Kapitel WT:VIII (Fortsetzung)

VIII. Semantic Web

- WWW heute
- Semantic Web Vision
- □ RDF: Einführung
- □ RDF: Konzepte
- □ RDF: XML-Serialisierung
- □ RDF: Anwendungen
- □ RDFS: Einführung
- □ RDFS: Konzepte
- Semantik im Web
- □ Semantik von RDF/RDFS
- Ontologien
- □ OWL: Konzepte
- OWL: Logikhintergrund
- □ OWL: Anwendungen

WT:VIII-171 Semantic Web: Ontology © STEIN 2005-2014

Definition

Ontologie [griechisch] – die Lehre vom Seienden

WT:VIII-172 Semantic Web: Ontology © STEIN 2005-2014

Definition

Ontologie [griechisch] – die Lehre vom Seienden

Definition 1 (Ontology [Gruber 1993])

An ontology is a formal specification of a shared conceptualization of a domain of interest.

formal specification

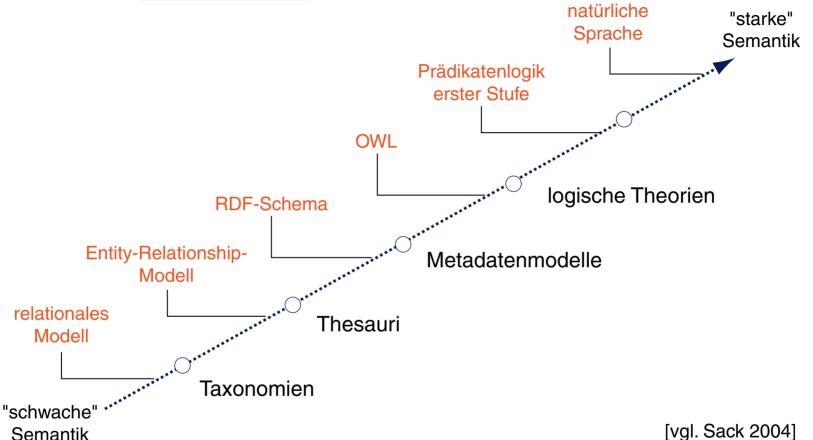
of a shared

conceptualization

of a domain of interest

- interpretierbar durch Maschinen
- → gemeinsames Verständnis einer Gruppe
- es geht um Begriffsbildung
- bekannter (abgegrenzter) Gegenstandsbereich

Einteilung nach Ausdrucksstärke



Je reicher die Möglichkeiten zur Ontologiebeschreibung, um so stärker ist die intendierte Semantik, und um so komplexer können Schlussfolgerungsprozesse im Worst-Case werden.

WT:VIII-174 Semantic Web: Ontology © STEIN 2005-2014

Taxonomien

Um 1740 entwickelt Carl von Lineé ein hierarchisches Klassifikationsschema für Tiere und Pflanzen.



Systematik der Silbermöwe:

Reich (Regnum) Tiere

Unterreich (Subregnum) Vielzeller Metazoa

Abteilung (Divisio) Vielzeller im engeren Sinne Eumetazoa

Stamm (Phylum) Chordatiere Chordata

Unterstamm (Subphylum) Wirbeltiere Vertebrata

Klasse (Classis) Vögel Aves

Ordnung (Ordo) Wat- und Möwenvögel Charadriiformes

Unterordnung (Subordo) Möwenartige Lari

Familie (Familia) Möwenvögel Laridae

Unterfamilie (Subfamilia) Larinae

Gattung (Genus) Möwen Larus

Art (Species) Silbermöwe Larus argentatus

WT:VIII-175 Semantic Web: Ontology

© STEIN 2005-2014

Sinn und Zweck

Ontologien ermöglichen:

- Realisierung "semantischer" Suchfunktionen
 Beispiele: automatische Spezialisierung, Generalisierung, Verfeinerung
- Realisierung "intelligenter" Dialoge mit Maschinen Beispiel: Nachfragen bei Mehrdeutigkeiten
- Interoperabilität beim Austausch Daten, die ein Markup besitzen

WT:VIII-176 Semantic Web: Ontology © STEIN 2005-2014

Sinn und Zweck

Ontologien ermöglichen:

- Realisierung "semantischer" Suchfunktionen
 Beispiele: automatische Spezialisierung, Generalisierung, Verfeinerung
- Realisierung "intelligenter" Dialoge mit Maschinen Beispiel: Nachfragen bei Mehrdeutigkeiten
- Interoperabilität beim Austausch Daten, die ein Markup besitzen

Elemente einer Ontologie-Sprache:

- Klassen, auch Konzepte genannt
- Eigenschaften der Klassen
- □ (binäre) Beziehungen zwischen Klassen und Eigenschaften

Anwendungen von formal beschriebenen Ontologien basieren darauf, logische Schlussfolgerungen über (Teil)Mengen von Konzepten zu ziehen.

WT:VIII-177 Semantic Web: Ontology © STEIN 2005-2014

Grenzen von RDF-Schema

- 1. RDF-Schema ist eine einfache Ontologie-Sprache.
- 2. "Echte" Ontologie-Sprachen stellen ein reicheres Beschreibungsvokabular als RDF-Schema zur Verfügung.

Einige Defizite von RDF-Schema:

- □ Mit rdfs:subClassOf kann keine Disjunktheit ausgedrückt werden.
- Die Individuen aller Unterklassen einer Klasse k\u00f6nnen nicht in ihrer Gesamtheit angesprochen werden.
- Kardinalitätsangaben für Properties sind nicht formulierbar.
- □ Spezielle Eigenschaften von Properties wie Transitivität, Umkehrbarkeit oder Funktionscharakter sind nicht formulierbar.

WT:VIII-178 Semantic Web: Ontology © STEIN 2005-2014



WT:VIII-179 Semantic Web: Ontology © STEIN 2005-2014

Bemerkungen:

□ Das Akronym für Web Ontology Language hätte eigentlich WOL, nicht OWL sein müssen. In Referenz auf die literarische Figur der Eule aus Milnes "Pu der Bär", die als einziges Tier im Wald ihren Namen schreiben konnte (und denselben aber mit einem Buchstabendreher im englischen Original WOL statt OWL schreibt), wurde dieser Buchstabendreher umgekehrt für die OWL übernommen.

[Wikipedia]

Historie von Ontologie-Sprachen für das Web

- 1995 SHOE. Simple HTML Ontology Extensions.
- 1997 OIL. Ontology Inference Layer.

 Die erste Ontologiesprache basierend auf RDF und XML-Schema.
- 2000 DAML. DARPA Agent Markup Language.
- DAML+OIL. Eine "Joint Ontology Language" zwischen Europa und USA. Konzepte dieser Sprache haben große Ähnlichkeit mit Konzepten in RDFS.
- 2001 Gründung der Ontology Working Group WebOnt des W3C.
- 2002 Erstes Release der "OWL Use Cases and Requirements".
- 2004 OWL Web Ontology Language. Semantik und abstrakte Syntax. Recommendation.

