


## **Darstellung von Ontologien im Semantic Web - RDFS-**

Cristina Vertan


### **Inhalt**

- Was kann man mit RDF nicht ausdrücken ? 
- Was sind Ontologien ?
- Wie entwirft man eine Ontologie?
- Wie beschreibt man eine Ontologie im Semantic Web ?

## Grenze von RDF

- Alle RDF-tags geben Informationen über Bedeutungen von Wörtern, Ausdrücken, Texte, Webseiten aber:
- Woher soll ein automatischer Prozess wissen, welche Semantik diese Tags haben ?
- Z.B. woher soll ein Prozess wissen, dass "Altstadt" ein Unterbegriff von "Stadt" ist. Oder dass er deutsche Namen nur für siebenbürgische Städte suchen soll und nicht für alle Städte in Rumänien ?
- Für solche inhaltliche Beziehungen muss man die Tags in einer Struktur organisieren

## Inhalt

- Was kann man mit RDF nicht ausdrücken ?
  - Was sind Ontologien ?
  - Wie entwirft man eine Ontologie?
  - Wie beschreibt man eine Ontologie im Semantic Web ?
- 

## Was ist eine Ontologie ? -1-

- Ursprünglich war es ein Begriff aus der Philosophie (seit Aristoteles) für:
  - Ein Forschungsgebiet der Metaphysik, das sich mit der Natur und Existenz beschäftigt oder
  - Eine spezielle Theorie über Existenz und Existenztypen
- In der Mathematik (seit dem XIX. Jahrhundert): Eine formale Ontologie ist eine Logik-Theorie
- In der Informatik - eine grobe Definition: Eine **Konzepthierarchie**

## Was ist eine Ontologie? -2-

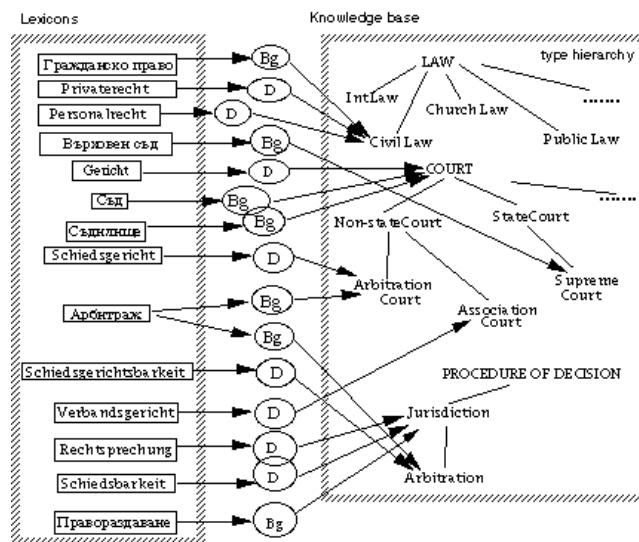
- Eine Ontologie definiert (Bezeichner für) eine begriffliche Beschreibung einer Domäne:
  - maschinell interpretierbare Definitionen von grundlegenden Begriffen des Domänes und
  - Beziehungen zwischen diesen Konzepten
- **Ontologien sind meistens domänenspezifisch**
- Da sie auf Konzepten und nicht auf natürlichsprachlichen Namen basieren, sind Ontologien grundsätzlich **sprachunabhängig**.

## Ontologien in Computeranwendungen

- Ontologien werden seit Jahrzehnten intensiv benutzt bei:
  - der Wissensrepräsentation
  - dem Wissensengineering
  - Sprachverarbeitung (z.B. MT, MAT)
  - Information Retrieval und Information Extraction
  - Anderen Web-Anwendungen
- Im Semantic Web sind die Ontologien der Hauptmechanismus für die Darstellung von semantischen Beziehungen zwischen Ressourcen.

