

Ontologien

RDF Schema (RDF) und die Web Ontology Language (OWL)

Albert Weichselbraun

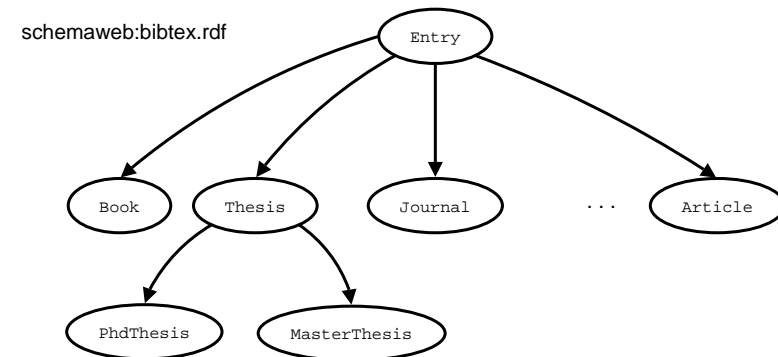
Ontologien

- RDFS und OWL erlauben es RDF Vokabular zu spezifizieren.
- Aussagen über Klassen (!=Individuen)
- Zugehörigkeit eines Individuums zu einer Klasse wird über das `rdf:type` Attribut spezifiziert
- Möglichkeit Hierarchien aufzubauen
 - Klassen (Subjekt, Objekt): `rdfs:subClassOf`
 - Eigenschaften (Prädikat): `rdfs:subPropertyOf`

Ontologien

- RDF-Vokabular, definiert Beziehungen zwischen
 1. Klassen,
 2. Eigenschaften und
 3. deren Assoziation mit Klassen.→ liefert Informationen über die *Semantik*
- *Vorteil*: ermöglicht Deduktion(!) und erweiterte Suchen

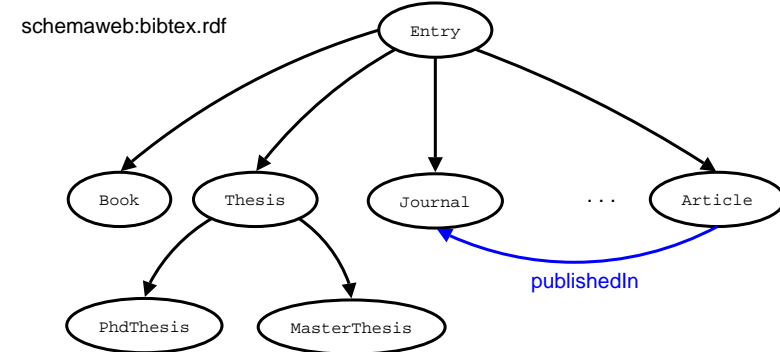
Klassenhierarchien



Klassenhierarchien (Syntax)

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/.../22-rdf-syntax-ns#"
4   xmlns:rdfs="http://www.w3.org/.../rdf-schema#"
5   xml:base="http://www.ai.wu.ac.at/.../bibTeX#"
6
7   <rdfs:Class rdf:ID="Thesis">
8     <rdfs:subClassOf rdf:resource="#Entry"/>
9   </rdfs:Class>
10  <rdfs:Class rdf:ID="MasterThesis">
11    <rdfs:subClassOf rdf:resource="#Thesis"/>
12  </rdfs:Class>
13  ...
14 </rdf:RDF>
```

Eigenschaften



Hierarchien von Eigenschaften

- Analog zu Klassenhierarchien

```
1 @prefix rdf: <http://.../22-rdf-syntax-ns#>.
2 @prefix rdfs: <http://.../rdf-schema#>.
3 @prefix aw: <http://bsp.at/.../bibTeX#>.
4
5 aw:publishedIn rdf:type rdf:Property.
6 aw:inJournal rdfs:subPropertyOf
  aw:publishedIn.
7 aw:inProc rdfs:subPropertyOf
  aw:publishedIn.
```

RDFS - Semantics

rdfs:subClassOf

- keine Angabe: Unterklasse von rdfs:Resource
- $A \sqsubseteq A$
- multiple Angaben: logisch UND verknüpft (intersection)
- *transitive* $A \sqsubseteq B \sqsubseteq C \Rightarrow A \sqsubseteq C$

analoges gilt für rdfs:subPropertyOf.

RDFS - Semantics

Gleichheit von Klassen/Properties:

```
1 @prefix   rdfs:   <http://.../rdf-schema#>.
2 @prefix   b2b:   <http://bsp.at/.../b2b#>.
3 @prefix   as:    <http://amazon.de/.../art#>.

