



OWL - Syntax & Intuition

Dr. Sebastian Rudolph

Einleitung und XML

Einführung in RDF

RDF Schema

Logik - Grundlagen

Semantik von RDF(S)

OWL - Syntax und Intuition

OWL - Semantik und Reasoning

OWL 2

SPARQL - Syntax und Intuition

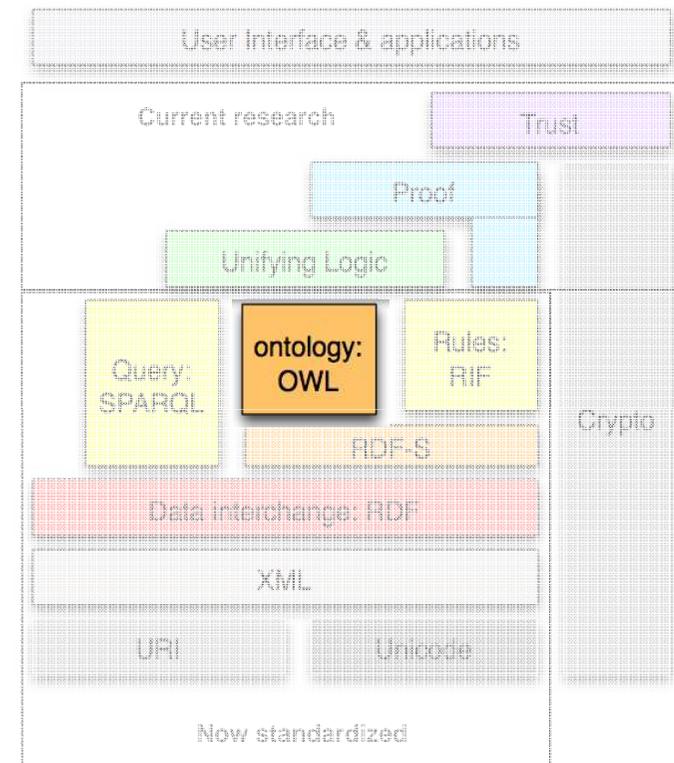
Konjunktive Anfragen / Einführung Regelsprachen

Regeln für OWL

Ontology Engineering

Semantic Web - Anwendungen

Semantic Web Architecture



Agenda

- Motivation
- OWL - Allgemeines
- Klassen, Rollen und Individuen
- Klassenbeziehungen
- komplexe Klassen
- Eigenschaften von Rollen
- OWL Varianten
- Anfragen an OWL-Ontologien

Agenda

- **Motivation**
- OWL - Allgemeines
- Klassen, Rollen und Individuen
- Klassenbeziehungen
- komplexe Klassen
- Eigenschaften von Rollen
- OWL Varianten
- Anfragen an OWL-Ontologien

Ontologie –philosophisch

- Begriff existiert nur in der Einzahl (es gibt also keine „Ontologien“)
- bezeichnet die „*Lehre vom Sein*“
- zu finden bei Aristoteles (Sokrates), Thomas von Aquin, Descartes, Kant, Hegel, Wittgenstein, Heidegger, Quine, ...

Ontologie – informatisch

Gruber (1993):

„An Ontology is a
formal specification
of a shared
conceptualization
of a domain of interest“

