

# Ontologische Repräsentation von Daten (OWL)

Claudius Herder

Fachbereich Informatik  
Universität Hamburg

28. Mai 2006

# Ontologien allgemein

- 1 Was sind Ontologien?
  - Definition
  - Wofür Ontologien?
  - Was müssen Ontologien leisten?
  - Aufbau von Ontologien
  - Liste verschiedener Ontologie Sprachen

# OWL

- 2 Was ist OWL?
  - Woher kommt der Name OWL?
  - Wofür OWL?
- 3 OWL Full, OWL DL und OWL Lite
  - OWL Full
  - OWL DL
  - OWL Lite
  - Syntax
  - Datentypen
  - Versions Informationen
- 4 Zusammenfassung

## Teil I

# Ontologien

## Definition

- “an ontology is a specification of a conceptualization like a formal specification of a program” Tom Gruber (Stanford)
- Ontologien sind formal definierte Konzepte und Relationen
- Ontologien bilden üblicherweise Wissensbereiche ab. (knowledge domain)
- sie strukturieren Daten
- sie müssen nicht vollständig, aber konsistent sein
- Ontologien spezifizieren das Vokabular und die Eigenschaften einer Datensammlung

## Wofür Ontologien?

- Ontologien ermöglichen die maschinelle Erfassung der Bedeutung von Daten
- Verknüpfung von Informationen
- Erschließung von neuen Informationen

## Was müssen Ontologien leisten?

- definierte Syntax
  - notwendig um Informationen maschinell auszuwerten
- ausdrucksstark
  - alle benötigten Informationen müssen dargestellt werden können
- formale Semantik
  - um die Informationen präzise zu beschreiben
- logisch beweisbar/folgerbar (reasoning support)
  - erlaubt Schlussfolgerungen zu treffen
  - notwendig um Ontologien zu testen (z.B. auf Konsistenz)

# Aufbau von Ontologien

- Individuals
- Classes
- Properties
- Relations

## Individuals (Exemplare)

- Fundament einer Ontologie
  - konkrete Objekte (Personen, Tiere, Autos, Planeten, usw.)
  - abstrakte Objekte (Wörter, Zahlen)

## Classes (Klassen)

- Gruppen, Mengen, Aufzählungen von Exemplaren
  - Person, die Klasse aller Personen
  - Zahlen, die Klasse aller Zahlen
  - Klasse, die Klasse aller Klassen

## Intensionale und Extensionale Klassen

- extensionale Klassen
  - zwei Klassen mit dem gleichen Inhalt sind identisch
  - mathematisch leichter zu behandeln
    - Lebewesen mit einem Herz, Lebewesen mit einer Niere (eine Klasse)
- intensionale Klassen
  - zwei Klassen mit dem gleichen Inhalt sind **nicht** identisch
  - ermöglichen feinere Unterschiede
    - Lebewesen mit einem Herz (eine Klasse) ,  
und Lebewesen mit einer Niere (eine andere Klasse)

# Partitionen

- Klassen können zu Partitionen zusammengefasst werden
  - die Regeln der Partition bestimmen z.B.
    - welches Exemplar zu welcher Klasse gehört
    - ob Klassen disjunkt sind
    - etc.

## Properties (Eigenschaften)

- bestimmen die Eigenschaften eines Exemplars
  - Name: Pinguin
  - Grösse: 30-120 cm
  - Gewicht: 1-40 kg
- Attribute können komplexe Datentypen sein
  - Listen, Klassen, etc.