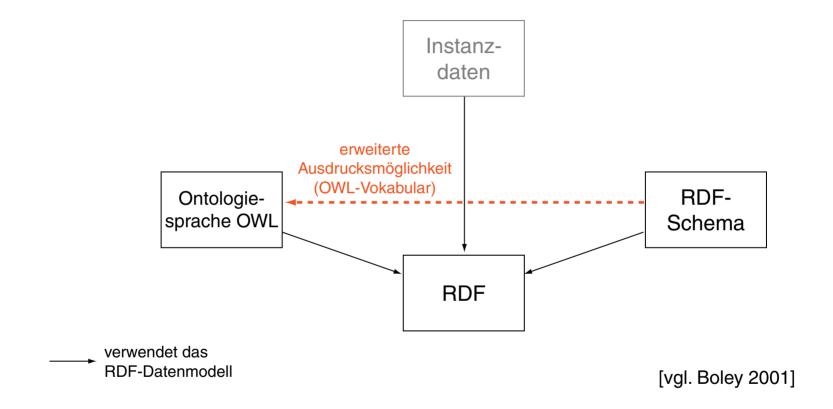
WT:VIII-181 Semantic Web: Ontology © STEIN 2005-2014

Zusammenhang zwischen OWL und RDF/RDFS



[vgl. Boley 2001]

Zusammenhang zwischen OWL und RDF/RDFS



Bemerkungen:

An OWL ontology is an RDF graph, which is in turn a set of RDF triples. As with any RDF graph, an OWL ontology graph can be written in many different syntactic forms.

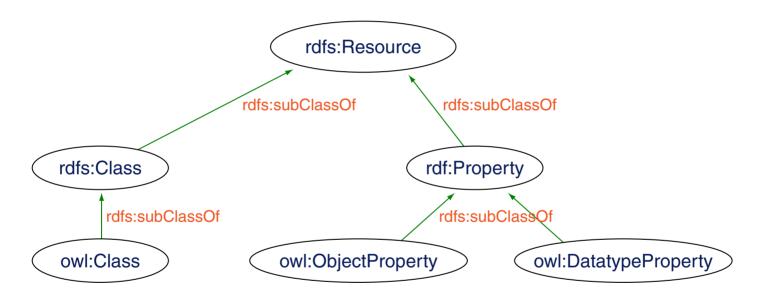
[www.w3.org/TR/owl-ref/]

- □ Zu den Design-Zielen von OWL gehört:
 - vernünftiger Kompromiss zwischen Ausdrucksfähigkeit und inferentieller Komplexität
 - Skalierbarkeit, insbesondere vor dem Hintergrund des World Wide Web
 - Kompatibilität zu existierenden Standards des WWW

Zusammenhang zwischen OWL und RDF/RDFS (Fortsetzung)

Eine OWL-Ontologie definiert Klassen, Individuen und Properties.

Individuen = Instanzen von Klassen

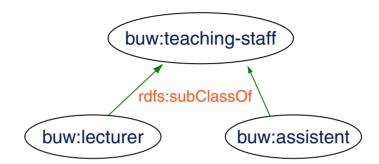


Unterscheidung von zwei Arten von Properties:

- 1. Object-Properties, um Beziehungen zwischen Individuen auszudrücken.
- 2. Datentyp-Properties, um Individuen mit Datentypen zu assoziieren.

Klassen

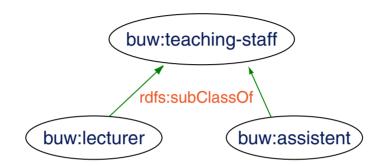
Definition von Klassen im Prinzip wie in RDFS:



```
<rdfs:Class rdf:about="http://www.buw.de/lecturer">
    <rdfs:subClassOf rdf:resource="http://www.buw.de/teaching-staff"/>
</rdfs:Class>
```

Klassen

Definition von Klassen im Prinzip wie in RDFS:



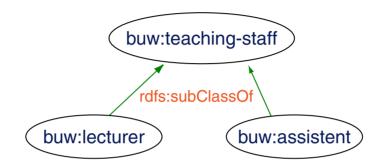
```
<rdfs:Class rdf:about="http://www.buw.de/lecturer">
    <rdfs:subClassOf rdf:resource="http://www.buw.de/teaching-staff"/>
</rdfs:Class>

<owl:Class rdf:about="http://www.buw.de/lecturer">
    <rdfs:subClassOf rdf:resource="http://www.buw.de/teaching-staff"/>
</owl:Class>
```

WT:VIII-187 Semantic Web: Ontology © STEIN 2005-2014

Klassen

Definition von Klassen im Prinzip wie in RDFS:



```
<rdfs:Class rdf:about="http://www.buw.de/lecturer">
    <rdfs:subClassOf rdf:resource="http://www.buw.de/teaching-staff"/>
</rdfs:Class>

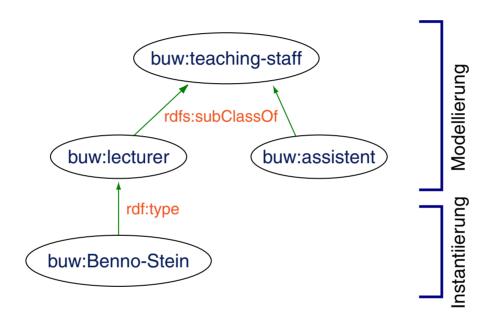
<owl:Class rdf:about="http://www.buw.de/lecturer">
    <rdfs:subClassOf rdf:resource="http://www.buw.de/teaching-staff"/>
    </owl:Class>

<owl:Class rdf:about="http://www.buw.de/lecturer">
    <owl:Class rdf:about="http://www.buw.de/lecturer">
    <owl:Class></owl:Class></owl:Class></owl:Class></owl:Class>
```

WT:VIII-188 Semantic Web: Ontology © STEIN 2005-2014

Klassen

Instanzen von Klassen (= Individuen) werden wie in RDF deklariert.



```
<rdf:Description rdf:about="http://www.buw.de/Benno-Stein">
   <rdf:type rdf:resource="http://www.buw.de/lecturer"/>
   </rdf:Description>
```

bzw. (abkürzende Schreibweise)

<buw:lecturer rdf:about="http://www.buw.de/Benno-Stein"/>

Klassen: Deklaration von Verknüpfungen und Restriktionen

- □ Vereinigung: owl:unionOf
- □ Durchschnittbildung: owl:intersectionOf
- Komplementbildung: owl:complementOf
- □ Disjunktheitsforderung: owl:disjointWith
- Definition durch Aufzählung: owl:oneOf
- Deklaration als Äquivalenzklasse: owl:equivalentClass

OWL besitzt zwei vordefinierte Klassen:

- 1. Jede Klasse ist Unterklasse der allgemeinsten Klasse owl: Thing.
- 2. Jeder Klasse ist Oberklasse der leeren Klasse owl: Nothing.