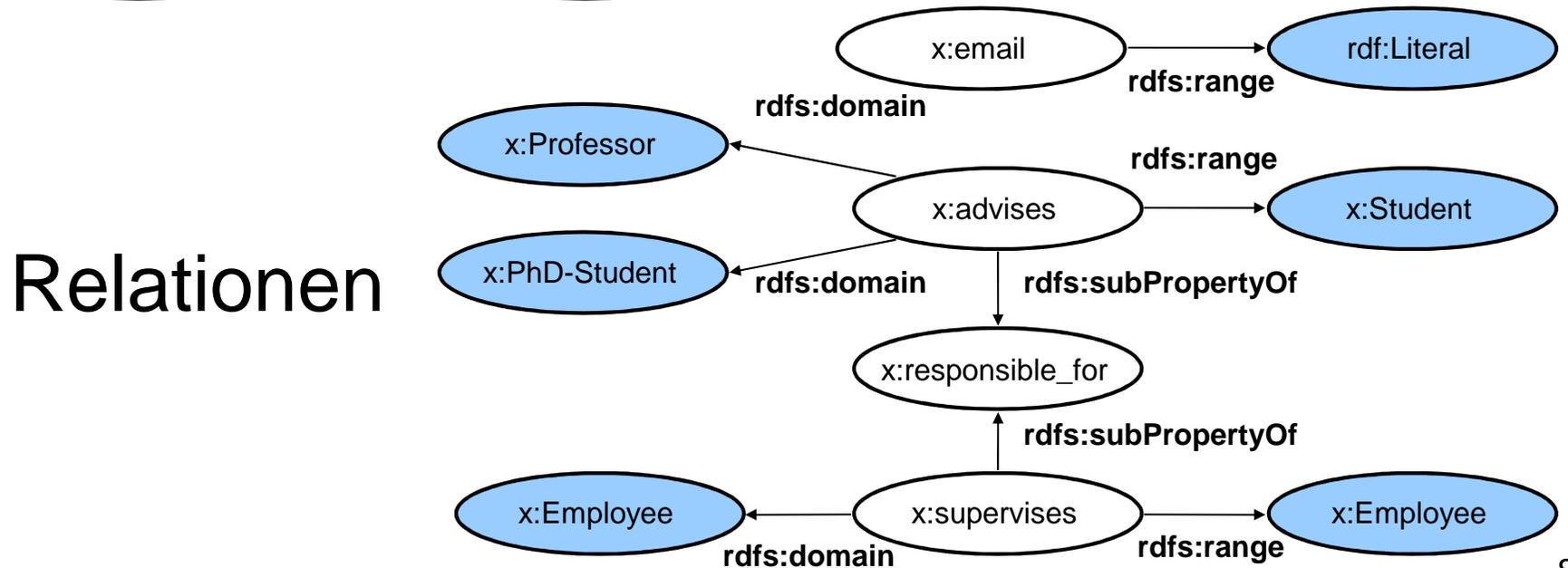
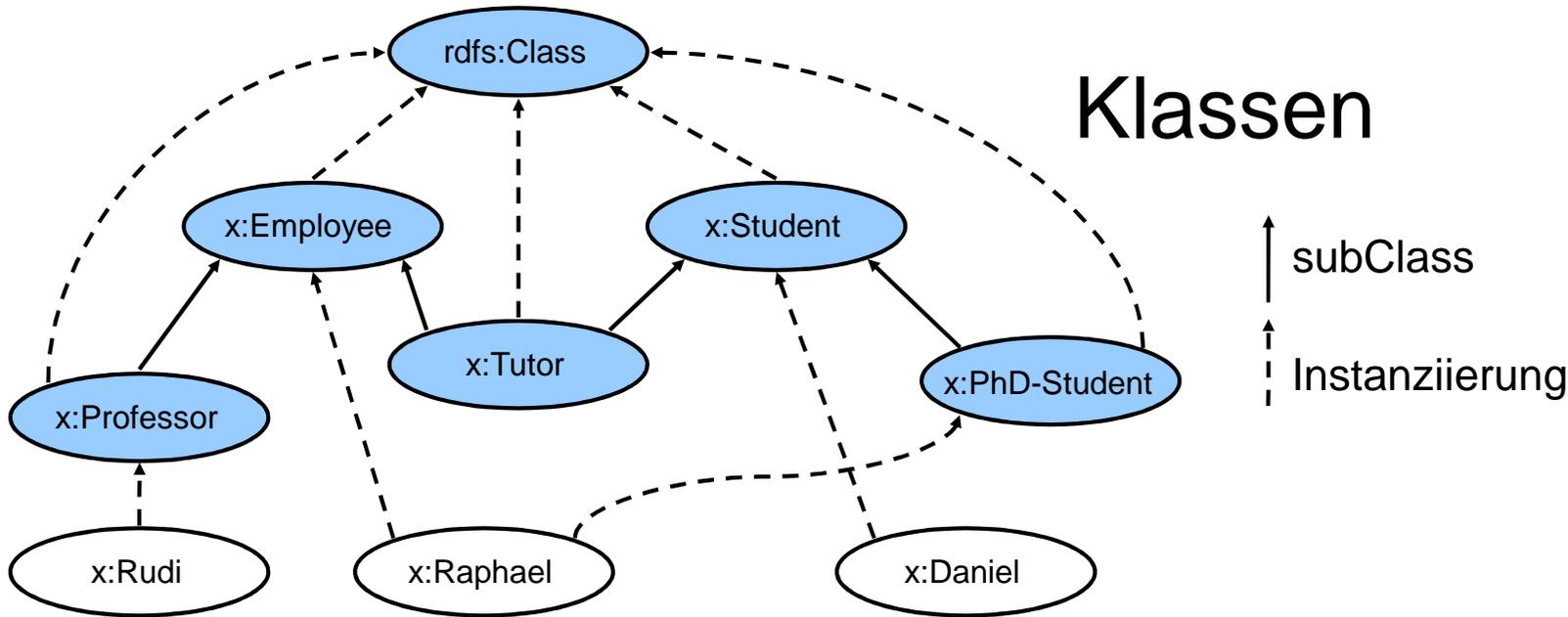
- maschinell interpretierbar
- beruht auf Konsens
- beschreibt Begrifflichkeiten
- bezogen auf ein „Thema“
(Gegenstandsbereich)