5 b2b:bruttoPreis rdfs:subPropertyOf as:price.
6 as:price       rdfs:subPropertyOf b2b:bruttoPreis.

8 b2b:item       rdfs:subClassOf  as:article.
9 as:article     rdfs:subClassOf  b2b:item.
```

Range, Domain

Ohne Angabe von Range und Domain ist die Zuordnung von ungültigen Werten möglich.

```
1 @prefix   rdf:    <http://.../22-rdf-syntax-ns#>.
2 @prefix   rdfs:   <http://.../rdf-schema#>.
3 @prefix   aw:     <http://bsp.at/.../bibTeX#>.
4 @prefix   p:      <http://bsp.at/.../people#>.

6 p:Anna                                p:age           "-22".
7 <http://x.at/article> aw:inJournal   aw:Person.
```

Range, Domain - Syntax

Definition von Einschränkungen für die Verwendung von Ressourcen:

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/.../22-rdf-syntax-ns#"
4   xmlns:rdfs="http://www.w3.org/.../rdf-schema#"
5   xml:base="http://.../2005/bibTeX#">

7   <rdf:Property rdf:ID="publishedIn">
8     <rdfs:domain rdf:resource="#Article" />
9     <rdfs:range  rdf:resource="#Journal" />
10  </rdf:Property>
11  ...
12 </rdf:RDF>
```

Range, Domain - Syntax

Definition von Einschränkungen für die Verwendung von Literalen:

```
1 @prefix   rdfs:   <http://.../rdf-schema#>.
2 @prefix   p:      <http://bsp.at/.../people#>.

4 p:age     rdfs:domain p:Person;
5           rdfs:range  xsd:nonNegativeInteger.
```

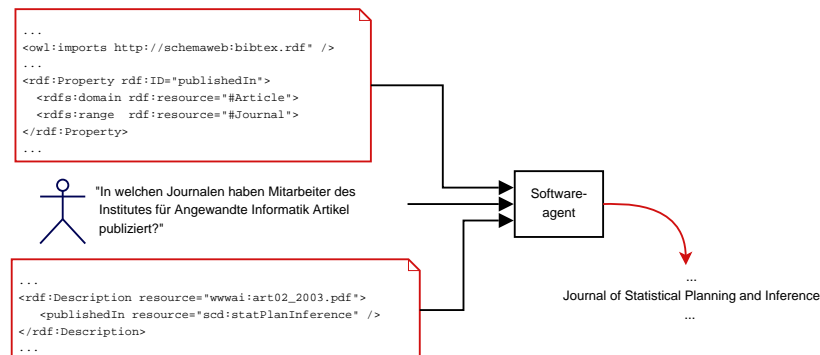
Range, Domain - Semantik

- `rdf:Property` ist im RDF-Namespaces(!)
- `rdfs:domain`: welche Klassen Eigenschaften annehmen können
- `rdfs:range`: welche Werte (Instanzen) Eigenschaften annehmen können
- 0..x `rdfs:range`, `rdf:domain` Statements
- Eigenschaften werden getrennt von Klassen definiert!
→ Vorteil: jederzeit neue Eigenschaft für Klasse x definierbar

Range, Domain - Eigenschaften

- Eigenschaften werden *vererbt*
- Definition einer Eigenschaft für zwei Klassen (*OR* nicht *AND*)
⇒ für die Superklasse definieren
- Hierarchien von Eigenschaften: `rdfs:subPropertyOf`