© STEIN 2005-2014

Klassen: Beispiele [vgl. Sack 2004]

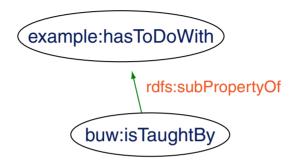
```
<owl:Class rdf:about="#Frucht">
  <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#SuessFrucht"/>
        <owl:Class rdf:about="#NichtSuessFrucht"/>
        </owl:unionOf>
</owl:Class>
```

Klassen: Beispiele [vgl. Sack 2004]

```
<owl:Class rdf:about="#Frucht">
 <owl:unionOf rdf:parseType="Collection">
   <owl:Class rdf:about="#SuessFrucht"/>
   <owl:Class rdf:about="#NichtSuessFrucht"/>
 </owl:unionOf>
</owl:Class>
<owl:Class rdf:ID="WeinFarbe">
 <rdfs:subClassOf rdf:resource="#WeinDeskriptor">
   <owl:oneOf rdf:parseType="Collection">
     <owl:Thing rdf:about="#Weiss"/>
     <owl:Thing rdf:about="#Rosé"/>
     <owl:Thing rdf:about="#Rot"/>
   </owl:oneOf>
 </rdfs:subClassOf>
</owl:Class>
```

Object-Properties

Definition von Object-Properties im Prinzip wie in RDFS:



```
<rdfs:Property rdf:about="http://www.buw.de/isTaughtBy">
  <rdfs:subPropertyOf rdf:resource="&example;hasToDoWith"/>
</rdfs:Property>
```

Object-Properties

Definition von Object-Properties im Prinzip wie in RDFS:

```
example:hasToDoWith

rdfs:subPropertyOf

buw:isTaughtBy
```

```
<rdfs:Property rdf:about="http://www.buw.de/isTaughtBy">
    <rdfs:subPropertyOf rdf:resource="&example; hasToDoWith"/>
</rdfs:Property>

<owl:ObjectProperty rdf:about="http://www.buw.de/isTaughtBy">
    <rdfs:subPropertyOf rdf:resource="&example; hasToDoWith"/>
    </owl:ObjectProperty>
```

Object-Properties

Definition von Object-Properties im Prinzip wie in RDFS:

```
example:hasToDoWith

rdfs:subPropertyOf

buw:isTaughtBy
```

```
<rdfs:Property rdf:about="http://www.buw.de/isTaughtBy">
    <rdfs:subPropertyOf rdf:resource="&example; hasToDoWith"/>
</rdfs:Property>

<owl:ObjectProperty rdf:about="http://www.buw.de/isTaughtBy">
    <rdfs:subPropertyOf rdf:resource="&example; hasToDoWith"/>
    </owl:ObjectProperty>
```

- 1. Properties können durch rdfs:subPropertyOf spezialisiert werden.
- 2. Für Properties können besondere Eigenschaften deklariert werden.
- 3. Domain und Range von Properties können eingeschränkt werden.

WT:VIII-195 Semantic Web: Ontology © STEIN 2005-2014

Object-Properties: Deklaration besonderer Eigenschaften (Characteristics)

□ Transitivität: owl:TransitiveProperty.

$$P(x,y) \wedge P(y,z) \rightarrow P(x,z)$$

□ Symmetrie: owl:SymmetricProperty.

$$P(x,y) \leftrightarrow P(y,x)$$

Object-Properties: Deklaration besonderer Eigenschaften (Characteristics)

□ Transitivität: owl:TransitiveProperty.

$$P(x,y) \wedge P(y,z) \rightarrow P(x,z)$$

□ Symmetrie: owl:SymmetricProperty.

$$P(x,y) \leftrightarrow P(y,x)$$

□ Funktionscharakter: owl:FunctionalProperty.

$$P(x,y) \wedge P(x,z) \rightarrow \textit{Equal}(y,z)$$

□ Umgekehrter Funktionscharakter: owl:InverseFunctionalProperty.

$$P(x,y) \wedge P(z,y) \rightarrow \textit{Equal}(x,z)$$

Object-Properties: Deklaration besonderer Eigenschaften (Characteristics)

□ Transitivität: owl:TransitiveProperty.

$$P(x,y) \wedge P(y,z) \rightarrow P(x,z)$$

□ Symmetrie: owl:SymmetricProperty.

$$P(x,y) \leftrightarrow P(y,x)$$

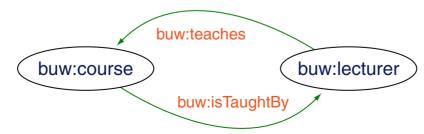
□ Funktionscharakter: owl:FunctionalProperty.

$$P(x,y) \wedge P(x,z) \rightarrow \textit{Equal}(y,z)$$

□ Umgekehrter Funktionscharakter: owl:InverseFunctionalProperty.

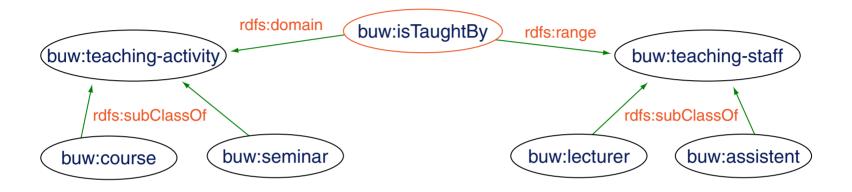
$$P(x,y) \wedge P(z,y) \rightarrow \textit{Equal}(x,z)$$

□ Umkehrbeziehung: owl:inverseOf.



Range-Einschränkung: Konstruktion spezialisierter Klassen I

Zugrundeliegendes Schema, Theorie oder Ontologie:

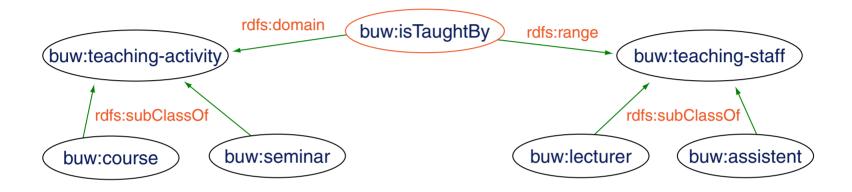


"Vorlesungen, die nur von Dozenten unterrichtet werden." =

"Die Elemente x aus buw:course, denen bzgl. der Property buw:isTaughtBy nur Range-Elemente y aus buw:lecturer zugeordnet sind."

Range-Einschränkung: Konstruktion spezialisierter Klassen I

Zugrundeliegendes Schema, Theorie oder Ontologie:



"Vorlesungen, die nur von Dozenten unterrichtet werden." =

"Die Elemente x aus buw:course, denen bzgl. der Property buw:isTaughtBy nur Range-Elemente y aus buw:lecturer zugeordnet sind."