Ontologie – praktisch

Einige Anforderungen

- Instanziierung von Klassen durch Individuen
- Begriffshierarchien (Taxonomien, „Vererbung“):
Klassen, Begriffe
- binäre Relationen zwischen Individuen: Properties,
Roles
- Eigenschaften von Relationen (z.B. range, transitive)
- Datentypen (z.B. Zahlen): concrete domains
- logische Ausdrucksmittel
- klare Semantik!

RDFS - einfache Ontologien



RDF Schema als Ontologiesprache?

- geeignet für einfache Ontologien
- Vorteil: automatisches Schlussfolgern ist relativ effizient
- aber: für komplexere Modellierungen ungeeignet
- Rückgriff auf mächtigere Sprachen, wie
 - OWL
 - F-Logik

Agenda

- Motivation
- **OWL - Allgemeines**
- Klassen, Rollen und Individuen
- Klassenbeziehungen
- komplexe Klassen
- Eigenschaften von Rollen
- OWL Varianten
- Anfragen an OWL-Ontologien

OWL - Allgemeines

- W3C Recommendation seit 2004
- Semantisches Fragment von FOL
- Drei Varianten:
OWL Lite → OWL DL → OWL Full
- Keine Reifikation in OWL DL
→ RDFS ist Fragment von OWL Full
- OWL DL ist entscheidbar
entspricht der Beschreibungslogik SHOIN(D)
- W3C-Dokumente enthalten Details, die hier nicht alle angesprochen werden können.

OWL Varianten

- OWL Full
 - Enthält OWL DL und OWL Lite
 - Enthält als einzige OWL-Teilsprache ganz RDFS
 - Unentscheidbar.
 - Wird durch aktuelle Softwarewerkzeuge nur bedingt unterstützt.
- OWL DL
 - Enthält OWL Lite und ist Teilsprache von OWL Full.
 - Entscheidbar.
 - Wird von aktuellen Softwarewerkzeugen fast vollständig unterstützt.
 - Komplexität NExpTime (worst-case).
- OWL Lite
 - Ist Teilsprache von OWL DL und OWL Full.
 - Entscheidbar.
 - Wenig ausdrucksstark.
 - Komplexität ExpTime (worst-case).

OWL Dokumente

- sind RDF Dokumente
(zumindest in der Standard-Syntax; es gibt auch andere)
- bestehen aus
 - Kopf mit allgemeinen Angaben
 - Rest mit der eigentlichen Ontologie

Der Kopf eines OWL Dokumentes

Definition von Namespaces in der Wurzel

```
<rdf:RDF
```

```
  xmlns = "http://www.semanticweb-grundlagen.de/beispielontologie#"
```

```
  xmlns:rdf    = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
  xmlns:xsd    = "http://www.w3.org/2001/XMLSchema#"
```

```
  xmlns:rdfs   = "http://www.w3.org/2000/01/rdf-schema#"
```

```
  xmlns:owl    = "http://www.w3.org/2002/07/owl#" >
```

```
...
```

```
</rdf:RDF>
```

Der Kopf eines OWL Dokumentes

Allgemeine Informationen

```
<owl:Ontology rdf:about="">  
  
  <rdfs:comment  
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">  
  
    SWRC Ontologie in der Version vom Dezember 2005  
  
  </rdfs:comment>  
  
  <owl:versionInfo>v0.5</owl:versionInfo>  
  
  <owl:imports rdf:resource="http://www.semanticweb-  
    grundlagen.de/foo"/>  
  
  <owl:priorVersion  
    rdf:resource="http://ontoware.org/projects/swrc"/>  
  
</owl:Ontology>
```