OWL - Deduktion



OWL/RDFS - Eigenschaftshierarchien - Beispiel

Eigenschaftshierarchie:

```
family:Parent rdfs:subPropertyOf family:Father
```

Datenquelle:

```
1 <Person rdf:ID="Anna"  
2   xmlns:rdf="http://www.w3.org/.../22-rdf-syntax-ns#"  
3   xmlns:family="http://.../onto/family#">  
4   <family:Father>  
5     <Person rdf:ID="Peter" />  
6   </family:Father>  
7 </Person>
```

“Anna’s Vater ist Peter” → Vater ist eine Untereigenschaft von *Elternteil* → Peter ist ein *Elternteil* von Anna

OWL/RDFS - Motivation

- RDF Schemata: begrenzte Ausdruckskraft; Konstrukte um
 - Kardinalitäten (Kind hat *einen* Vater),
 - erweiterte Eigenschaften von Properties (symmetrisch, funktional abhängig, ...)abzubilden fehlen.
- OWL → umfangreicher
⇒ ↑ Ausdruckskraft jedoch komplexer (Beispiel: *Mitfreude*)
- Drei Ausprägungen: OWL Lite, OWL DL, OWL Full
- OWL Vokabular wird meist gemeinsam mit RDFS verwendet

OWL - owl:sameIndividualAs

- owl:sameIndividualAs → zwei Ressourcen sind ident!
- Beispiel: The Robber and the Speeder (xfront.com)

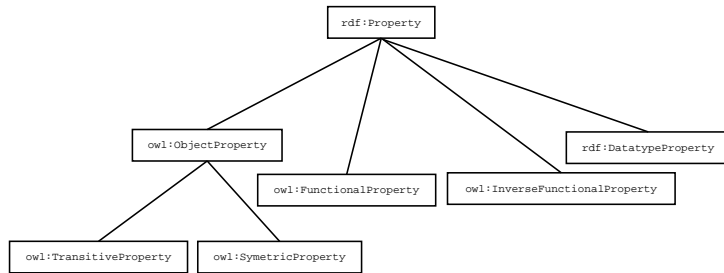
OWL - Definition von Eigenschaften

- Beziehungen zwischen Eigenschaften
 - owl:equivalentProperty
 - owl:inverseOf
- Beschränkungen der Kardinalitäten
 - owl:SymmetricProperty
 - owl:TransitiveProperty
- Logische Charakteristika von Eigenschaften:
 - owl:FunctionalProperty (Beispiel)
 - owl:InverseFunctionalProperty

OWL - Eigenschaften

- zwei Klassen von Eigenschaften
 - owl:ObjectProperty
rdfs:Resource → rdfs:Resource
 - owl:DatatypeProperty
rdfs:Resource → {XML Datentyp|rdfs:Literal}
 - owl:(Inverse)FunctionalProperty
rdfs:Resource → {rdfs:Resource|XML Datentyp|rdfs:Literal}

OWL - Eigenschaften - Klassenhierarchie



⇒ `owl:SymmetricProperty` nur für Ressourcen anwendbar!

⇒ `owl:inverseOf`, `owl:equivalentProperty` Eigenschaften ≠ Klassen

OWL - Eigenschaften - Syntax

```
1 <rdf:Property rdf:ID="publishedIn">
2   <rdfs:domain rdf:resource="#Article" />
3   <rdfs:range rdf:resource="#Journal" />
4 </rdf:Property>
```

↕

```
1 <owl:ObjectProperty rdf:ID="publishedIn">
2   <rdfs:domain rdf:resource="#Article" />
3   <rdfs:range rdf:resource="#Journal" />
4 </owl:ObjectProperty>
```

OWL - Eigenschaften - Beispiel

```
1 <owl:DatatypeProperty rdf:ID="matrikelNummer">
2   <rdf:type
3     rdf:resource = "http://...#FunctionalProperty" />
4   <rdfs:domain rdf:resource="#Student" />
5   <rdfs:range
6     rdf:resource="http://...XMLSchema#nonNegativeInteg
7 </owl:DatatypeProperty>
```