Formulierung in Prädikatenlogik:

 $Course(x) \land (\forall y : IsTaughtBy(x, y) \rightarrow Lecturer(y))$

Range-Einschränkung: Konstruktion spezialisierter Klassen I

Formulierung in RDF/XML:

WT:VIII-201 Semantic Web: Ontology © STEIN 2005-2014

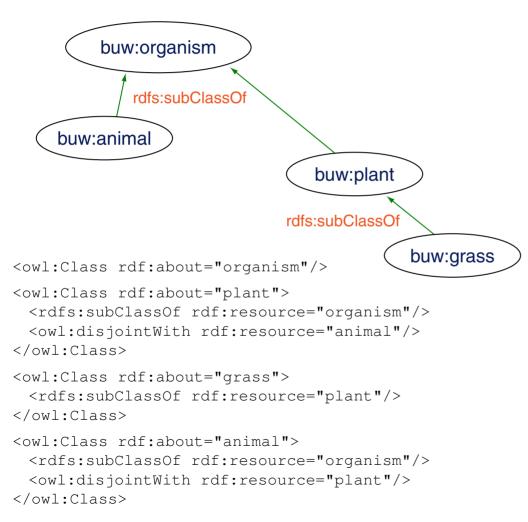
Range-Einschränkung: Konstruktion spezialisierter Klassen I

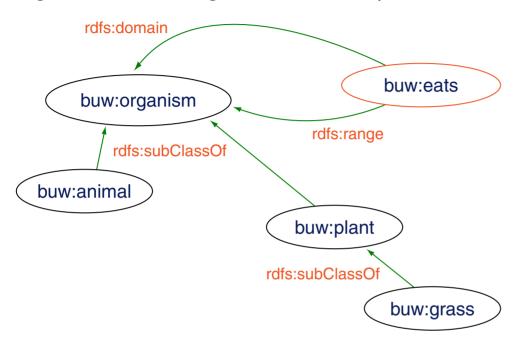
Formulierung in RDF/XML:

Formulierung in Prädikatenlogik (hier als Menge mit qualifizierender Formel):

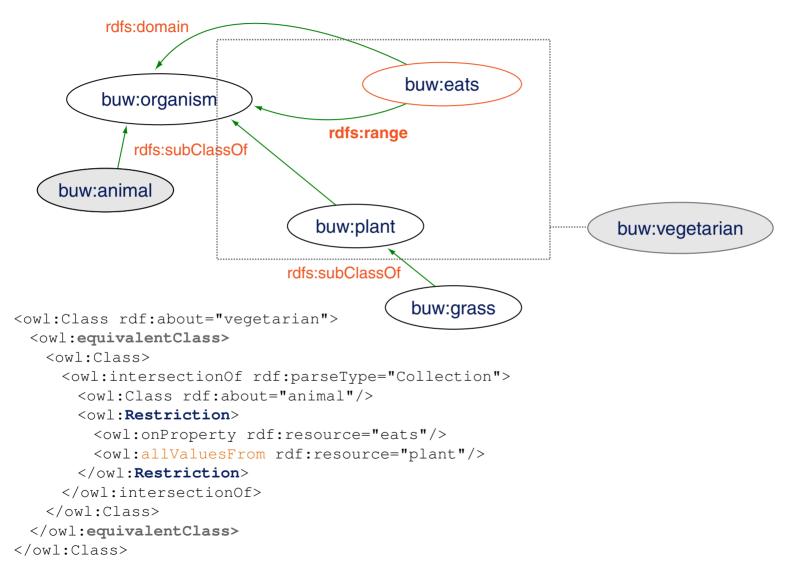
```
\{x \mid Course(x) \land (\forall y : IsTaughtBy(x,y) \rightarrow Lecturer(y))\}
```

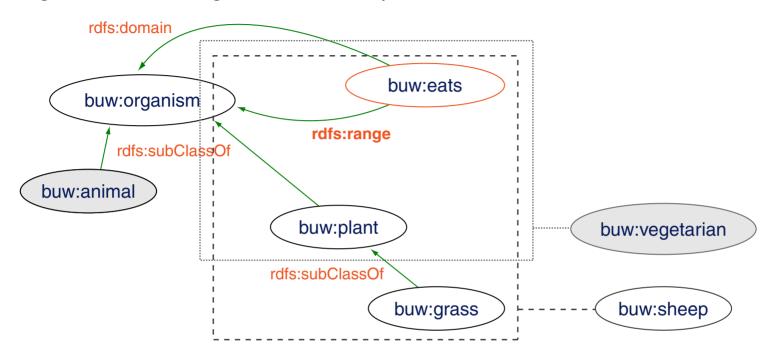
Formulierung in Beschreibungslogik:





```
<owl:ObjectProperty rdf:about="eats">
  <rdfs:domain rdf:resource="organism" />
  <rdfs:range rdf:resource="organism" />
</owl:ObjectProperty>
```

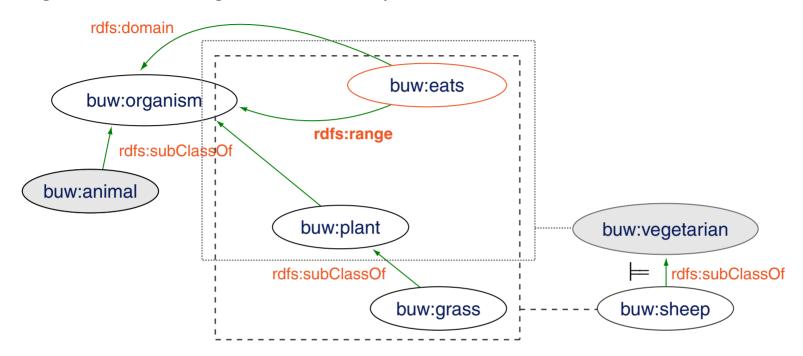




```
<owl:Class rdf:about="sheep">
  <rdfs:subClassOf rdf:resource="animal"/>
        <rdfs:subClassOf>
        <owl:Restriction>
        <owl:onProperty rdf:resource="eats"/>
              <owl:allValuesFrom rdf:resource="grass"/>
              </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>
```

OWL: Konzepte

Range-Einschränkung: Konstruktion spezialisierter Klassen II

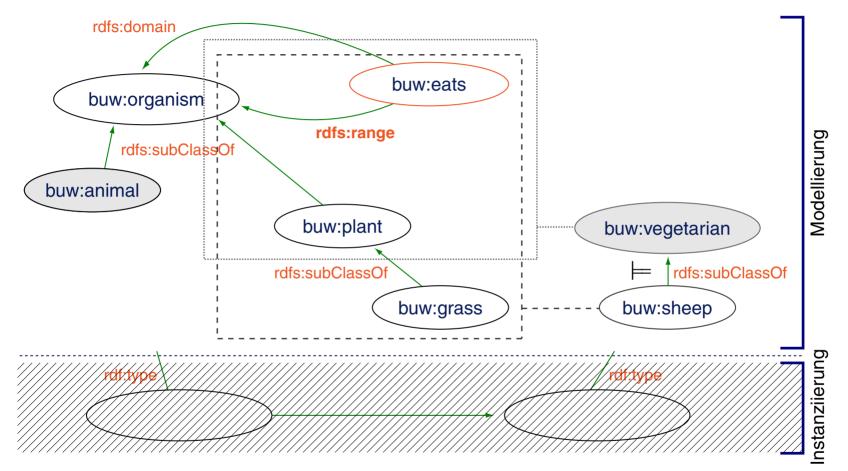


Schlussfolgerung auf der Modellierungsebene.