Der Kopf eines OWL Dokumentes

von RDFS geerbt

`rdfs:comment`

`rdfs:label`

`rdfs:seeAlso`

`rdfs:isDefinedBy`

außerdem

`owl:imports`

für Versionierung

`owl:versionInfo`

`owl:priorVersion`

`owl:backwardCompatibleWith`

`owl:incompatibleWith`

`owl:DeprecatedClass`

`owl:DeprecatedProperty`

Agenda

- Motivation
- OWL - Allgemeines
- **Klassen, Rollen und Individuen**
- Klassenbeziehungen
- komplexe Klassen
- Eigenschaften von Rollen
- OWL Varianten
- Anfragen an OWL-Ontologien

Klassen, Rollen und Individuen

Die drei Bausteine von Ontologieaxiomen.

Klassen

Vergleichbar mit Klassen in RDFS

Individuen

Vergleichbar mit Objekten in RDFS

Rollen

Vergleichbar mit Properties in RDFS

Klassen

Definition

```
<owl:Class rdf:ID="Professor" />
```

vordefiniert:

```
owl:Thing
```

```
owl:Nothing
```

Individuen

Definition durch Klassenzugehörigkeit

```
<rdf:Description rdf:ID="RudiStuder">  
<rdf:type rdf:resource="#Professor" />  
</rdf:Description>
```

gleichbedeutend:

```
<Professor rdf:ID="RudiStuder" />
```

abstrakte Rollen

abstrakte Rollen werden definiert wie Klassen

```
<owl:ObjectProperty  
  rdf:ID="Zugehoerigkeit" />
```

Domain und Range abstrakter Rollen

```
<owl:ObjectProperty rdf:ID="Zugehoerigkeit">  
  <rdfs:domain rdf:resource="#Person" />  
  <rdfs:range rdf:resource="#Organisation" />  
</owl:ObjectProperty>
```

konkrete Rollen

konkrete Rollen haben Datentypen im Range

```
<owl:DatatypeProperty rdf:ID="Vorname" />
```

Domain und Range konkreter Rollen

```
<owl:DatatypeProperty rdf:ID="Vorname">
```

```
  <rdfs:domain rdf:resource="#Person" />
```

```
  <rdfs:range   rdf:resource="&xsd:string" />
```

```
</owl:DatatypeProperty>
```

Viele XML Datentypen können verwendet werden.

Im Standard vorgeschrieben sind `integer` und `string`.

Individuen und Rollen

```
<Person rdf:ID="RudiStuder">  
  <Zugehoerigkeit rdf:resource="#AIFB"/>  
  <Zugehoerigkeit rdf:resource="#ontoprise"/>  
  <Vorname rdf:datatype="&xsd:string">Rudi</Vorname>  
</Person>
```

Rollen sind im allgemeinen nicht funktional.

Agenda

- Motivation
- OWL - Allgemeines
- Klassen, Rollen und Individuen
- **Klassenbeziehungen**
- komplexe Klassen
- Eigenschaften von Rollen
- OWL Varianten
- Anfragen an OWL-Ontologien

Einfache Klassenbeziehungen

```
<owl:Class rdf:ID="Professor">  
  <rdfs:subClassOf  
    rdf:resource="#Fakultaetsmitglied"/>  
</owl:Class>  
  
<owl:Class rdf:ID="Fakultaetsmitglied">  
  <rdfs:subClassOf rdf:resource="#Person"/>  
</owl:Class>
```

Es folgt durch Inferenz, dass `Professor` eine Subklasse von `Person` ist.

Einfache Klassenbeziehungen

```
<owl:Class rdf:ID="Professor">
  <rdfs:subClassOf
    rdf:resource="#Fakultaetsmitglied"/>
</owl:Class>
<owl:Class rdf:ID="Buch">
  <rdfs:subClassOf rdf:resource="#Publikation"/>
</owl:Class>
<owl:Class rdf:about="#Fakultaetsmitglied">
  <owl:disjointWith rdf:resource="#Publikation"/>
</owl:Class>
```

Es folgt durch Inferenz, dass Professor und Buch ebenfalls disjunkte Klassen sind.