## Ontologie - Beispiel -

Ontologie in  
DBR-MAT  
MAT System



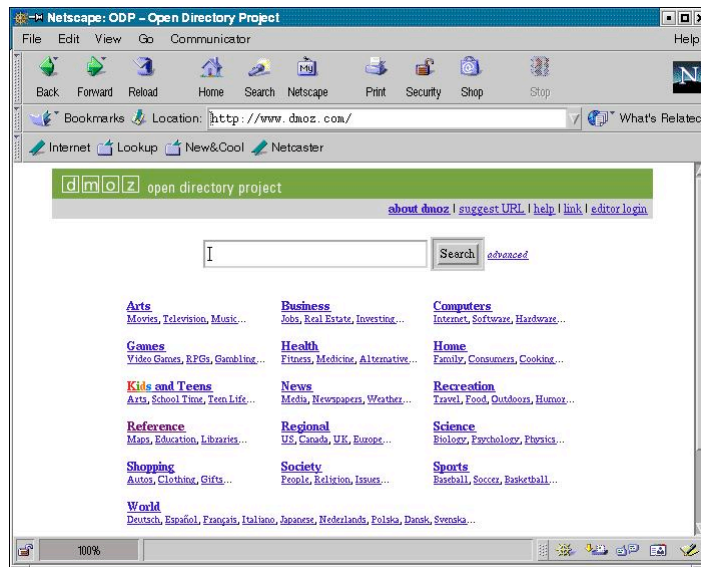
## Ontologietypen

- Die primitivste Form einer Ontologie ist ein Thesaurus, (Glossar) d.h, man spezifiziert nur, welche Begriffe speziell für eine Domäne sind.
- Eine typische Ontologie enthält eine hierarchische Organisation von Begriffen. Wenn man nur die Beziehung "Subclass-Of" ("Superclass-Of") repräsentiert, heisst die Ontologie **Taxonomie**
- Eine spezielle Form sind die **strukturierten Ontologien**, die auch auch Klassenmerkmale und Werteeigenschaften und -bereiche (häufig leider auch "Domänen" genannt) enthalten.

## Einfache Ontologien - Taxonomien

- Sind relativ einfach zu entwerfen
- Merkmale:
  - kontrollierter Wortschatz für die Begriffe eines dargestellten Gebiets
  - meistens benutzt zur Datenorganisation
  - liefern die allgemeine Struktur: die Konzepte können dannach instantiiert werden (z.B. "The Universal Standard Product and Services Classification)
  - Die in Web vorhandenen Taxonomien enthalten auch browsing support
- Viele Browsers bieten als Navigationsmöglichkeit das Browsing eine Taxonomie (z.B. DMOZ: directory Mozilla www.dmoz.com hat über 460,000 Klassen und 45,000 Editoren)