WT:VIII-207 Semantic Web: Ontology © STEIN 2005-2014

OWL: Konzepte

Range-Einschränkung: Konstruktion spezialisierter Klassen II



Bemerkungen:

- □ Wegen der Open-World-Semantic des Semantic Web kann jeder Anwender Instanzen für die Property buw:eats hinzufügen. Bei den hier konstruierten, spezialisierten Klassen (buw:vegetarian, buw:sheep) kann das zur Folge haben, dass Instanzen (Tiere), die zu einem Zeitpunkt t_1 zur Klasse buw:vegetarian oder buw:sheep gehörten, zu einem späteren Zeitpunkt t_2 nicht mehr dazu gehören, weil sie die owl:allvaluesFrom-Restriktion nicht mehr erfüllen. Es handelt sich um eine nicht-monotone Situation: obwohl das Wissen (= Anzahl der Property-Instanzen) zunimmt, sinkt die Anzahl ableitbarer Schlussfolgerungen (= Anzahl der Instanzen in einer spezialisierten Klasse).
- Mit monotonen Schlussfolgerungsverfahren/-werkzeugen (Protégé, OWL-API) lassen sich nur Schlussfolgerungen ziehen, die immer gültig sind. Die Klassenmitgliedschaft (Type-Property) bei den spezialisierten Klassen buw:vegetarian oder buw:sheep fällt nicht hierunter. Ein nicht-monotoner Reasoner kann mit dieser Siutation umgehen, in dem er gemachte Schlussfolgerungen wieder zurückzieht, falls sie nicht mehr gültig sind. Stichwort: Nicht-monotones Schließen. [vgl. Vorlesung Logik, Lettmann/Stein 1996-2013]
- □ Auf der Modellierungsebene ist die Problematik der Nicht-Monotonie nicht gegeben, weil die Ontologie vorliegt und über Eigenschaften ihrer Klassen zeitunabhängig geschlussfolgert werden kann.
- Es existieren weitere Möglichkeiten, um Restriktionen zu spezifizieren:
 - owl:someValuesFrom
 - owl:hasValue
 - owl:cardinality, owl:minCardinality, owl:maxCardinality

OWL: Konzepte

Header- und Metadaten für OWL-Ontologien

OWL-Dateien sind RDF-Dokumente und werden als OWL-Ontologien bezeichnet.

Üblicher RDF-Dokument-Header:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<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/2000/10/XMLSchema#">
```

OWL: Konzepte

Header- und Metadaten für OWL-Ontologien

OWL-Dateien sind RDF-Dokumente und werden als OWL-Ontologien bezeichnet.

Üblicher RDF-Dokument-Header:

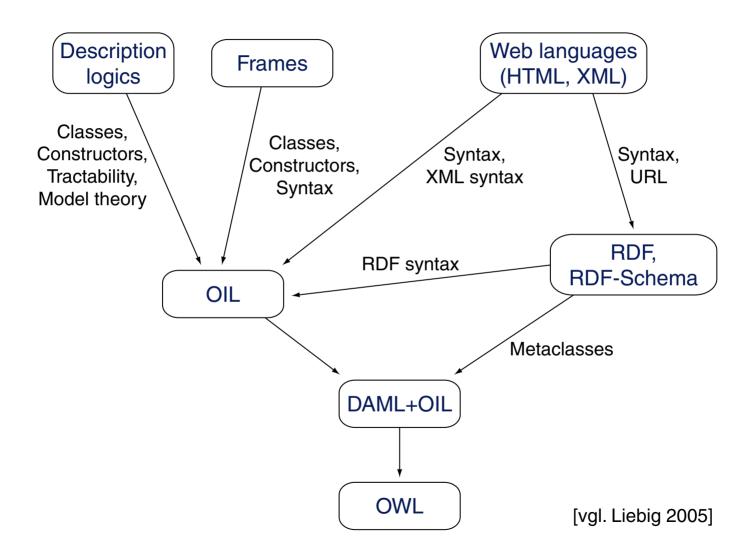
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF
  xmlns:owl ="http://www.w3.org/2002/07/owl#"
  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#">
```

Ontologie-Metadaten:

```
<owl:Ontology rdf:about="">
    <rdfs:comment>An example OWL ontology</rdfs:comment>
    <owl:priorVersion
        rdf:resource="http://buw-ontologies-old/teaching-ontology"/>
        <owl:imports rdf:resource="http://buw-ontologies/person-ontology"/>
        <rdfs:label>Teaching Ontology</rdfs:label>
</owl:Ontology>
```

WT:VIII-211 Semantic Web: Ontology © STEIN 2005-2014

Sprachliche Wurzeln von OWL



Ausdruckstärke von OWL-Varianten

OWL lässt sich in der Prädikatenlogik (erster Stufe, PL1) axiomatisieren.

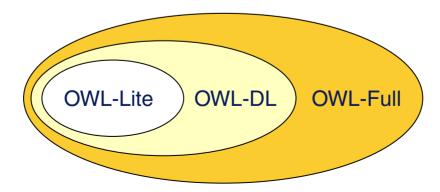
Warum sollte man diese nicht zur Beschreibung von Ontologien einsetzen?

Ausdruckstärke von OWL-Varianten

OWL lässt sich in der Prädikatenlogik (erster Stufe, PL1) axiomatisieren.

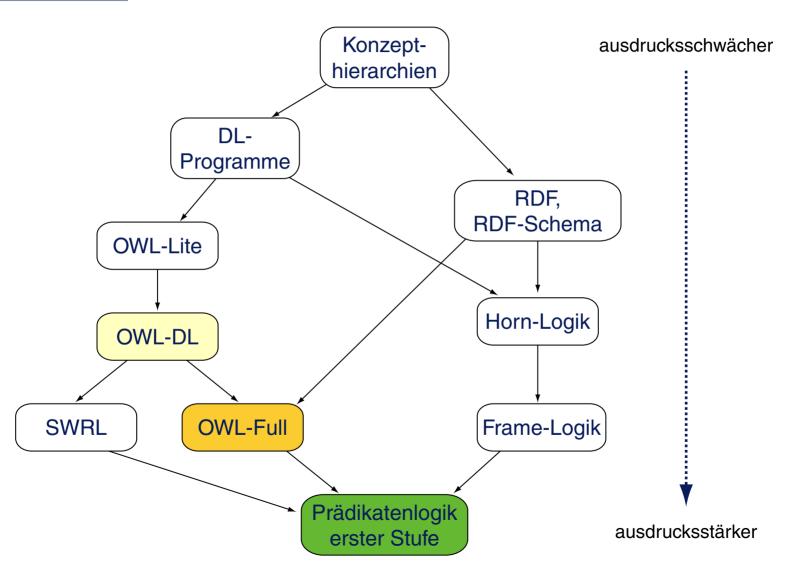
Warum sollte man diese nicht zur Beschreibung von Ontologien einsetzen?

