 2. A *lexicon* for an abstract ontology $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ is a structure $Lex := (S_C, S_R, Ref_C, Ref_R)$ consisting of

- two sets S_C and S_R whose elements are called *signs (lexical entries) for concepts and relations*, resp.,
- and two relations $Ref_C \subseteq S_C \times C$ and $Ref_R \subseteq S_R \times R$ called *lexical reference assignments for concepts/relations*, resp.

Slide 29

Ontology

Lexicon



Definition 2. A *lexicon* for an abstract ontology $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ is a structure $Lex := (S_C, S_R, Ref_C, Ref_R)$ consisting of

- two sets S_C and S_R whose elements are called *signs (lexical entries) for concepts and relations*, resp.,
- and two relations $Ref_C \subseteq S_C \times C$ and $Ref_R \subseteq S_R \times R$ called *lexical reference assignments for concepts/relations*, resp.

Based on Ref_C , we define, for $s \in S_C$,

$$Ref_C(s) := \{c \in C \mid (s, c) \in Ref_C\}$$

Slide 30

Ontology

Lexicon



Definition 2. A *lexicon* for an abstract ontology $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ is a structure $Lex := (S_C, S_R, Ref_C, Ref_R)$ consisting of

- two sets S_C and S_R whose elements are called *signs (lexical entries) for concepts and relations*, resp.,
- and two relations $Ref_C \subseteq S_C \times C$ and $Ref_R \subseteq S_R \times R$ called *lexical reference assignments for concepts/relations*, resp.

Based on Ref_C , we define, for $s \in S_C$,

$$Ref_C(s) := \{c \in C \mid (s, c) \in Ref_C\}$$

and, for $c \in C$,

$$Ref_C^{-1}(c) := \{s \in S \mid (s, c) \in Ref_C\}.$$

Ref_R and Ref_R^{-1} are defined analogously.

Slide 31

Ontology

(Concrete) Ontology



Definition 3. A (concrete) *ontology* (in the narrow sense) is a pair (\mathcal{O}, Lex) where \mathcal{O} is an abstract ontology and Lex is a lexicon for \mathcal{O} .

Slide 32

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*