## DMOZ -Beispiel- 1

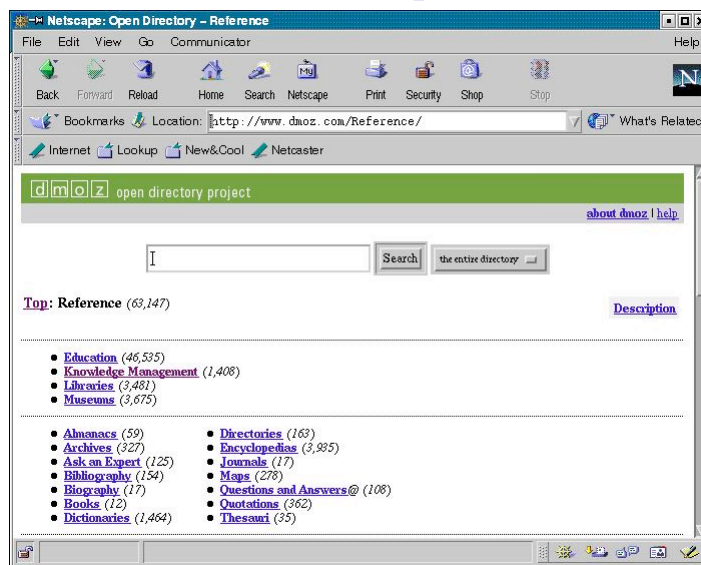


17.11.2003

WiSe 03/04

11

## DMOZ -Beispiel- 2

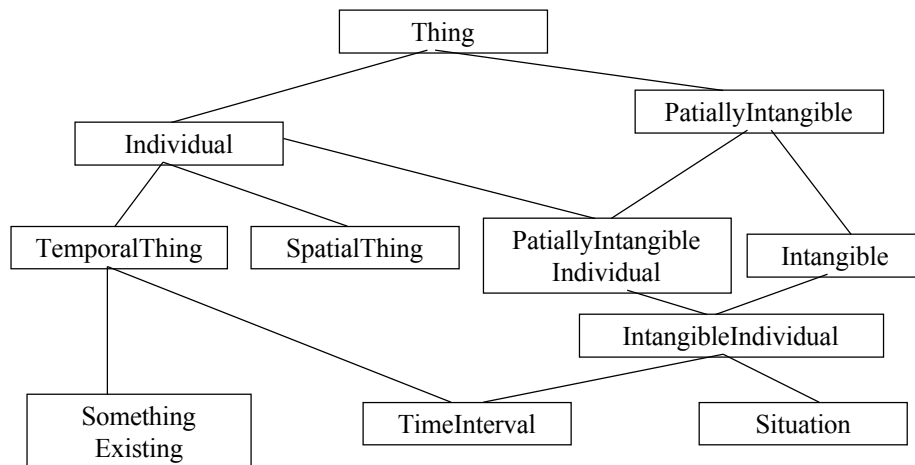


17.11.2003

WiSe 03/04

12

## Komplexere Ontologien



## Strukturierte Ontologien

- Die Klassen enthalten auch
  - Merkmale
  - Werte und gültige Domänen für diese Merkmale
- Vorteile:
  - Man kann sehr schnell einen Konsistenztest machen,
  - Man kann Inferenzregeln beschreiben,
  - Die Beziehungen zwischen Konzepten können detailliert beschrieben werden
  - Die Suche kann speziell oder allgemein sein (durch Beschränkung der Werte)

## Strukturierte Ontologien -Beispiel

The screenshot shows the Protégé software interface for editing an ontology. The main window is titled 'Reporter' and shows the class definition for 'Reporter' (type=STANDARD-CLASS). The left pane shows a class hierarchy with 'Reporter' selected. The right pane shows the class details, including documentation, role, and a table of template slots.


Name	Type	Cardinality	Other Facets
name	String	single	
date_hired	String	single	
salary	Float	single	
current_job_title	String	single	
phone_number	String	single	
other_information	String	single	

17.11.2003

WiSe 03/04

15

## Inhalt

- Was kann man mit RDF nicht ausdrücken ?
- Was sind Ontologien ?
- Wie entwirft man eine Ontologie? 
- Wie beschreibt man eine Ontologie im Semantic Web ?

17.11.2003

WiSe 03/04

16



## Hauptschritte beim Ontologieentwurf

- Definition von Ontologieklassen
- Entwurf einer Hierarchie (Taxonomie) zwischen Klassen (Super-/Subclass)
- Definition von Konzeptattributen /Merkmalen (slots) und deren Werten
- Eine Ontologie zusammen mit Klasseninstanzen = Wissensbasis.

## Hauptprinzipien des Ontologieentwurfs

- Es gibt mehrere Alternativen, eine Domäne zu modellieren (d.h es gibt nicht nur **eine** Ontologie). Sehr oft ist die Ontologie anwendungsorientiert und es gibt mehrere Ontologien über ein Gebiet je nach Anwendungssicht)
- Der Entwurf einer Ontologie ist ein iterativer Prozess.
- Die Bezeichner sollen plausibel für die logischen oder physische Objekte und Beziehungen der modellierten Domäne sein. Sehr oft benutzt man für eine Ontologie:
  - Nomen für die Objekte ("Drehbank")
  - Verbausdruck für die Beziehungen ("hat\_Eigenschaft")

## **Schritt 1- Domäne und Ziel der Ontologie**

- Man muss das Gebiet und die Anwendung, in der die Ontologie benutzt wird, klar identifizieren d.h:
  - Welche Domäne soll die Ontologie modellieren?
  - In welcher Anwendung wird die Ontologie benutzt,
  - Für welche Typen von Anfragen will man die Ontologie durchsuchen?
  - Wer wird die Ontologie weiter pflegen?

## **Schritt 2: Wiederbenutzbarkeit?**

- Es gibt schon viele Ontologien für die unterschiedlichsten Gebiete. Sehr oft sind sie anwendungsunabhängig kodiert (besonders solche, die für Webanwendungen entworfen wurden)
- Ontologie-Bibliotheken:
  - <http://www.ksl.stanford.edu/software/ontolingua>
  - <http://www.daml.org/ontologies>

### **Schritt 3 - Wichtige Konzepte**

- Man muss wichtige Konzepte im modellierten Gebiet identifizieren
- Diese Identifizierung steht immer in Zusammenhang mit der Anwendung, d.h. man muss sich überlegen, welche Konzepte sollten danach erklärt werden
- In unserem Fall z.B. muss man das Fragekorpus ansehen und festlegen, welche mögliche angefragte Konzepte sind.

### **Schritt 4 - Klassenhierarchie**

- Es gibt 2 Möglichkeiten, eine Klassenhierarchie zu definieren:
  - Bottom-up man beschreibt die sehr spezifischen Klassen (Blätter) und dann gruppiert man sie schrittweise in allgemeineren Konzepten
  - Top-down : man definiert erst die sehr allgemeinen Konzepte und dann verfeinert man sie.

