

Wintersemester 2006 • Institut für Germanistik I

Vorlesung Computerphilologie

Ontologien und Ontologie-Sprachen

Wie kann man Inhalte (von Webseiten) erschließen?

Was bringen Ontologien für die Analyse?

Aus der Semantic Web-Idee haben wir folgendes gelernt:

Im Web werden Millionen von Dokumenten von mehr als 300 Millionen Benutzern verwendet.



Je größer die Information im Web wird, desto schwerer wird die Organisation, Pflege und Wiederauffindung von Daten.



Semantic Web (nach Berners-Lee): verbesserter Informationszugang durch Referenz aus der Site auf standardisierte gemeinsame semantische Meta-Daten: Ontologien als Beziehungen zwischen Domain-Objekten

Ontologien sind die Schlüssel für das “Semantic Web”. Sie wurden schon seit langer Zeit als eine Form von Wissensrepräsentation benutzt.

Sprachen für das Semantic Web

- XML - oberflächenorientierte Syntax für strukturierte Dokumente. Die Sprache kann aber keine Information über die Bedeutung dieser Dokumente geben
- XML Schema - beschreibt syntaktische Regeln für XML-tags: XML-DTDs.
- RDF - ist ein Datenmodell für Objekte und Beziehungen zwischen konkreten Objekten
- RDFS - beschreibt Merkmale und Beziehungen zwischen abstrakten RDF-Objekte
- OWL ist eine vollständige spezifische Sprache zur Ontologie-Darstellung

„Layer-cake“ Architektur (nach Tim Berners-Lee)

Trust	
Proof	
Logic	
XML Familie	Ontologies & Owl
	RDF Schemas
	RDF, Topic Maps, ähnliche Technologien
	XML-Schemas, Namespaces
	XML-Dokumente
URI	Unicode

Wie stellt man Ontologien im Netz dar?

Warum nicht XML?

Grenzen von XML zur *Darstellung* von Ontologien:

- XML kann benutzt werden
 - für standardisierte Datendarstellung in Web
 - zur Syntaxdefinition andererer Markupssprachen (HTML)
 - zur Trennung von Form und Inhalt auf Webseiten
- Aber:
 - Man kann mit XML aber nicht die *Interpretation* (semantische Auswertung) von Texten beschreiben
 - Man kann mit XML nicht Ontologie spezifische Merkmale (class/subclass, Instanzen, Beziehungen zwischen Klassen, usw.) darstellen

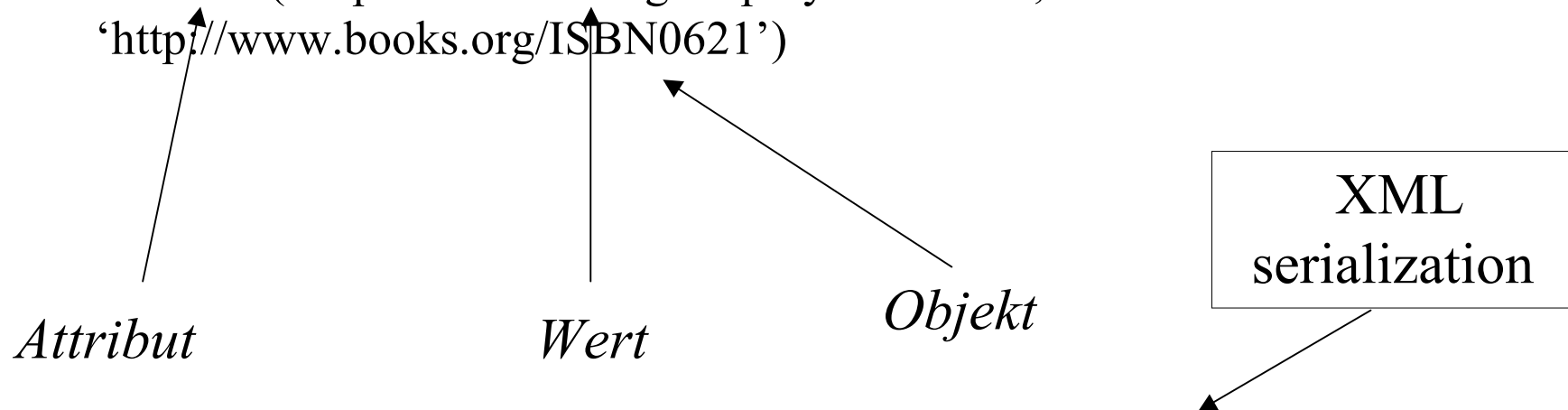
Die folgenden Folien folgen:

Davies, J., Fensel, D., van Harmelen, F.:

Towards the Semantic Web. West Sussex: Wiley 2003

Das RDF Model (Resource Description Framework)

- Ist eine W3C Empfehlung, die dazu dient, die Beschreibungen von Metadaten für Web-basierte Ressourcen zu standardisieren.
- Jeder Block in RDF ist ein Objekt-Attribut-Wert >(„Subjekt“, „Prädikat, „Objekt“) Tripel:
- authorOf(‘http://www.w3.org/employee/id1321’, ‘http://www.books.org/ISBN0621’)



```
<rdf:Description rdf:about:“http://www.w3.org/employee/id1321”>  
  <authorOf rdf: resource=“http://www.books.org/ISBN0621”/>  
</rdf:Description>
```

RDFS (RDF Schema)

- Beschreibt die grundlegenden Elemente, die für die Modellierung der Ontologien benutzt werden können.
- Man kann Klassen, Unterklassen, Merkmale und Untermerkmale definieren.
- RDFS gibt keine syntaktischen Bedingungen an.
- Es gibt Schlüsselwörter, wie: Class, subPropertyOf, subClassOf etc.

```
<rdfs: Class rdf: about="HardCover">
```

```
<rdfs: subClassOf rdf:ressource=#Book"/>
```

```
</rdfs: Class>
```

Grenzen von RDFS für Ontologiedarstellung

- RDFS kann nicht für komplizierte Ontologiedarstellungen benutzt werden:
- Man kann z.B. nicht beschreiben:
 - ein Buch hat nur einen Preis aber vielleicht mehrere Autoren
 - Datentypen für verschiedene Teile des Buchs (Titel, Preise usw.)
 - ein Buch darf nicht z.B. Hardcover und Softcover gleichzeitig sein können.

OWL-1-

- Veröffentlichung & Austausch von Ontologien
- Dedizierte Beschreibungs-Sprache
- Austausch-Syntax ist RDF/XML
- Elemente
 - Taxonomische Beziehungen zwischen Klassen
 - Eigenschaften und Datentypen
 - Objekteigenschaften (Eigenschaften von Individuen)
 - Instanzen von Klassen und Eigenschaften

OWL -2-

- Menge von OWL-Aussagen & Schlussfolgerungen = Wissensbasis (Knowledge base (KB))
- Typen von OWL:
 - Drei zunehmend komplexe Sprachen
 - OWL Lite
 - OWL DL
 - OWL Full

OWL-Header

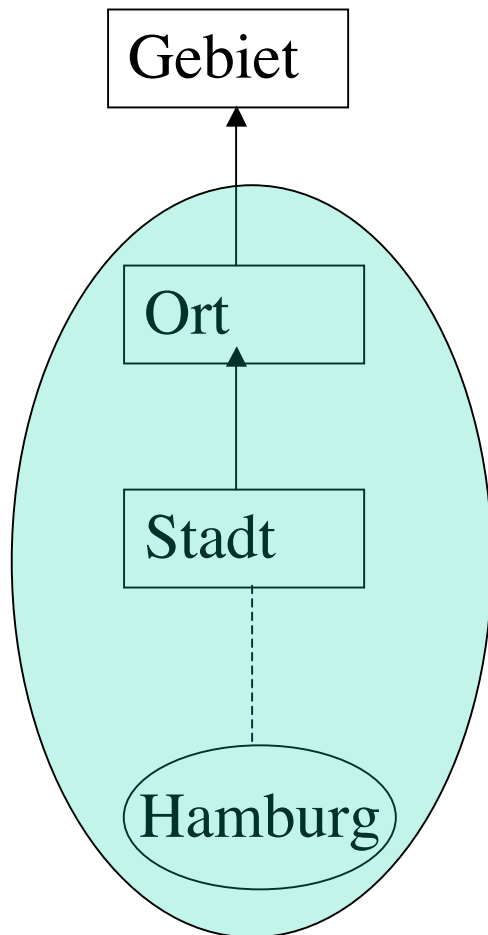
- Header
 - Namensräume (rdf, rdfs, owl, dc, ...)
 - Kommentare
 - Versionskontrolle (lite, ...)
 - Verbindungen anderer Ontologien

