Ontologien in OWL und Medizin

FH Brandenburg

Dipl.-Inform. I. Boersch
FB Informatik und Medien

Next

- Ontologie und die beiden Visionen des ‚Semantic Web'
- RDF, RDFS
- Web Ontology Language OWL
  - OWL lesen – Sprachkonstrukte
  - Beispiel einer Ontologie: FOAF
- Von OWL zu OWL2
- Anfragen an Ontologien
  - OWL-Reasoner sind zurückhaltend (Open World Assumption)
- Ontologien in den Life Sciences
  - GALEN, SNOMED
- Wertung DL

- Übung: Mit dem Tool Protege eine Pizza belegen

Was ist eine Ontologie?

PR

Eine Ontologie ist …

eine **formale**, explizite Spezifikation

einer **gemeinsamen**

Konzeptualisierung.

maschinenverständlich, geeignet zur Inferenz, Dokument

Personen einigen sich über Begrifflichkeiten, kontrolliertes Vokabular

Konzepte und Beziehungen, nicht mehr

- Eine Möglichkeit zur Repräsentation von Ontologien sind **Terminologien**, also Mengen von Konzept- und Rollendefinitionen in beschreibungslogischen Ausdrücken
- Ontologiesprachen OWL bzw. OWL 2 sind Grundlage des Semantic Web
  - offene Standards des w3C

PR = PRÜFUNGSRELEVANT

Alles andere: wichtig für die Arbeit als (Medizin-) Informatiker

Wozu Ontologien?

Die zwei Visionen des Semantic Web

Terminologien in Form des Semantic Web bieten als ‚Mehrwert':

1. **Vernetzung** von Wissen mit Hilfe eindeutiger Begriffe (IRI*, gemeinsame OWL-Ontologie als kontrolliertes Vokabular)
2. **Inferenz** in semantischen Netzen (maschinelle Inferenz!)

PR

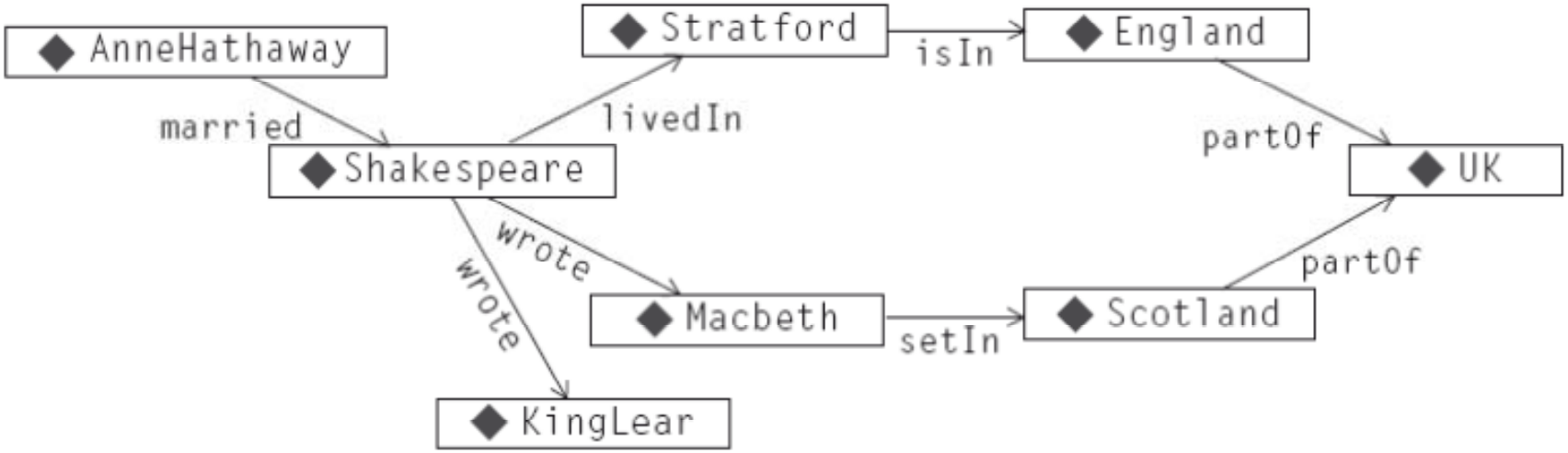
- Sie möchten etwas beschreiben? Dann sollten Sie sich fragen:
  - „Gibt es dafür eine fertige Ontologie?"