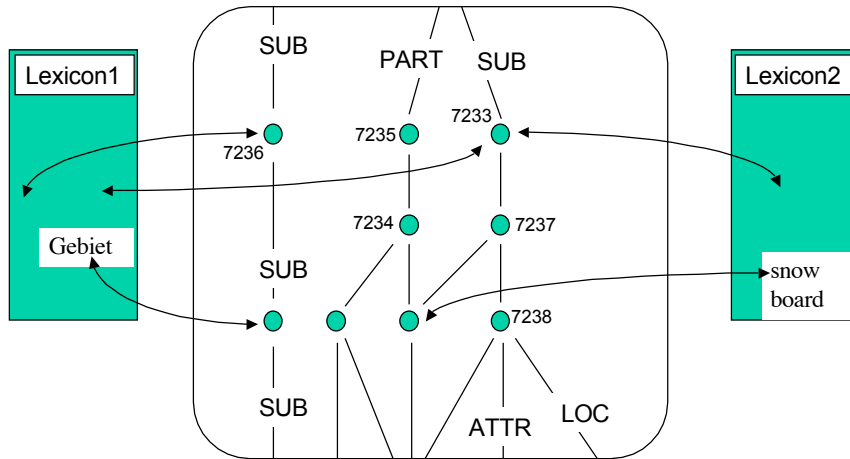
## **Schritt 5 - Entwurf einer strukturierter Ontologie**

- Man muss für jede Klasse Merkmale identifizieren und für jedes Merkmal
  - Die Kardinalität: wieviele mögliche Werte
  - Werttypen (string, number, boolean, enumeration)
  - Wertedomäne = die Klassen die dieses Merkmal haben
- Für Wertedomänen muss man Redundanz vermeiden:
  - Wenn in eine Wertedomäne eine Klasse und zugleich eine ihrer Unterklassen enthält, muss man die Unterklasse löschen
  - Wenn in einer Wertedomäne nur Unterklassen von Klasse X aber nicht die Klasse X selbst sind, muss man die Unterklassen durch die Klasse X ersetzen

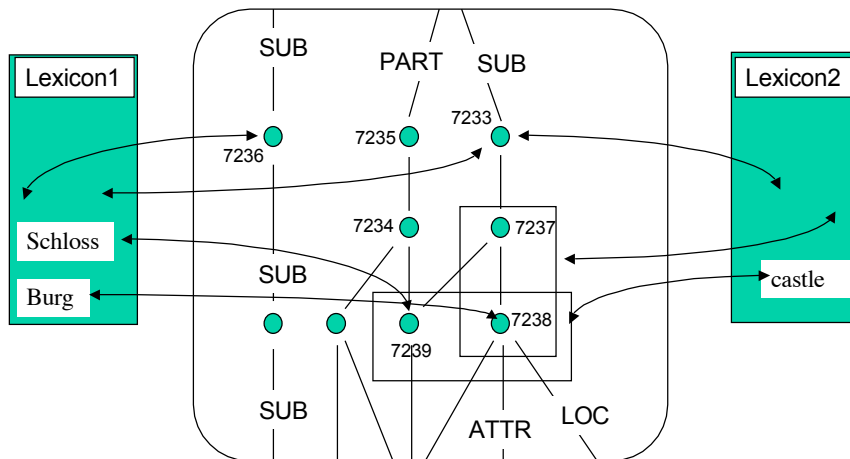
## **Schritt 6 Instantiierung**

- Für eine Klasseninstantiierung muss man:
  - Eine Klasse auswählen
  - Ein Individuum dieser Klasse spezifizieren
  - Die Attributwerte spezifizieren
- Man muss sich immer überlegen, welche Begriffe eines Gebietes Klassen sind und welche Begriffe Instanzen von Klassen

## Zweiseitige lexikalische Anbindung




## Mehrfache Vererbung für die unterschiedliche Konzeptualisierung



## Weitere Hinweise

- **Ontology Development: A guide to Creating Your First Ontology**  
[http://protege.stanford.edu/publications/ontology\\_development/ontology\\_101.html](http://protege.stanford.edu/publications/ontology_development/ontology_101.html)

## Inhalt

- Was kann man mit RDF nicht ausdrücken ?
- Was sind Ontologien ?
- Wie entwirft man eine Ontologie?
- Wie beschreibt man eine Ontologie im Semantic Web ? 