## Relations (Beziehungen)

- sind Attribute eines Exemplars
- beschreiben die Semantik einer Domäne
- legen den “Baum der Ontologie” fest

## Liste verschiedener Ontologie Sprachen

- Traditionelle Ontologie Sprachen:
  - KIF
  - OCML
  - LOOM
  - CycL
  - KM programming language
  - F-Logic
  - OKBC (Open Knowledge Base Connectivity)

# Liste verschiedener Ontologie Sprachen

- Markup Ontologien Sprachen
  - SHOE
  - XOL
  - DAML+OIL (Vorgänger von OWL)
  - RDF
  - RDFS
  - OW

## Teil II

# OWL

## Woher kommt der Name OWL?

- Web Ontology Language
- The owl (Winnie the Pooh)
- Eulen verbindet man mit Weisheit und Ehre
- “Why not be inconsistent in at least one aspect of a language which is all about consistency”  
Guus Schreiber

## Wofür OWL?

- OWL ist ein Teil der Empfehlungen des W3C zum Semantik Web.
  - Die Grundbausteine des Semantik Web sind:
    - XML bietet die Syntax um Informationen zu strukturieren
    - XMLS legt die Struktur der Dokumente fest
    - RDF stellt das Datenmodell zur Verfügung um Objekte und die Beziehungen zwischen ihnen auszudrücken
    - RDFS definiert die Semantik und das Vokabular zur Beschreibung von Eigenschaften und Klassen
    - OWL erweitert das Vokabular von RDFS, ermöglicht Beziehungen zwischen Klassen
  - OWL wurde entwickelt um die maschinelle Suche, Bearbeitung und Zusammenfassung von Informationen zu ermöglichen bzw. zu erleichtern

# OWL ist eine Weiterentwicklung von RDFS

- Erweiterungen von OWL:
  - Local scope ermöglicht es einzelnen Klassen bestimmte Eigenschaften zuzuweisen
    - z.B. Kühe essen Gras, während andere Tiere auch Fleisch essen.
  - Disjunkte Klassen
    - z.B. Vögel und Reptilien, in RDF sind beide Klassen Subklassen von Tiere
  - Boolesche Kombinationen
    - z.B. die Klasse Person ist die disjunkte Vereinigung von Männern und Frauen
  - Kardinalitäts Einschränkungen
    - z.B. jeder Mensch hat genau einen Vater
  - Besondere Eigenschaften
    - z.B. Transitivität, einzigartig, Gegenteil von, usw.

## OWL Full

- die ausdrucksstärkste Form von OWL
- syntaktische Freiheiten von RDF
- Erweiterung von RDFS
- erlaubt Klassen als Exemplare, Erweiterung des RDF Vokabulars
- zu komplex um vollständig und effizient logisch bewiesen zu werden
- Jedes gültige RDF Dokument ist normalerweise auch ein OWL Full Dokument, es sei denn, es wurde extra in OWL DL oder Lite verfaßt.

## OWL DL - Description Logic

- Teilmenge von OWL Full, welche “Beschreibungslogiken” (reasoning) erfüllt
- ermöglicht die Erschließung neuen Wissens durch prädikatenlogische Ausdrücke
- ist formal beweisbar
- jedes OWL DL Dokument ist ein RDF Dokument, allerdings nicht umgekehrt

# OWL Lite

- ermöglicht einen schnellen unkomplizierten Einstieg
- einfach zu implementieren, d.h. viele Programme können OWL Lite Dokumente erzeugen
- für die meisten Domänen vollkommen ausreichend

# Syntax

- Normalerweise XML
- abstrakte Syntak
- grafische Syntak (UML)

# Header

- Die Wurzel (root) des hierarchischen Baumes

## Header

```
<rdf:RDF
  xmlns:owl=
    " http://www.w3.org/2002/07/owl#"
  xmlns:rdf=
    " http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs=
    " http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd=
    " http://www.w3.org/2001/XMLSchema#">
```

## Einfache Ontologie Definition

- Ontologien können andere Ontologien importieren
- die Version der Ontologie wird angegeben

### Einfache Ontologie Definition

```
<owl:Ontology rdf:about="">  
<rdfs:comment>An example OWL ontology </rdfs:comment>  
<owl:priorVersion  
  rdf:resource=" http://www.mydomain.org/uni-ns-old" />  
<owl:imports  
  rdf:resource=" http://www.mydomain.org/persons" />  
<rdfs:label>University Ontology</rdfs:label>  
</owl:Ontology>
```