- Die Ausdruckstärke der Prädikatenlogik kann die Modellierung unhandlich machen und Anwender überfordern.
- Die M\u00e4chtigkeit und die Freiheiten der Pr\u00e4dikatenlogik erschweren den Abgleich zwischen und den Austausch von Modellen.
- □ Die Prädikatenlogik ist beweistheoretisch komplex.
- Suche nach einer geeigneten Teilmenge der Prädikatenlogik.



WT:VIII-214 Semantic Web: Ontology © STEIN 2005-2014

Ausdruckstärke von OWL-Varianten (Fortsetzung)



Ausdruckstärke von OWL-Varianten (Fortsetzung)

OWL-Lite ⊂ OWL-DL ⊂ OWL-Full ⊂ Prädikatenlogik

OWL-DL:

- keine Reification
- ist entscheidbar
- entspricht der Beschreibungslogik SHOIN(D)

OWL-Full:

- alle Sprachkonstrukte sind verwendbar, solange gültiges RDF vorliegt
- \square OWL-Full = OWL-DL \cup RDFS bzw. RDFS \subseteq OWL-Full
- erlaubt Reification
- ist nicht entscheidbar
- gilt als "unschöne" Teilmenge der Prädikatenlogik

Bemerkungen:

- OWL-Full ist für Anwendungen, die maximale Ausdrucksstärke und die syntaktische Freiheit von RDF/RDFS erfordern.
- OWL-Full schafft die Möglichkeit, innerhalb einer Ontologie die Bedeutung der vordefinierten RDF-, RDFS- und OWL-Vokabulare zu erweitern.
- RDF/RDFS ist im Allgemeinen in OWL-Full, und schon nicht mehr in OWL-DL.
- Es ist nicht zu erwarten, dass eine schlussfolgernde Anwendung jedes Konzept von OWL-Full unterstützen kann.

WT:VIII-217 Semantic Web: Ontology © STEIN 2005-2014

Ausdruckstärke von OWL-Varianten: OWL-DL

Um in der Komplexitätklasse von Beschreibungslogiken zu bleiben, sind folgende Bedingungen einzuhalten [vgl. Antoniou/Harmelen 2004]:

- Partionierung des Vokabulars.
 Eine Ressource ist entweder eine Klasse oder ein Individuum oder ein Datenwert etc.
- 2. Explizite Typangaben.

```
</owl:Class rdf:ID="C2"/>
<owl:Class rdf:ID="C1">
     <rdfs:subClassOf rdf:resource="#C2"/>
</owl:Class>
```

- 3. Wegen Punkt 1. sind folgende Elemente nicht als Datentypen erlaubt:
 - ☐ owl:inverseOf
 - □ owl:FunctionalProperty
 - □ owl:InverseFunctionalProperty
 - ☐ owl:SymmetricProperty
- 4. Kardinalitätsangaben sind in transitiven Properties nicht erlaubt.
- 5. Eingeschränkte Verwendung anonymer Klassen.

Bemerkungen:

- □ zu (2) explizite Typangaben:
 - Obwohl aus <rdfs:subClassOf rdf:resource="#C2"/> folgerbar ist, dass C2 eine Klasse ist, muss für C2 z. B. mit </owl:Class rdf:ID="C2"/> der Typ "Klasse" explizit spezifiziert werden.
- □ OWL-DL ist eine Sprache mit hoher Ausdrucksmächtigkeit bei gleichzeitiger Entscheidbarkeit.
- OWL-DL wurde u. a. auch deswegen spezifiziert, um das existierende Geschäftsfeld und die Befürworter der Beschreibungslogik zu bedienen.

WT:VIII-219 Semantic Web: Ontology © STEIN 2005-2014

Ausdruckstärke von OWL-Varianten: OWL-Lite

Folgende Sprachkonstrukte sind nur eingeschränkt verwendbar:

- ☐ owl:intersectionOf
- ☐ owl:minCardinality
- ☐ owl:maxCardinality
- owl:cardinality

Folgende Sprachkonstrukte können nicht verwendet werden:

- ☐ owl:unionOf
- owl:complementOf
- owl:oneOf
- owl:hasValue
- ☐ owl:disjointWith

Bemerkungen:

- OWL-Lite ist für Anwendungen, in denen einfache Klassenhierarchien, Taxonomien und leicht axiomatisierbare Ontologien modelliert werden sollen.
- OWL-Lite stellt eine einfach zu implementierende Untermenge von OWL bereit.
- Einschränkung von Sprachkonstrukten: Argumente der Schnittmengenbildung dürfen nur Klassen und nicht-geschachtelte Restriktionen sein; als Kardinalitätsangaben sind nur die Werte 0 und 1 zugelassen.

WT:VIII-221 Semantic Web: Ontology © STEIN 2005-2014

Ausdruckstärke von OWL-Varianten (Fortsetzung)

Worst-Case-Komplexitäten hinsichtlich Inferenz, Erfüllbarkeit, Widersprüchlichkeit einer Formel oder Wissensbasis [Hitzler 2005]:

OWL-Variante	Komplexität
OWL-Lite	Exptime
OWL-DL	NExptime
OWL-Full	unentscheidbar

Insbesondere gilt:

- 1. Jede OWL-Lite-Ontologie ist auch eine OWL-DL-Ontologie. Jede OWL-DL-Ontologie ist auch eine OWL-Full-Ontologie.
- 2. Jede OWL-Lite-Schlussfolgerung ist auch eine OWL-DL-Schlussfolgerung. Jede OWL-DL-Schlussfolgerung ist auch eine OWL-Full-Schlussfolgerung.

Beschreibungslogik (Description Logics, DL)

Beschreibungslogik ist ein Sammelbegriff für verschiedene logikbasierte Sprachen, die mit dem Schwerpunkten "Wissensrepräsentation" und "Konzeptualisierung" eines Gegenstandsbereichs entwickelt wurden.

Beschreibungslogik (Description Logics, DL)

Beschreibungslogik ist ein Sammelbegriff für verschiedene logikbasierte Sprachen, die mit dem Schwerpunkten "Wissensrepräsentation" und "Konzeptualisierung" eines Gegenstandsbereichs entwickelt wurden.