Einfache Klassenbeziehungen

```
<owl:Class rdf:ID="Buch">  
  <rdfs:subClassOf rdf:resource="#Publikation"/>  
</owl:Class>
```

```
<owl:Class rdf:about="#Publikation">  
  <owl:equivalentClass  
    rdf:resource="#Publikation"/>  
</owl:Class>
```

Es folgt durch Inferenz, dass Buch eine Subklasse von Publication ist.

Individuen und Klassenbeziehungen

```
<Buch rdf:ID="SemanticWebGrundlagen" >
  <Autor rdf:resource="#PascalHitzler" />
  <Autor rdf:resource="#MarkusKrötzsch" />
  <Autor rdf:resource="#SebastianRudolph" />
  <Autor rdf:resource="#YorkSure" />
</Buch>

<owl:Class rdf:about="#Buch" >
  <rdfs:subClassOf rdf:resource="#Publikation" />
</owl:Class>
```

Es folgt durch Inferenz, dass `SemanticWebGrundlagen` eine `Publikation` ist.

Beziehungen zwischen Individuen

```
<Professor rdf:ID="RudiStuder" />
```

```
<rdf:Description rdf:about="#RudiStuder">
```

```
  <owl:sameAs
```

```
    rdf:resource="#ProfessorStuder" />
```

```
</rdf:Description>
```

Es folgt durch Inferenz, dass `ProfessorStuder` ein Professor ist.

Verschiedenheit von Individuen mittels

`owl:differentFrom`.

Beziehungen zwischen Individuen

```
<owl:AllDifferent>
```

```
  <owl:distinctMembers  
    rdf:parseType="Collection">
```

```
    <Person rdf:about="#RudiStuder" />
```

```
    <Person rdf:about="#YorkSure" />
```

```
    <Person rdf:about="#PascalHitzler" />
```

```
  </owl:distinctMembers>
```

```
</owl:AllDifferent>
```

Abgekürzte Schreibweise anstelle der Verwendung von mehreren `owl:differentFrom`.

Der Einsatz von `owl:AllDifferent` und `owl:distinctMembers` ist nur dafür vorgesehen.

Abgeschlossene Klassen

```
<owl:Class rdf:about="#SekretaerinnenVonStuder">  
  <owl:oneOf rdf:parseType="Collection">  
    <Person rdf:about="#GiselaSchillinger"/>  
    <Person rdf:about="#BeateKuehner"/>  
  </owl:oneOf>  
</owl:Class>
```

Dies besagt, dass es nur **genau diese beiden** SekretaeerinnenVonStuder gibt.

Agenda

- Motivation
- OWL - Allgemeines
- Klassen, Rollen und Individuen
- Klassenbeziehungen
- **komplexe Klassen**
- Eigenschaften von Rollen
- OWL Varianten
- Anfragen an OWL-Ontologien

Logische Klassenkonstruktoren

- logisches Und (Konjunktion):
`owl:intersectionOf`
- logisches Oder (Disjunktion):
`owl:unionOf`
- logisches Nicht (Negation):
`owl:complementOf`
- Werden verwendet, um komplexe Klassen aus einfachen Klassen zu konstruieren.

Konjunktion

```
<owl:Class rdf:about="#SekretaerinnenVonStuder">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Sekretaerinnen"/>
        <owl:Class rdf:about="#AngehoeerigeAGStuder"/>
      </owl:intersectionOf>
    <owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

Es folgt z.B. durch Inferenz, dass alle
SekretaerinnenVonStuder auch
Sekretaerinnen sind.

Disjunktion

```
<owl:Class rdf:about="#Professor">
  <rdfs:subClassOf>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#aktivLehrend"/>
        <owl:Class rdf:about="#imRuhestand"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
```

Negation

```
<owl:Class rdf:about="#Fakultaetsmitglied">
  <rdfs:subClassOf>
    <owl:Class>
      <owl:complementOf rdf:resource="#Publikation"/>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
```

semantisch äquivalente Aussage:

```
<owl:Class rdf:about="#Fakultaetsmitglied">
  <owl:disjointWith rdf:resource="#Publikation"/>
</owl:Class>
```

Rolleneinschränkungen (allValuesFrom)

dienen der Definition komplexer Klassen durch Rollen

```
<owl:Class rdf:ID="Pruefung">  
  <rdfs:subClassOf>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="#hatPruefer"/>  
      <owl:allValuesFrom rdf:resource="#Professor"/>  
    </owl:Restriction>  
  </rdfs:subClassOf>  
</owl:Class>
```

D.h. *alle* Prüfer einer Prüfung müssen Professoren sein.

Rolleneinschränkungen (someValuesFrom)

```
<owl:Class rdf:about="#Pruefung">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hatPruefer" />
      <owl:someValuesFrom rdf:resource="#Person" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

D.h. jede Prüfung muss *mindestens einen* Prüfer haben.