Beispiel: The Friend of a Friend (FOAF) project

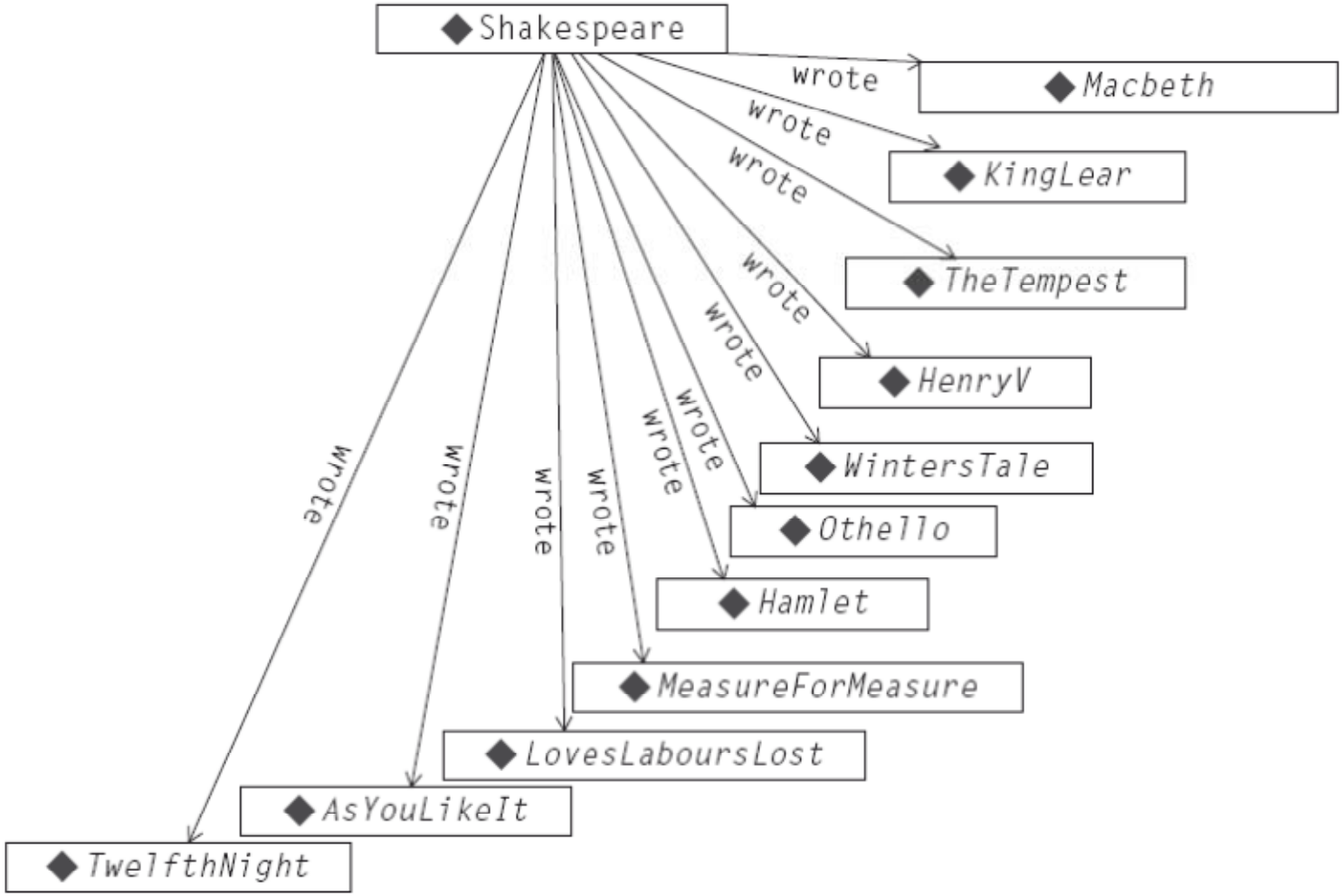
- describing people, the links between them and the things they create and do
- http://xmlns.com/foaf/spec/index.rdf