Charakteristika:

- Teilmenge der Prädikatenlogik erster Stufe (PL1), meist entscheidbar
- Ausdrucksstärke orientiert an praktischer Verwendbarkeit
- □ Ursprung in semantischen Netzen, u. a. KL-ONE
- Definition eines Vokabulars, bestehend aus Konzepten und Rollen.
 - Konzepte bezeichnen Mengen von Individuen; sie entspechen einstelligen Prädikaten.
 - Rollen bezeichnen Beziehungen zwischen je zwei Individuen; sie entsprechen zweistelligen Prädikaten.
- Konstruktoren für komplexe Konzepte und Rollen auf Basis von atomaren Konzepten und Rollen

WT:VIII-224 Semantic Web: Ontology © STEIN 2005-2014

Beschreibungslogik

- ALC ist die kleinste DL, die aussagenlogisch abgeschlossen ist. Die Beschreibungslogik hinter OWL-DL heißt SHOIN(D).
- □ Konstruktoren sind \land , \lor und \neg , geschrieben als \sqcap , \sqcup , und \neg .
- □ Mittels der Quantoren ∀ und ∃ werden Rollen eingeschränkt:

Beispiel: ∀hasChild.Female

- = diejenigen x, bei denen alle Kinder weiblich sind = die nur Töchter haben.
- Kardinalitätsangaben für Rollen und Konzepte
- Datentypen:

Beispiel: hasAge.(≥21)

□ inverse Rollen:

Beispiel: hasChild[−] ≡ hasParent

transitive Rollen:

Beispiel: hasAncestor*(descendant)

Rollenkomposition:

Beispiel: hasParent.hasBrother(uncle)

Beschreibungslogik: Wissensbasis

Die Formeln einer DL-Wissensbasis werden oft in zwei Mengen aufgeteilt:

TBox Axiome, um Prinzipien des Gegenstandsbereichs zu beschreiben:

Lecturer

☐ TeachingStaff

ABox Axiome, um Ausprägungen des Gegenstandsbereichs zu beschreiben:

Lecturer(Benno-Stein)

Beschreibungslogik: Wissensbasis

Die Formeln einer DL-Wissensbasis werden oft in zwei Mengen aufgeteilt:

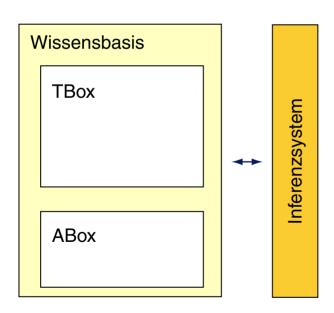
TBox Axiome, um Prinzipien des Gegenstandsbereichs zu beschreiben:

Lecturer

☐ TeachingStaff

ABox Axiome, um Ausprägungen des Gegenstandsbereichs zu beschreiben:

Lecturer(Benno-Stein)



Beschreibungslogik: Wissensbasis

Die Formeln einer DL-Wissensbasis werden oft in zwei Mengen aufgeteilt:

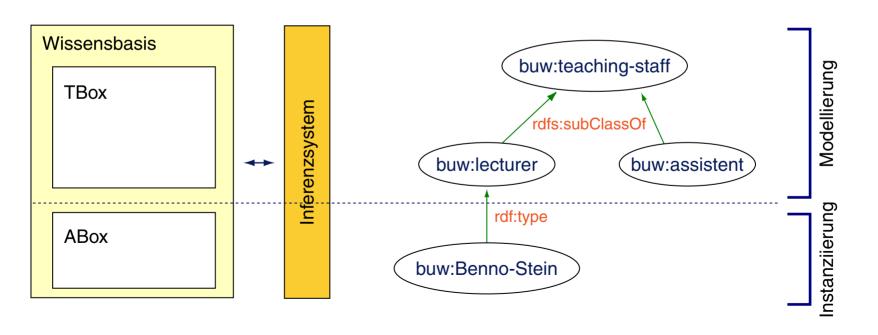
TBox Axiome, um Prinzipien des Gegenstandsbereichs zu beschreiben:

Lecturer

☐ TeachingStaff

ABox Axiome, um Ausprägungen des Gegenstandsbereichs zu beschreiben:

Lecturer(Benno-Stein)



Bemerkungen:

- □ Die TBox beinhaltet die terminologische Beschreibung (terminology) also die Axiomatisierung bzw. das Modell des Gegenstandsbereichs; die ABox beinhaltet konkrete Behauptungen (assertions).
- Aus pr\u00e4dikatenlogischer Sicht entspricht die Unterscheidung einer TBox und einer ABox der Aufteilung einer Formelmenge in Formeln mit und ohne Variablen. Sie ist aus logischen Gr\u00fcnden nicht notwendig.

WT:VIII-229 Semantic Web: Ontology © STEIN 2005-2014

OWL-Name	DL-Syntax	PL-Syntax	Beispiel in DL-Syntax
intersectionOF	$C_1 \sqcap \ldots \sqcap C_n$	$C_1(x) \wedge \ldots \wedge C_n(x)$	Human ⊓ Male

Beschreibungslogik: wichtige Sprachelemente

OWL-Name	DL-Syntax	PL-Syntax	Beispiel in DL-Syntax
intersectionOF	$C_1 \sqcap \ldots \sqcap C_n$	$C_1(x) \wedge \ldots \wedge C_n(x)$	Human ⊓ Male
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	$C_1(x) \vee \ldots \vee C_n(x)$	Lecturer ⊔ Assistent

WT:VIII-231 Semantic Web: Ontology © STEIN 2005-2014

OWL-Name	DL-Syntax	PL-Syntax	Beispiel in DL-Syntax
intersectionOF	$C_1 \sqcap \ldots \sqcap C_n$	$C_1(x) \wedge \ldots \wedge C_n(x)$	Human ⊓ Male
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	$C_1(x) \vee \ldots \vee C_n(x)$	Lecturer ⊔ Assistent
complementOf	$\neg C$	$\neg C(x)$	\neg Female
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	$x = x_1 \lor \ldots \lor x = x_n$	{Alice} ⊔ {Bob}

OWL-Name	DL-Syntax	PL-Syntax	Beispiel in DL-Syntax
intersectionOF	$C_1 \sqcap \ldots \sqcap C_n$	$C_1(x) \wedge \ldots \wedge C_n(x)$	Human ⊓ Male
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	$C_1(x) \vee \ldots \vee C_n(x)$	Lecturer ⊔ Assistent
complementOf	$\neg C$	$\neg C(x)$	\neg Female
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	$x = x_1 \lor \ldots \lor x = x_n$	{Alice} ⊔ {Bob}
allValuesFrom	$\forall P.C$	$\forall y: P(x,y) \to C(y)$	∀hasChild.Male
someValuesFrom	$\exists P.C$	$\exists y: P(x,y) \land C(y)$	∃hasChild.Male