# Classes

- Klassen werden mit owl:Class erstellt

## Einfache Klasse associateProfessor

```
<owl:Class rdf:about='#associateProfessor'>  
<owl:disjointWith rdf:resource='#professor' />  
<owl:disjointWith  
    rdf:resource='#assistantProfessor' />  
<\owl:Class>
```

# Thing and Nothing

- owl:Thing ist die allgemeine Klasse, die alle Klassen enthält
- owl:Nothing ist die leere Klasse

## Object Properties

- Objekte die als Eigenschaft Objekte haben

### Object Properties

```
<owl:ObjectProperty rdf:ID="isTaughtBy">  
<owl:domain rdf:resource="\#course" />  
<owl:range rdf:resource="\#academicStaffMember" />  
<rdfs:subPropertyOf rdf:resource="\#involves" />  
</owl:ObjectProperty>
```

## Data Type Properties

- Objekte die Datentypen als Eigenschaft haben

### Data Type Properties

```
<owl:DatatypeProperty rdf:ID="age">  
<rdfs:range  
  rdf:resource=  
  "http://www.w3.org/2001/XMLSchema#nonNegativeInteger" />  
</owl:DatatypeProperty>
```

## Equivalent Properties

- Eigenschaften mit gleicher Bedeutung

### Equivalent Properties

```
<owl:ObjectProperty rdf:ID="lecturesIn">  
<owl:equivalentProperty rdf:resource="#teaches" />  
</owl:ObjectProperty>
```

## Inverse Property

- Eigenschaften, die das Gegenteil von anderen Eigenschaften sind

### Inverse Property

```
<owl:ObjectProperty rdf:ID="teaches">  
<rdfs:range rdf:resource="#course" />  
<rdfs:domain rdf:resource="#academicStaffMember" />  
<owl:inverseOf rdf:resource="#isTaughtBy" />  
</owl:ObjectProperty>
```

## Property Restriction

- owl:allValuesFrom Alle Exemplare haben diese Eigenschaft

### allValuesFrom

```
<owl:Class rdf:about="#firstYearCourse">  
<rdfs:subClassOf>  
<owl:Restriction>  
<owl:onProperty rdf:resource="#isTaughtBy" />  
<owl:allValuesFrom rdf:resource="#Professor" />  
</owl:Restriction>  
</rdfs:subClassOf>  
</owl:Class>
```

## Property Restriction

- owl:hasValue Exemplar hat diese Eigenschaft

### hasValue

```
<owl:Class rdf:about="#mathCourse">  
  <rdfs:subClassOf>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="#isTaughtBy" />  
      <owl:hasValue rdf:resource="#949352" />  
    </owl:Restriction>  
  </rdfs:subClassOf>  
</owl:Class>
```

## Property Restriction

- owl:someValuesFrom Exemplar hat diese Eigenschaften

### someValuesFrom

```
<rdfs:subClassOf>  
<owl:Restriction>  
<owl:onProperty rdf:resource="#teaches" />  
<owl:someValuesFrom rdf:resource="#undergraduateCourse" />  
</owl:Restriction>  
</rdfs:subClassOf>  
</owl:Class>
```

## Cardinality Restrictions

- Einschränkungen der Anzahl

### Cardinality Restrictions

```
<owl:Class rdf:about="#course">  
<rdfs:subClassOf>  
<owl:Restriction>  
<owl:onProperty rdf:resource="#isTaughtBy" />  
<owl:minCardinality  
    rdf:datatype="&xsd; nonNegativeInteger">1  
</owl:minCardinality>  
</owl:Restriction>  
</rdfs:subClassOf>  
</owl:Class>
```