Wichtig für den Parser

OWL Basis-Definitionen -1-

- Bezeichnung mit `rdf:ID`
- Hierarchische Klassen
 - `Class`, `subClassOf` (`rdfs`)
`<owl:Class rdf:ID="Gebiet"> ...`
- Individuen (Elemente)
 - Implizit Elemente von `owl:Thing`
`<Gebiet rdf:ID=...>`
 - `Thing` (Zuordnung zu einer Klasse mit `<owl:Thing rdf:about=...>`)

Beispiel



```
<owl:Class rdf:Id="Ort"/>
```

```
<owl:Class rdf:ID="Stadt">
```

```
  <rdfs:subclassOf rdf:resource="#Ort"/>
```

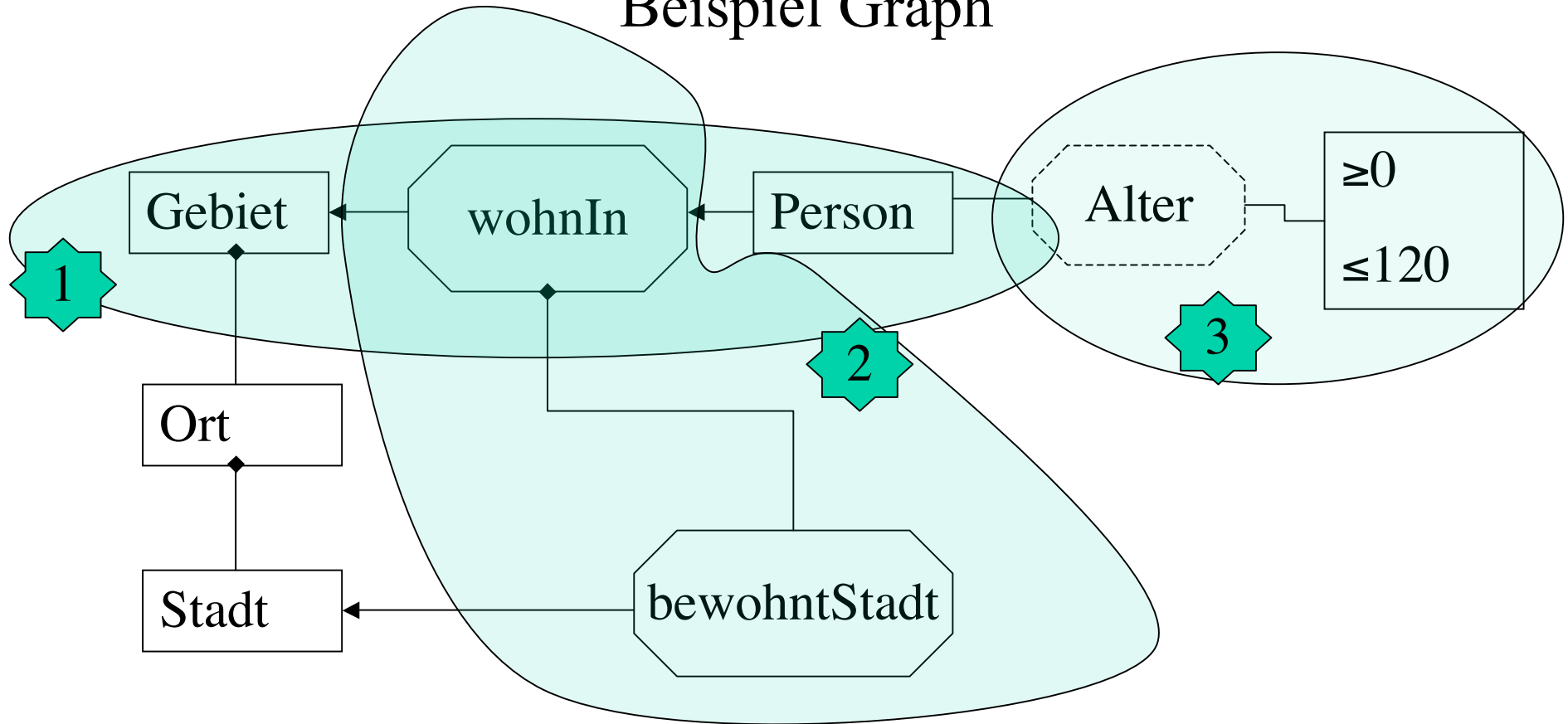
```
</owl:class>
```

```
<Stadt rdf:ID="Hamburg"/>
```