Rolleneinschränkungen (Kardinalitäten)

```
<owl:Class rdf:about="#Pruefung">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hatPruefer"/>
      <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">
        2
      </owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Eine Prüfung kann *höchstens zwei* Prüfer haben.

Rolleneinschränkungen (Kardinalitäten)

```
<owl:Class rdf:about="#Pruefung">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hatThema" />
      <owl:minCardinality
rdf:datatype="&xsd;nonNegativeInteger">
3
      </owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Eine Prüfung muss sich über *mindestens drei* Themengebiete erstrecken.

Rolleneinschränkungen (Kardinalitäten)

```
<owl:Class rdf:about="#Pruefung">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hatThema" />
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">
        3
      </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Eine Prüfung muss sich über *genau drei* Themengebiete erstrecken.

Rolleneinschränkungen (hasValue)

```
<owl:Class rdf:ID="PruefungBeiStuder">
  <rdfs:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hatPruefer"/>
      <owl:hasValue rdf:resource="#RudiStuder"/>
    </owl:Restriction>
  </rdfs:equivalentClass>
</owl:Class>
```

`owl:hasValue` verweist immer auf eine konkrete Instanz. Dies ist äquivalent zum Beispiel auf der nächsten Folie.

Rolleneinschränkungen (hasValue)

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

  <owl:equivalentClass>

    <owl:Restriction>

      <owl:onProperty rdf:resource="#hatPruefer"/>

      <owl:someValuesFrom>

        <owl:oneOf rdf:parseType="Collection">

          <owl:Thing rdf:about="#RudiStuder"/>

        </owl:oneOf>

      </owl:someValuesFrom>

    </owl:Restriction>

  </owl:equivalentClass>

</owl:Class>
```

Agenda

- Motivation
- OWL - Allgemeines
- Klassen, Rollen und Individuen
- Klassenbeziehungen
- komplexe Klassen
- **Eigenschaften von Rollen**
- OWL Varianten
- Anfragen an OWL-Ontologien

Rollenbeziehungen

```
<owl:ObjectProperty rdf:ID="hatPruefer">  
  <rdfs:subPropertyOf  
    rdf:resource="#hatAnwesenden" />  
</owl:ObjectProperty>
```

Ebenso: `owl:equivalentProperty`

Rollen können auch invers zueinander sein:

```
<owl:ObjectProperty rdf:ID="hatPruefer">  
  <owl:inverseOf rdf:resource="#prueferVon" />  
</owl:ObjectProperty>
```

Rolleneigenschaften

- Domain
- Range
- Transitivität, d.h.
 $r(a,b)$ und $r(b,c)$ impliziert $r(a,c)$
- Symmetrie, d.h.
 $r(a,b)$ impliziert $r(b,a)$
- Funktionalität
 $r(a,b)$ und $r(a,c)$ impliziert $b=c$
- Inverse Funktionalität
 $r(a,b)$ und $r(c,b)$ impliziert $a=c$

Domain und Range

```
<owl:ObjectProperty rdf:ID="Zugehoerigkeit">  
  <rdfs:range rdf:resource="#Organisation"/>  
</owl:ObjectProperty>
```

ist gleichbedeutend mit dem Folgenden:

```
<owl:Class rdf:about="\&owl;Thing">  
  <rdfs:subClassOf>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="#Zugehoerigkeit"/>  
      <owl:allValuesFrom rdf:resource="#Organisation"/>  
    </owl:Restriction>  
  </rdfs:subClassOf>  
</owl:Class>
```

Domain und Range: Vorsicht!

```
<owl:ObjectProperty rdf:ID="Zugehoerigkeit">  
  <rdfs:range rdf:resource="#Organisation"/>  
</owl:ObjectProperty>  
  
<Zahl rdf:ID="Fuenf">  
  <Zugehoerigkeit rdf:resource="#Primzahlen"/>  
</Zahl>
```

Es folgt nun, dass Primzahlen eine
Organisation ist!

