

Semantic Web

Sebastian Iwanowski
FH Wedel

Kap. 3:
RDF, RDFS, OWL: Eine anwendungsorientierte Einführung

Namensräume

Eindeutigkeit der Begriffe über URIs

- Referenzen sind sehr zeichenfüllend und unübersichtlich
- Das gilt vor allem für Relationen zwischen Begriffen

Lösung: Namensräume

`prefix xsd:, namespace URI: http://www.w3.org/2001/XMLSchema#`

`prefix extterms:, namespace URI: http://www.example.org/terms/ (for terms used by an example organization),`

`prefix exstaff:, namespace URI: http://www.example.org/staffid/ (for the example organization's staff`

xsd:integer entspricht: <http://www.w3.org/2001/XMLSchema#integer>

RDF

Entity-Relations: Subjekt – Prädikat – Objekt

- Subjekt und Objekt sind beliebige Entities
- Prädikat ist eine Relation zwischen zwei Entities

Bsp.:

exstaff:85740 exterms: address “1501 Grant Avenue, Bedford, Massachusetts 01730

RDF

Leerknoten

- Relationen in RDF sind per Definition binär
- Leerknoten sind identitätslose Entities zur Realisierung höherwertigerer Relationen

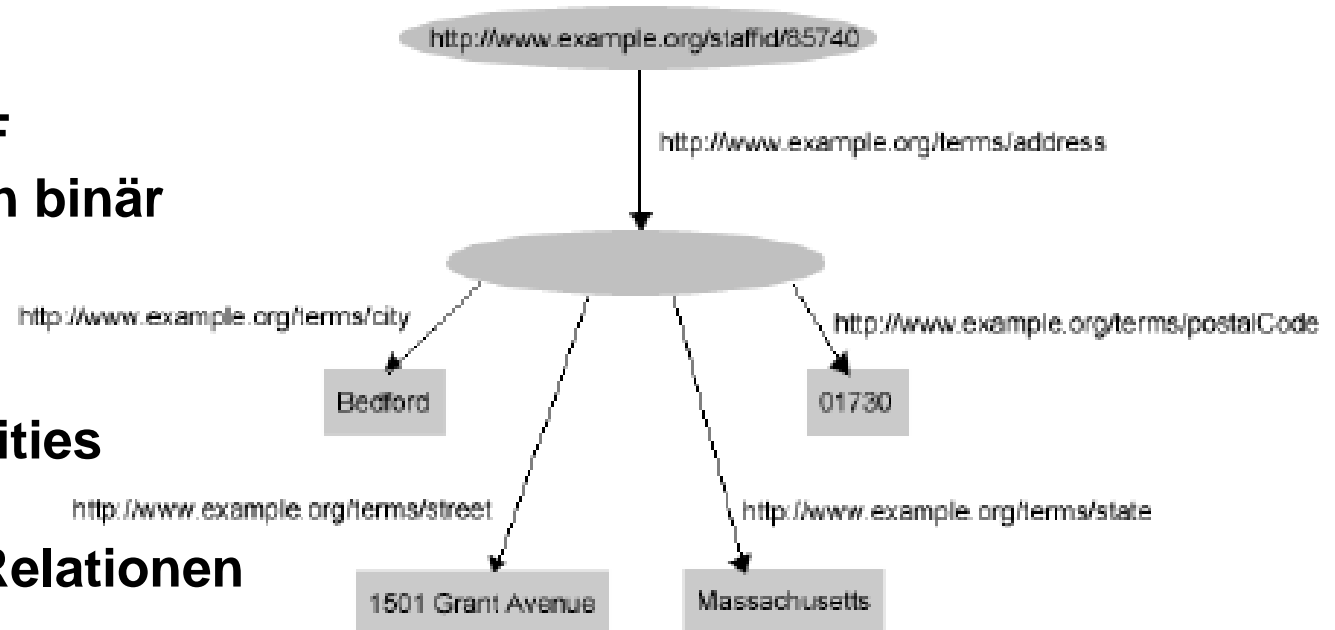


Figure 6: Using a Blank Node

```
exstaff:85740      exterms:address      _:johnaddress .
_:johnaddress     exterms:street       "1501 Grant Avenue" .
_:johnaddress     exterms:city         "Bedford" .
_:johnaddress     exterms:state        "Massachusetts" .
_:johnaddress     exterms:postalCode  "01730" .
```

RDF

Datentypen

- **Verweis auf extern definierte Datentypen möglich, z.B. aus XMLSchema**

```
<!DOCTYPE rdf:RDF [!ENTITY xsd "http://www.w3.org/2001/XMLSchema#"]>
```

```
ex:index.html    exterms:creation-date    "1999-08-16"^^xsd:date .
```

ausführliche Deklaration:

```
1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:exterms="http://www.example.org/terms/"
4.     <rdf:Description rdf:about="http://www.example.org/index.html">
5.         <exterms:creation-date rdf:datatype=
6.             "http://www.w3.org/2001/XMLSchema#date">1999-08-16
7.         </exterms:creation-date>
8.     </rdf:Description>
9. </rdf:RDF>
```

RDF

Datentypen

- **Es gibt die RDF-Strukturen Bag, Seq, Alt**

Bag: Mengen, wobei Mehrfachvorkommen extra gezählt werden

Seq: Tupel, Listen

Alt: Alternativen (logisches Oder, nicht ausschließlich (?))