OWL - `rdfs:Class` ↔ `owl:Class`

- `owl:Class` = Unterklasse von `rdfs:Class`
- Grund: in OWL DL und OWL Lite gelten zusätzliche Restriktionen für Klassen
Beispiel: keine Metaklassen möglich (Klassen von Klassen)
→ in OWL DL und OWL Lite Ontologien nur `owl:Class` verwendbar
- OWL Full → `owl:Class` ≡ `rdfs:Class`

OWL - Anonyme Klassen

```
1 <owl:Class rdf:ID="IAESTE_Vienna_Vorstand">
2   <rdf:subClassOf rdf:resource="#IAESTE_Vienna" />
3   <owl:Restriction>
4     <owl:onProperty rdf:resource="#memberOf" />
5     <owl:allValuesFrom
6       rdf:resource="#VorstandsMember" />
7   </owl:Restriction>
8 </owl:Class>
```

Schnittmenge: "alle Vorstandsmitglieder in .at" \cap "IAESTE-Vienna Mitglieder"

OWL - Anonyme Klassen

```
1 <owl:Class rdf:ID="Wine">
2   <rdfs:subClassOf rdf:resource="&food;PotableLiquid" />
3   <rdfs:subClassOf>
4     <owl:Restriction>
5       <owl:onProperty rdf:resource="#hasMaker" />
6       <owl:someValuesFrom rdf:resource="#Winery" />
7     </owl:Restriction>
8   </rdfs:subClassOf>
9   ...
10 </owl:Class>
```

Source: *Web Ontology Language Guide 1.0*

OWL - Set Operatoren

- erlauben die Definition von Klassen
- verfügbare Operatoren:
 - intersectionOf
 - unionOf
 - complementOf

OWL - Set Operatoren - Syntax

```
1 <owl:Class rdf:ID="Frau">
2   <owl:Class rdf:ID="Mensch">
3     <owl:Class>
4       <owl:complementOf rdf:resource="#Mann" />
5     </owl:Class>
6   </owl:Class>
7 </owl:Class>
```

OWL - Klassendefinitionen

- Definition von Klassen via:
 1. Aufzählung der Instanzen (`owl:oneOf`)
 2. Spezifikation einer äquivalenten Klasse (`owl:equivalentClass`)
 3. Angabe einer disjunkten Klasse (`owl:disjointWith`)

- Klassendefinitionen- Syntax

```
1 <owl:Class rdf:ID="Naturschutzgebiete">
2   <rdfs:subClassOf rdf:resource="#Gebiete" />
3   <owl:oneOf rdf:parseType="Collection">
4     <geo:Nationalpark
5       rdf:about="http://www.hohetauern.at/" />
6     <geo:Nationalpark
7       rdf:about="http://www.donauauen.at/" />
8     ...
9   </owl:oneOf>
10 </owl:Class>
```

OWL - Klassen - Andere Konstrukte

- `owl:Thing`, `owl:Nothing`
- `owl:cardinality`
→ Anzahl an Attributen (z.B. *eine* Maximale Tiefe für einen Ozean)
- `owl:minCardinality`, `owl:maxCardinality`
- `owl:equivalentProperty`

OWL - Individuen

- folgende Aussagen möglich
 - `owl:sameIndividualAs`
 - `owl:differentFrom`

Probleme

- Kollision von Ontologien - Beheben von Widersprüchen
 - unterschiedliche Metadaten zu einer Ressource → wem glaubt man?
 - RDF/OWL-Poisoning
- Technologieebene:
 - URI/URL noch nicht Unicode fähig
 - endliche Bearbeitungszeit
 - skaliert diese Technologie?

Acknowledgment

- Nicola Guarino, Institute for Cognitive Science and Technologies, Trento-Roma, Italy
- Roger L. Costello, David B. Jacobs, xfront.com - OWL/RDF/RDFS Intros
- Sean Bechhofer, Ian Horrocks and Peter F. Patel-Schneider: Tutorial on OWL