Rolleneigenschaften

```
<owl:ObjectProperty rdf:ID="hatKollegen">
  <rdf:type rdf:resource="&owl;TransitiveProperty" />
  <rdf:type rdf:resource="&owl;SymmetricProperty" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hatProjektleiter">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="istProjektleiterFuer">
  <rdf:type
    rdf:resource="&owl;InverseFunctionalProperty" />
</owl:ObjectProperty>
<Person rdf:ID="YorkSure">
  <hatKollegen rdf:resource="#PascalHitzler" />
  <hatKollegen rdf:resource="#AnupriyaAnkolekar" />
  <istProjektleiterFuer rdf:resource="#SEKT" />
</Person>
<Projekt rdf:ID="SmartWeb">
  <hatProjektleiter rdf:resource="#PascalHitzler" />
  <hatProjektleiter rdf:resource="#HitzlerPascal" />
</Projekt>
```

Folgerungen aus dem Beispiel

- AnupriyaAnkolekar hatKollegen
YorkSure
- AnupriyaAnkolekar hatKollegen
PascalHitzler
- PascalHitzler owl:sameAs
HitzlerPascal

Agenda

- Motivation
- OWL - Allgemeines
- Klassen, Rollen und Individuen
- Klassenbeziehungen
- komplexe Klassen
- Eigenschaften von Rollen
- **OWL Varianten**
- Anfragen an OWL-Ontologien

OWL Varianten

- OWL Full
 - Enthält OWL DL und OWL Lite
 - Enthält als einzige OWL-Teilsprache ganz RDFS
 - Unentscheidbar.
 - Wird durch aktuelle Softwarewerkzeuge nur bedingt unterstützt.
- OWL DL
 - Enthält OWL Lite und ist Teilsprache von OWL Full.
 - Entscheidbar.
 - Wird von aktuellen Softwarewerkzeugen fast vollständig unterstützt.
 - Komplexität NExpTime (worst-case).
- OWL Lite
 - Ist Teilsprache von OWL DL und OWL Full.
 - Entscheidbar.
 - Wenig ausdrucksstark.
 - Komplexität ExpTime (worst-case).

OWL Full

- Uneingeschränkte Nutzung aller OWL und RDFS-Sprachelemente (muss gültiges RDFS sein).
- Schwierig z.B.: nicht vorhandene Typentrennung (Klassen, Rollen, Individuen), dadurch:
 - `owl:Thing` dasselbe wie `rdfs:resource`
 - `owl:Class` dasselbe wie `rdfs:Class`
 - `owl:DatatypeProperty` Subklasse von `owl:ObjectProperty`
 - `owl:ObjectProperty` dasselbe wie `rdf:Property`

Beispiel für Typendurchmischung in OWL Full

```
<owl:Class rdf:about="#Buch">  
  <englischerName rdf:datatype="&xsd:string">  
    book  
  </englischerName>  
  <franzoesischerName rdf:datatype="&xsd:string">  
    livre  
  </franzoesischerName>  
</owl:Class>
```

Inferenzen über solche Konstrukte werden oft nicht wirklich benötigt.

OWL DL

- Nur Verwendung von explizit erlaubten RDFS Sprachelementen (z.B. die in unseren Beispielen).
Nicht erlaubt: `rdfs:Class`, `rdfs:Property`
- Typentrennung. Klassen und Rollen müssen explizit deklariert werden.
- Konkrete Rollen dürfen nicht als transitiv, symmetrisch, invers oder invers funktional deklariert werden.
- Zahlenrestriktionen dürfen nicht mit transitiven Rollen, deren Subrollen, oder Inversen davon verwendet werden.

OWL Lite

- alle Einschränkungen für OWL DL
- außerdem:
 - nicht erlaubt: `oneOf`, `unionOf`,
`complementOf`, `hasValue`, `disjointWith`
 - Zahlenrestriktionen nur mit 0 und 1 erlaubt.
 - Einige Einschränkungen zum Auftreten von anonymen (komplexen) Klassen, z.B. nur im Subjekt von `rdfs:subClassOf`.