RDF Schema

Objektorientierte Klassenhierarchien

- Unterscheidung zwischen Klassen und Instanzen
- Vererbung über Unterklassenbildung (`rdfs:subClassOf`)
- Mehrfachvererbung erlaubt
- Ressourcen sind beliebige Objekte (Instanz oder Klasse)
- Über `rdf:type` wird eine Instanz einer Klasse zugeordnet.

RDF Schema

Semantische Strukturierung durch Basisklassen

- **`rdfs:Resource`: Basisklasse für alle Objekte**
- **`rdfs:Class`: Basisklasse für beliebige Klassen**
- **`rdf:Property`: Basisklasse für beliebige Relationen
(Unterklassen werden durch `subPropertyOf` gebildet)**
- **`rdfs:Literal`: Basisklasse für Strings**

RDF Schema

Typisierung für Relationen

- **`rdfs:domain`: legt Definitionsbereich einer Relation fest**
- **`rdfs:range`: legt Zielmenge einer Relation fest**
- **Mehrere Festlegungen gelten konjunktiv: Instanzen müssen zu allen Klassen gehören, die festgelegt wurden.**

OWL

Feinere Deklarationsmöglichkeiten als RDFS:

- Die Zielmenge einer Relation soll vom Definitionsbereich abhängen dürfen.
- Klassen sollen explizit als disjunkt deklariert werden können.
- Mengentheoretische Operationen auf Klassen
- Spezielle Eigenschaften von Relationen wie Symmetrie, etc.
- Eindeutigkeit bzw. genaue Kardinalität der Relationswerte

OWL

Automatische Prüfungen und Inferenzen:

- **Konsistenzprüfungen werden ermöglicht für bestimmte Modellchecker, die auf Beschreibungslogiken beruhen.**
- **Instanzen können automatisch über Vererbung und Relationseigenschaften klassifiziert werden.**
- **Relationen können über andere Relationen mit vordefinierten Eigenschaften automatisch generiert werden.**

OWL

Verschiedene Versionen:

- **OWL Full: sehr mächtig, aber unentscheidbar, kann nicht auf Konsistenz überprüft werden**
- **OWL-DL: übliche Version, zum Beispiel durch Protege-Editor unterstützt, kann auf Konsistenz überprüft werden.**
- **OWL Lite: abgespeckte Version zum schnelleren Selberbauen**

OWL

Verschiedene Objektklassentypen:

- **owl:Class** ist die Mutterklasse von Instanzen, die keine Relationen beschreiben
- **owl:ObjectProperty** bezieht sich auf Relationen, die zwischen Instanzen von **owl:Class** bestehen.
- **owl:DatatypeProperty** bezieht sich auf Relationen, die zwischen einer Instanz von **owl:Class** auf der einen Seite und einem RDF-Literal oder einem Datentyp aus XML-Schema auf der anderen Seite bestehen.

OWL

Spezielle Relationseigenschaften:

- **owl:TransitiveProperty ist die Mutterklasse von Relationen, die transitiv sind**

```
<owl:ObjectProperty rdf:ID="locatedIn">  
  <rdf:type rdf:resource="#owl:TransitiveProperty" />  
  <rdfs:domain rdf:resource="#owl:Thing" />  
  <rdfs:range rdf:resource="#Region" />  
</owl:ObjectProperty>
```

```
<Region rdf:ID="SantaCruzMountainsRegion">  
  <locatedIn rdf:resource="#CaliforniaRegion" />  
</Region>
```

```
<Region rdf:ID="CaliforniaRegion">  
  <locatedIn rdf:resource="#USRegion" />  
</Region>
```

- **Analog sind definiert: owl:SymmetricProperty, owl:FunctionalProperty, owl:InverseFunctionalProperty**

OWL

Spezielle Relationseigenschaften:

- **Existenzquantoren und Allquantoren werden definiert für Properties in anonymen Klassen:**

```
<owl:Class rdf:about="&OntologyPOI;GeoPlace">  
  <rdfs:subClassOf>  
    <owl:Restriction> ← anonyme Klasse  
      <owl:onProperty rdf:resource="&OntologyPOI;latitude"/>  
      <owl:allValuesFrom rdf:resource="&xsd;float"/>  
    </owl:Restriction>  
  </rdfs:subClassOf>  
</owl:Class>
```

Allquantor:

Jeder Wert von Relation latitude in einer Instanz von GeoPlace muss ein Float sein, aber es muss keine Werte geben.

```
<owl:Class rdf:about="&OntologyPOI;GeoPlace">  
  <rdfs:subClassOf>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="&OntologyPOI;latitude"/>  
      <owl:someValuesFrom rdf:resource="&xsd;float"/>  
    </owl:Restriction>  
  </rdfs:subClassOf>  
</owl:Class>
```

Existenzquantor:

Mindestens ein Wert von Relation latitude in einer Instanz von GeoPlace muss ein Float sein, und es muss auch mindestens einen geben.

Editor für das Semantic Web

Protege

- **Open Source Tool**
verschiedener Universitäten
- **generiert aus graphischer**
Bedienungsfläche
automatisch XML-Code