* Internationalized Resource Identifier

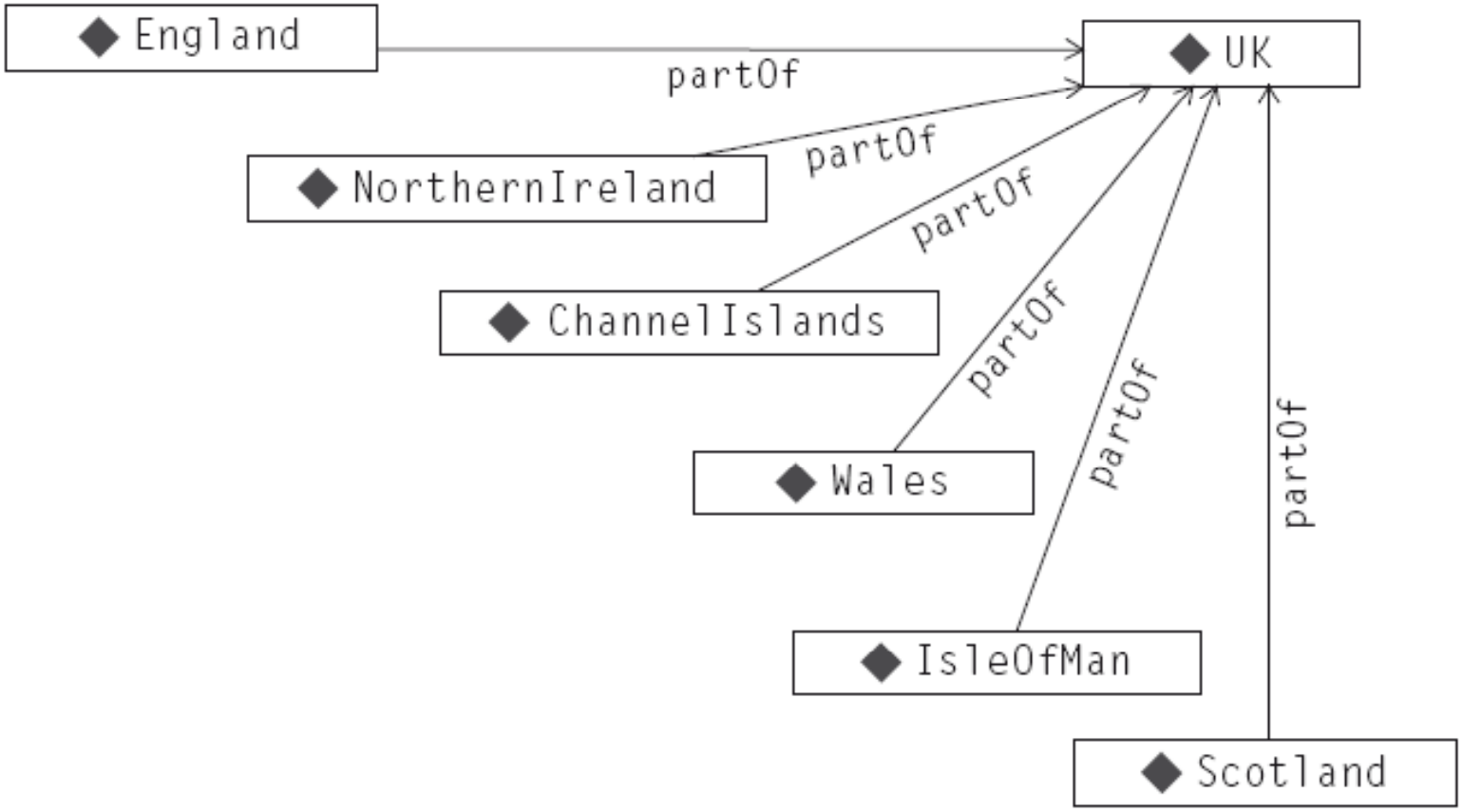
Die 1. Vision - Vernetzung



Die 1. Vision - Vernetzung



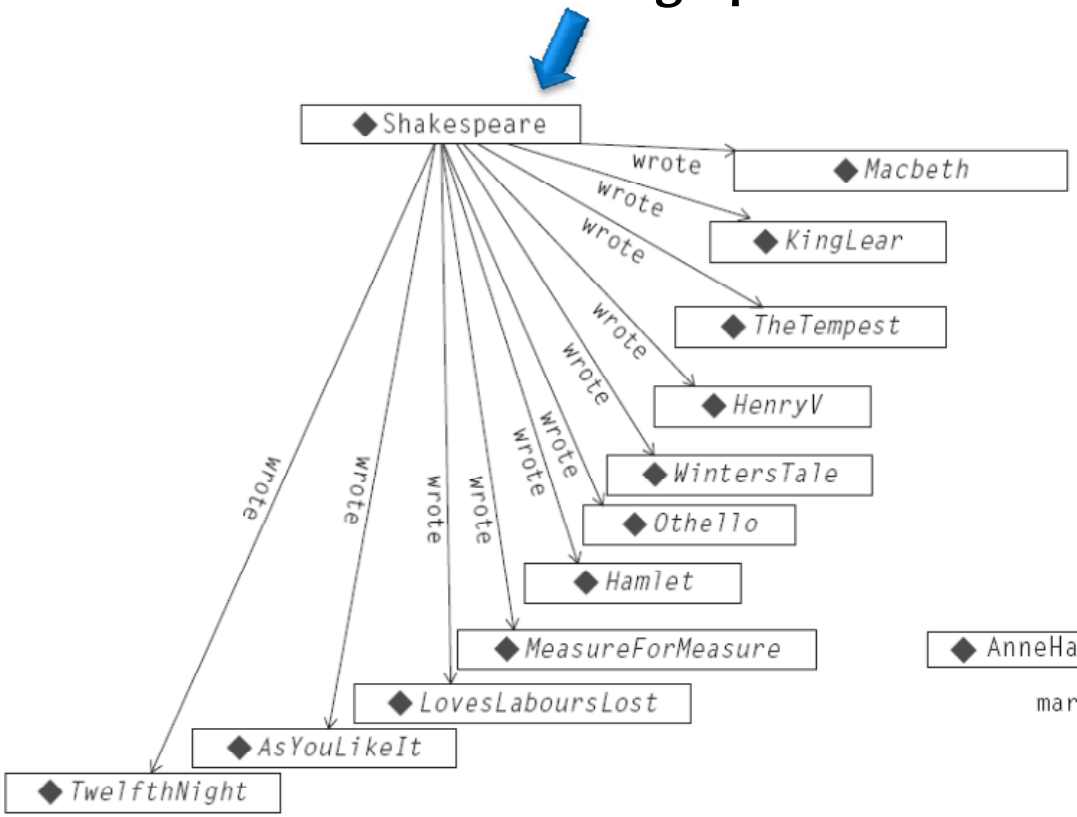
Die 1. Vision - Vernetzung