Basis - Definitionen

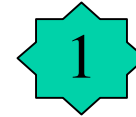
- Definieren von Eigenschaften
 - Objekteigenschaften
 - ObjectProperty (hierarchisch)
 - Domain (Definitionsbereich)
 - Range (Wertebereich, Objekte)
 - subPropertyOf (rdfs)
 - Rdf:resource (Einzuschränkende Objekteigenschaft)
 - Range
 - Datentypeigenschaften
 - DataTypeProperty
 - Domain
 - Range(Wertebereich, Strings oder einfache XML-Datentypen)

Beispiel Graph



OWL-Beispiel-Code

```
<owl:ObjectProperty rdf="wohntIn">  
  <rdfs:domain rdf:resource="#Person"/>  
  <rdfs:range rdf:resource="#Gebiet"/>  
</owl:ObjectProperty>
```



```
<owl:ObjectProperty rdf:ID="bewohntStadt">  
  <rdfs:subPropertyOf rdf:resource="#wohntIn"/>  
  <rdfs:range rdf:resource="Stadt"/>  
</owl:ObjectProperty>
```



```
<owl:DatatypeProperty rdf:ID="Alter">  
  <rdfs:domain rdf:resource="#Person"/>  
  <rdfs:range rdf:resource="&dt;Alter">  
</owl:DatatypeProperty>
```



Was haben wir daraus für die semantische Analyse in der CP gelernt?

Für viele Analyse-Aufgaben muss man auf Weltwissen oder Wissen aus einer bestimmten Domäne zurückgreifen.

- Statt dieses Wissen nur implizit zu verwenden
 - man schreibt einfach Penthesilea statt *Unsere Heldin*, oder
- das Wissen jedesmal explizit in die Tags zu schreiben
 - `<w ref = „Penthesilea“ >Unsere Heldin </w>`
- Kann man eine Ontologie der Personen und ihrer Funktionen aufstellen.

Ein anderes Beispiel

Man stelle sich ein getaggttes Corpus von Bühnenstücken vor.

Man hat dort alle Bezüge auf die Natur getaggt und möchte mit allen Referenzen auf Nicht-Natur vergleichen.

Spontane Idee:

Man stellt zusammen, was man unter Natur (die Natur-Tags, das Personal, Nennungen von Personenennamen, Personalpronomen etc.) und Nicht-Natur (Artefakte, Ideen, andere Pronomen etc.) versteht und sucht das gesamte Material vielfach danach durch.

Besseres Verfahren: Man erstellt eine Ontologie, die eine Hierarchie aus der „Domäne“ (der Welt) des Bühnenstückes enthält. Aus ihr kann man entnehmen, was als „Natur“ (z.B. slots in sub-classes von „Belebt“) unter welchen Bedingungen (filler constraints) gelten kann oder als „Nicht-Natur“. Diese Ontologie kann man dann auch für andere Werke benutzen