## Special Properties

- owl:TransitiveProperty Eigenschaft ist transitiv
  - “hat eine bessere Note als”, “ist Nachfolger von”
- owl:SymmetricProperty Eigenschaft ist symmetrisch
  - “hat die gleiche Note wie”, “ist verwandt mit”
- owl:FunctionalProperty mindestens ein Wert für jede Eigenschaft
  - “Alter”, “Größe”, “Vorgesetzter”
- owl:InverseFunctionalProperty zwei Exemplare können nicht den gleichen Wert haben

## Special Properties

```
<owl:ObjectProperty rdf:ID="hasSameGradeAs">  
<rdf:type rdf:resource="&owl;TransitiveProperty" />  
<rdf:type rdf:resource="&owl;SymmetricProperty" />  
<rdfs:domain rdf:resource="#student" />  
<rdfs:range rdf:resource="#student" />  
</owl:ObjectProperty>
```

## Boolean Combinations

- Klassen können logisch kombiniert werden
  - z.B. Durchschnitt, Vereinigung, Komplement

### Boolean Combinations

```
<owl:Class rdf:about="#course">  
<rdfs:subClassOf>  
<owl:Restriction>  
<owl:complementOf rdf:resource="#staffMember" />  
</owl:Restriction>  
</rdfs:subClassOf>  
</owl:Class>
```

# Enumerations

- Aufzählungen

## Enumerations

```
<owl:oneOf rdf:parseType="Collection">  
<owl:Thing rdf:about="#Monday" />  
<owl:Thing rdf:about="#Tuesday" />  
<owl:Thing rdf:about="#Wednesday" />  
<owl:Thing rdf:about="#Thursday" />  
<owl:Thing rdf:about="#Friday" />  
<owl:Thing rdf:about="#Saturday" />  
<owl:Thing rdf:about="#Sunday" />  
</owl:oneOf>
```

# Instances

- Exemplare von Klassen

## Instances

```
<rdf:Description rdf:ID=" 949352">  
<rdf:type rdf:resource="#academicStaffMember" />  
</rdf:Description>  
<academicStaffMember rdf:ID=" 949352">  
<uni:age rdf:datatype="&xsd;integer">39<uni:age>  
</academicStaffMember>
```

# Datentypen

- **keine** konstruierten Datentypen
- es werden auch nicht alle Datentypen von XMLS unterstützt

## Versions Informationen

- owl:priorVersion
  - die Vorgänger Version
- owl:versionInfo
  - aktuelle Version
- owl:backwardCompatibleWith
  - abwärts kompatible Version
- owl:incompatibleWith
  - inkompatible Version

# Zusammenfassung

- OWL ist der vom W3C vorgeschlagene Standard für Ontologien
- OWL baut auf RDF und RDF Schema auf
  - (XML-basierte) RDF Syntax wird verwendet
  - Exemplare werden mit RDF definiert
- Korrektheit ist formal beweisbar

# Quellen

- <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>
- [http://de.wikipedia.org/wiki/Ontologie\\_%28Informatik%29](http://de.wikipedia.org/wiki/Ontologie_%28Informatik%29)
- <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>
- <http://www.gi-ev.de/service/informatiklexikon/informatiklexikon-detailansicht/meldung/57/>
- [http://en.wikipedia.org/wiki/Ontology\\_computer\\_science](http://en.wikipedia.org/wiki/Ontology_computer_science)
- [http://en.wikipedia.org/wiki/Web\\_Ontology\\_Language](http://en.wikipedia.org/wiki/Web_Ontology_Language)
- <http://www.w3.org/TR/owl-ref/#Sublanguage-def>
- <http://www.w3.org/2004/OWL/>
- <http://www.w3.org/TR/owl-features/>
- <http://en.wikipedia.org/wiki/Reasoning>
- A Semantic Web Primer Grigoris Antoniou and Frank van Harmelen (Chapter 4 Web Ontology Language)
- <http://www.ics.forth.gr/isl/swprimer/presentations/Chapter4.ppt>