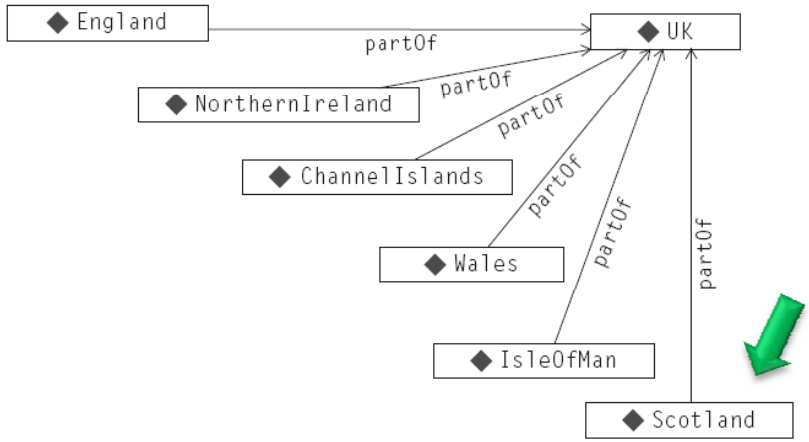


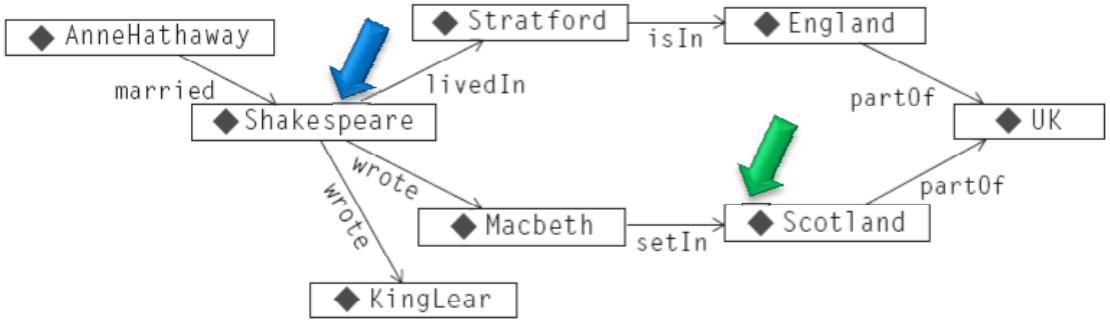
3 Wissensbasen

Konzeptäquivalenz?

When is a node in one graph the same node as a node in another graph?