Agenda

- Motivation
- OWL - Allgemeines
- Klassen, Rollen und Individuen
- Klassenbeziehungen
- komplexe Klassen
- Eigenschaften von Rollen
- OWL Varianten
- **Anfragen an OWL-Ontologien**

Terminologische Anfragen an OWL (nur Klassen und Rollen)

- Klassenäquivalenz
- Subklassenbeziehung
- Disjunktheit von Klassen
- globale Konsistenz (Erfüllbarkeit, Widerspruchsfreiheit)
- Klassenkonsistenz: Eine Klasse ist *inkonsistent*, wenn sie äquivalent zu `owl:Nothing` ist - dies deutet oft auf einen Modellierungsfehler hin:

```
<owl:Class rdf:about="#Buch">  
  <owl:subClassOf rdf:resource="#Publikation"/>  
  <owl:disjointWith rdf:resource="#Publikation"/>  
</owl:Class>
```

Assertionale Anfragen an OWL (mit Individuen)

- Instanzüberprüfung: Gehört gegebenes Individuum zu gegebener Klasse?
- Suche nach allen Individuen, die in einer Klasse enthalten sind.
- Werden zwei gegebene Individuen durch Rolle verknüpft?
- Suche nach allen Individuenpaaren, die durch eine Rolle verknüpft sind.
- ...Vorsicht: es wird nur nach „beweisbaren“ Antworten gesucht!

OWL Werkzeuge

- Editoren
 - Protegé, <http://protege.stanford.edu>
 - SWOOP, <http://www.mindswap.org/2004/SWOOP/>
 - OWL Tools, <http://owltools.ontoware.org/>
- Inferenzmaschinen
 - Pellet, <http://www.mindswap.org/2003/pellet/index.shtml>
 - KAON2, <http://kaon2.semanticweb.org>
 - FACT++, <http://owl.man.ac.uk/factplusplus/>
 - Racer, <http://www.racer-systems.com/>
 - Cerebra, <http://www.cerebra.com/index.html>

OWL Sprachelemente

- Kopf
- `rdfs:comment`
- `rdfs:label`
- `rdfs:seeAlso`
- `rdfs:isDefinedBy`
- `owl:versionInfo`
- `owl:priorVersion`
- `owl:backwardCompatibleWith`
- `owl:incompatibleWith`
- `owl:DeprecatedClass`
- `owl:DeprecatedProperty`
- `owl:imports`

Beziehungen zwischen Individuen

- `owl:sameAs`
- `owl:differentFrom`
- `owl:AllDifferent`
(zusammen mit
`owl:distinctMembers`)

Vorgeschriebene Datentypen

- `xsd:string`
- `xsd:integer`

OWL Sprachelemente

- **Klassenkonstruktoren und -beziehungen**

- `owl:Class`
- `owl:Thing`
- `owl:Nothing`
- `rdfs:subClassOf`
- `owl:disjointWith`
- `owl:equivalentClass`
- `owl:intersectionOf`
- `owl:unionOf`
- `owl:complementOf`

Rollenrestriktionen

- `owl:allValuesFrom`
- `owl:someValuesFrom`
- `owl:hasValue`
- `owl:cardinality`
- `owl:minCardinality`
- `owl:maxCardinality`
- `owl:oneOf`

OWL Sprachelemente

- Rollenkonstruktoren, -beziehungen und -eigenschaften
 - `owl:ObjectProperty`
 - `owl:DatatypeProperty`
 - `rdfs:subPropertyOf`
 - `owl:equivalentProperty`
 - `owl:inverseOf`
 - `rdfs:domain`
 - `rdfs:range`
 - `rdf:resource="&owl;TransitiveProperty"`
 - `rdf:resource="&owl;SymmetricProperty"`
 - `rdf:resource="&owl;FunctionalProperty"`
 - `rdf:resource="&owl;InverseFunctionalProperty"`

Weiterführende Literatur

- <http://www.w3.org/2004/OWL/>
zentrale W3C Webseite für OWL.
- <http://www.w3.org/TR/owl-features/>
Überblick über OWL.
- <http://www.w3.org/TR/owl-ref/>
vollständige Beschreibung der OWL-Sprachkomponenten.
- <http://www.w3.org/TR/owl-guide/>
zeigt, wie OWL zur Wissensmodellierung verwendet werden kann.
- <http://www.w3.org/TR/owl-semantics/>
beschreibt die Semantik von OWL, die wir auf andere Weise später behandeln werden. Es beschreibt außerdem die abstrakte Syntax für OWL DL, die wir hier später noch ansprechen.
- Deutsche Übersetzungen mancher W3C Dokumente findet man unter <http://www.w3.org/2005/11/Translations/Lists/ListLang-de.html>