intersectionOF $C_1\sqcap\ldots\sqcap C_n$ $C_1(x)\wedge\ldots\wedge C_n(x)$ Human \sqcap Male				
$\begin{array}{llllllllllllllllllllllllllllllllllll$	OWL-Name	DL-Syntax	PL-Syntax	Beispiel in DL-Syntax
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	intersectionOF	$C_1 \sqcap \ldots \sqcap C_n$	$C_1(x) \wedge \ldots \wedge C_n(x)$	Human ⊓ Male
oneOf $\{x_1\} \sqcup \ldots \sqcup \{x_n\} \qquad x = x_1 \vee \ldots \vee x = x_n \qquad \{\text{Alice}\} \sqcup \{\text{Bob}\}$ allValuesFrom $\forall P.C \qquad \forall y: P(x,y) \to C(y) \qquad \forall \text{hasChild.Male}$ someValuesFrom $\exists P.C \qquad \exists y: P(x,y) \wedge C(y) \qquad \exists \text{hasChild.Male}$ maxCardinality $\leq nP \qquad \exists \leq^n y: P(x,y) \qquad \leq n \text{hasChild}$	unionOf	$C_1 \sqcup \ldots \sqcup C_n$	$C_1(x) \vee \ldots \vee C_n(x)$	Lecturer ⊔ Assistent
allValuesFrom $\forall P.C$ $\forall y: P(x,y) \to C(y)$ $\forall \text{hasChild.Male}$ someValuesFrom $\exists P.C$ $\exists y: P(x,y) \land C(y)$ $\exists \text{hasChild.Male}$ maxCardinality $\leq nP$ $\exists^{\leq n}y: P(x,y)$ $\leq n\text{hasChild}$	complementOf	$\neg C$	$\neg C(x)$	¬ Female
	oneOf	$\{x_1\}\sqcup\ldots\sqcup\{x_n\}$	$x = x_1 \vee \ldots \vee x = x_n$	{Alice} ⊔ {Bob}
$\mbox{maxCardinality} \qquad \leq nP \qquad \qquad \exists^{\leq n}y: P(x,y) \qquad \qquad \leq n \mbox{hasChild}$	allValuesFrom	$\forall P.C$	$\forall y: P(x,y) \to C(y)$	∀hasChild.Male
	someValuesFrom	$\exists P.C$	$\exists y: P(x,y) \land C(y)$	∃hasChild.Male
minCardinality $\geq nP$ $\exists^{\geq n}y: P(x,y)$ $\geq nhasChild$	maxCardinality	$\leq nP$	$\exists^{\leq n}y:P(x,y)$	$\leq n$ hasChild
	minCardinality	$\geq nP$	$\exists^{\geq n}y:P(x,y)$	$\geq n$ hasChild

OWL-Name	DL-Syntax	PL-Syntax	Beispiel in DL-Syntax
intersectionOF	$C_1 \sqcap \ldots \sqcap C_n$	$C_1(x) \wedge \ldots \wedge C_n(x)$	Human ⊓ Male
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	$C_1(x) \vee \ldots \vee C_n(x)$	Lecturer ⊔ Assistent
complementOf	$\neg C$	$\neg C(x)$	\neg Female
oneOf	$\{x_1\}\sqcup\ldots\sqcup\{x_n\}$	$x = x_1 \vee \ldots \vee x = x_n$	{Alice} ⊔ {Bob}
allValuesFrom	$\forall P.C$	$\forall y: P(x,y) \to C(y)$	∀hasChild.Male
someValuesFrom	$\exists P.C$	$\exists y: P(x,y) \land C(y)$	∃hasChild.Male
maxCardinality	$\leq nP$	$\exists^{\leq n}y:P(x,y)$	$\leq n$ hasChild
minCardinality	$\geq nP$	$\exists^{\geq n}y:P(x,y)$	$\geq n$ hasChild
subClassOf	$C_1 \sqsubseteq C_2$	$\forall x: C_1(x) \to C_2(x)$	Human ⊑ Animal ⊓ Biped
equivalentClass	$C_1 \equiv C_2$	$\forall x: C_1(x) \leftrightarrow C_2(x)$	Man ≡ Human ⊓ Male

Bemerkungen:

$$\{x \mid \mathsf{wenn}\ (x,y) \in P \mathsf{dann}\ y \in C\}$$

$$\{x \mid (x,y) \in P \text{ und } y \in C\}$$

Beschreibungslogik: Inferenzprobleme [vgl. Hitzler 2005]

- 1. Wissensbasiskonsistenz. Ist Wissensbasis KB widerspruchsfrei? $KB \models \mathsf{false}$?
- 2. Klassenkonsistenz. Ist Klasse C leer? $C \equiv \perp$?
- 3. Subsumption. Ist C Unterklasse von D? $C \sqsubseteq D$?
- 4. Klassenäquivalenz. Sind die Klassen C und D identisch? $C \equiv D$?
- 5. Klassendisjunktheit. Sind die Klassen C und D disjunkt? $C \sqcup D = \bot$?
- 6. Klassenzugehörigkeit. Ist Individuum a aus Klasse C? C(a) ?

Gegenüberstellung von Sprachen zur Wissensformulierung

Formulierung in natürlicher Sprache:

"Dozenten bilden einen Teil des Lehrpersonals."

Formulierung in Prädikatenlogik:

 $\forall x : Lecturer(x) \rightarrow TeachingStaff(x)$

Formulierung in Beschreibungslogik:

Lecturer

☐ TeachingStaff

Formulierung in RDF/XML:

```
<owl:Class rdf:about="http://www.buw.de/lecturer">
  <rdfs:subClassOf rdf:resource="http://www.buw.de/teaching-staff"/>
  </owl:Class>
```

Quellen zum Nachlernen und Nachschlagen im Web

- □ W3C OWL-Übersichtseite. www.w3.org/2004/OWL
- D.L. McGuinness, F. van Harmelen, Eds. OWL Web Ontology Language Overview. W3C Recommendation. www.w3.org/TR/owl-features
- Smith, Welty, McGuinness, Eds. OWL Web Ontology Language Guide. W3C Recommendation.
 www.w3.org/TR/owl-guide,
 www.semaweb.org/dokumente/w3/TR/2004/REC-owl-guide-20040210-DE.html
- □ Dean, Schreiber, Eds. *OWL Web Ontology Language Reference*. W3C Recommendation. www.w3.org/TR/owl-ref
- Patel-Schneider, Hayes, Horrocks, Eds. OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation.
 www.w3.org/TR/owl-semantics

WT:VIII-239 Semantic Web: Ontology © STEIN 2005-2014

Quellen zum Nachlernen und Nachschlagen im Web (Fortsetzung)

- □ DAML Homepage.www.daml.org
- OIL Homepage.www.ontoknowledge.org/oil
- □ HP Labs Semantic Web Research. www.hpl.hp.com/semweb
- WonderWeb Homepage.wonderweb.semanticweb.org

WT:VIII-240 Semantic Web: Ontology © STEIN 2005-2014

Quellen zum Nachlernen und Nachschlagen im Web (Fortsetzung)

Anwendung und Implementierung:

- Jena. Semantic Web Framework for Java. jena.sourceforge.net
- Protégé Ontology Editor. protege.stanford.edu
- BBN Ontology Validator. owl.bbn.com/validator
- WonderWeb Ontology Validator.phoebus.cs.man.ac.uk:9999/OWL/Validator
- □ Knublauch, Oberle, Tetlow, Wallace, Eds. *A Semantic Web Primer for Object-Oriented Software Developers.* Suggested W3C Note. www.w3.org/2001/sw/BestPractices/SE/ODSD

WT:VIII-241 Semantic Web: Ontology © STEIN 2005-2014

Quellen zum Nachlernen und Nachschlagen im Web (Fortsetzung)

Beschreibungslogik:

- □ F. Baader, W. Nutt. Basic Description Logics. www.inf.unibz.it/~franconi/dl/course/dlhb/dlhb-02.pdf
- □ E. Franconi. Tutorials für Aussagen-, Prädikaten- und Beschreibungslogik. www.inf.unibz.it/~franconi/dl/course

WT:VIII-242 Semantic Web: Ontology © STEIN 2005-2014