## Was ist RDFS ?

- RDFS = Resource Description Framework Schema
- Beschreibt genau, was wir brauchen: Die Konzepte und deren Bezeichner (Wortschatz, Vocabulary) für eine RDF Annotierung d.h. man erklärt mit RDFS-Ausdrücken, welche semantischen Beziehungen es zwischen den RDF-Tags gibt.
- Man kann damit Klassen und Merkmale beschreiben
- RDFS entspricht dem RDF-Modell, d.h jeder RDFS-Ausdruck ist ein Tripel (Subjekt, Prädikat, Objekt)

## Wie beschreibt man Klassen?

- Jede Klasse ist eine Ressource, die das Merkmal `rdf:type` und als Wert (Objekt) die Ressource `rdfs:Class` hat
- Z.B.  
Praktikum : Burg `rdf:type rdfs:Class`
- Entsprechend festgelegten Abkürzungsregeln kann man das "`rdf:type`" weglassen:

```
<rdfs:Class rdf:ID=„Burg“/>
```

## Wie beschreibt man Unterklassen ?

- Eine Unterklasse ist eine Ressource, die als Merkmal: `rdf:subClassOf` und als Objekt die Oberklasse hat
- Z.B.  
Praktikum: Kirchburg `rdfs:subClassOf praktikum:Burg`

```
<rdfs:class rdf:ID=„Kirchburg“>  
<rdfs:subClassOf rdf:resource="#Burg"/>  
</rdfs:Kirchburg>
```

## Wie instantiiert man Klassen

- Ein Individuum ist eine Ressource, die als Merkmal `rdf:type` hat und als Objekt eine definierte Klasse

z.B.

praktikum : Schässburg `rdf:type rdfs:Burg`

```
<praktikum:Burg rdf:ID=„ Schässburg“/>
```



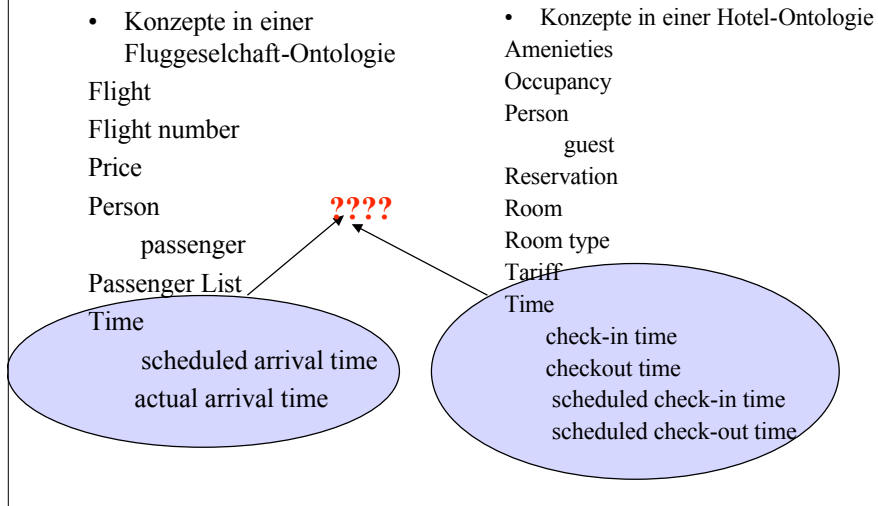
## Wie beschreibt man Klassenmerkmale

- Merkmale in RDF sind als Instanzen der Klasse `rdf:Property` beschrieben.
- zB.  
Praktikum:AltStadt `rdf:type` `rdfs:Class`  
Praktikum: deutscherName `rdf:type` `rdf:Property`  
Praktikum:deutscherName `rdfs:domain` praktikum:AltStadt  
`<rdf:Property rdf:ID=„deutscherName“>`  
`<rdfs:domain rdf:ressource =„#AltStadt“>`  
`</rdf:Property>`

## Ontologiebasierte Anwendungen

- Information Retrieval and Extraction
- Enterprise Integration: Beim Zusammenschluss von weltweiten aber auch regionalen Firmen treten viele Probleme auf, die sich durch unterschiedliche Unternehmenstruktur, verschiedene Arbeitsbereiche oder einfach durch unterschiedliche Sprachen ergeben. Eine Ontologie könnte hier helfen, die Probleme zu durchschauen und schneller zu lösen.
- Wissensmanagement
- Multilinguale (Web-)Anwendungen

## Merging 2 Ontologien



## Ontologien für Semantic Web

- To be useful for the Semantic Web, an ontology language must do more than just define a vocabulary and place constraints on the use of terms. It must be
  - Able to reference concepts defined elsewhere on the Web
  - Sharable over the Web
  - Able to work with one or more languages in use
  - Able to merge several ontologies
  - Widely accepted as a standard
  - Expressive enough for serious use
  - Able to support kinds of logical reasoning that are found to be needed to conduct the business of the Semantic Web

➔ OWL

©W3C