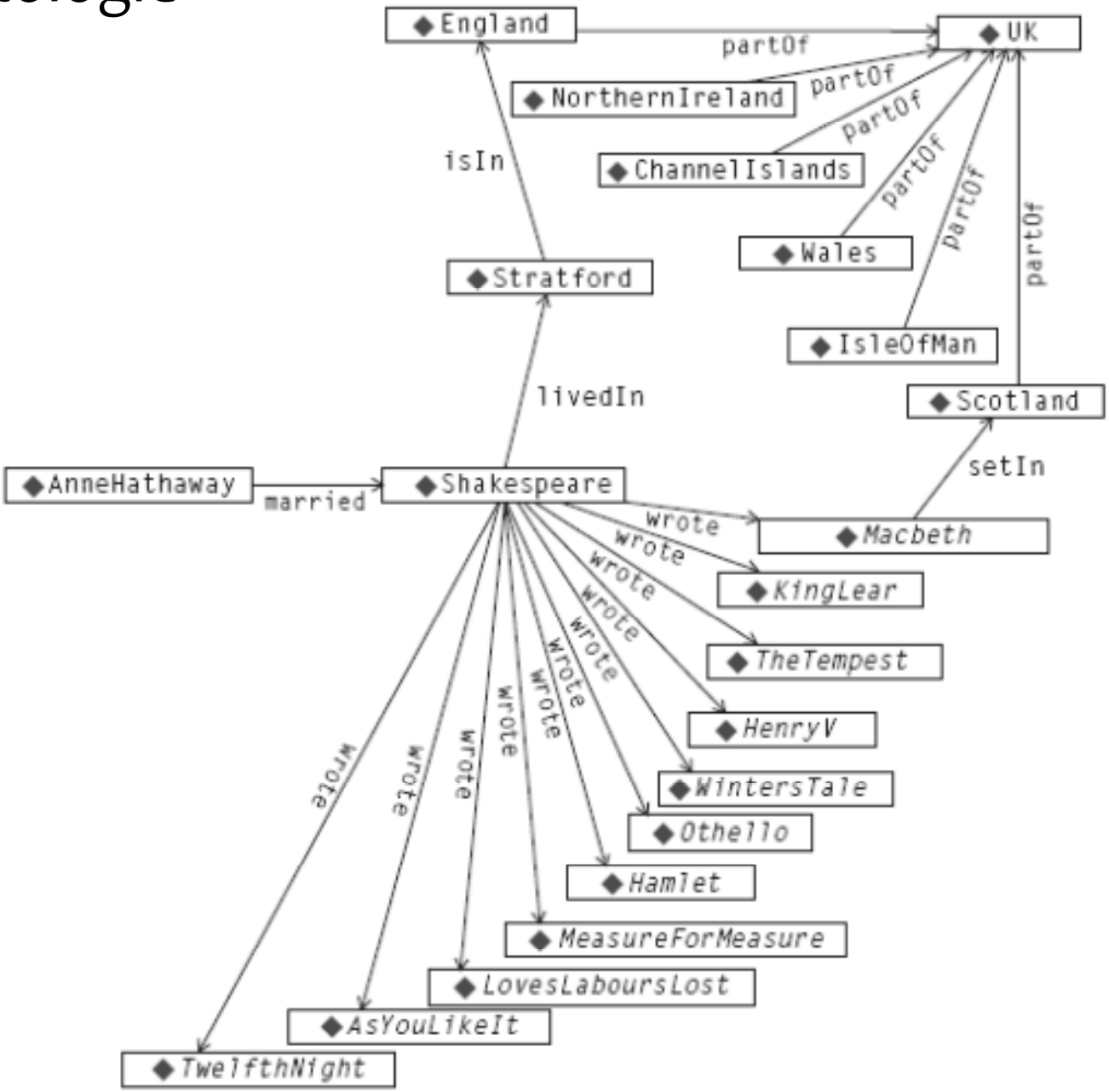






Die gemeinsame Ontologie Same URI

- A node from one graph is merged with a node from another graph—exactly, if they have the **same URI.**
- Verknüpfung von Wissensbasen
- Das leisten bereits semantische Netze, beschrieben in **RDF und RDFS**



Aktuelles Beispiel 4.6.2011: schema.org

- Kontrolliertes Vokabular
- Google + Microsoft + Yahoo
- Relativ kleine Monohierarchie von Konzepten
- Einfaches semantisches Netz in RDFS, d.h. mit beschränkter Semantik

Konzept „School“

Thing > Organization > EducationalOrganization > School

A school.

Property	Expected Type	Description
Properties from Thing		
description	Text	A short description of the item.
image	URL	URL of an image of the item.
name	Text	The name of the item.
url	Text	URL of the item.
Properties from Organization		
address	PostalAddress	Physical address of the item.
aggregateRating	AggregateRating	The overall rating, based on a collection of reviews or ratings, of the item.
contactPoints	ContactPoint	A contact point for a person or organization.
email	Text	Email address.
employees	Person	People working for this organization.
events	Event	The events held at this place or organization.
[illegible]	Text	The fax number.

Geerbte Relation „adress“

School is_a EducationalOrganization

http://schema.rdfs.org/all.rdf

```
– <rdf:Description rdf:about="http://schema.org/Preschool">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:label xml:lang="en">Preschool</rdfs:label>
    <rdfs:comment xml:lang="en">A preschool.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="http://schema.org/EducationalOrganization"/>
– <rdf:Description rdf:about="http://schema.org/School">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:label xml:lang="en">School</rdfs:label>
    <rdfs:comment xml:lang="en">A school.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="http://schema.org/EducationalOrganization"/>
    <rdfs:isDefinedBy rdf:resource="http://schema.org/School"/>
  </rdf:Description>
    <rdfs:isDefinedBy rdf:resource="http://schema.org/School"/>
  </rdf:Description>
– <rdf:Description rdf:about="http://schema.org/GovernmentOrganization">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:label xml:lang="en">Government Organization</rdfs:label>
    <rdfs:comment xml:lang="en">A governmental organization or agency.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="http://schema.org/Organization"/>
    <rdfs:isDefinedBy rdf:resource="http://schema.org/GovernmentOrganization"/>
  </rdf:Description>
```

School ist ein Konzept. Das Konzept heißt im Englischen ‚School', eine Kurzbeschreibung ist ‚A School', Webseite dazu http://schema.org/School (siehe vorige Folie)

School ist Subklasse von EO – eine spezielle Relation

Aktuelles Beispiel 4.6.2011: schema.org

Date
Number
- Float
- Integer

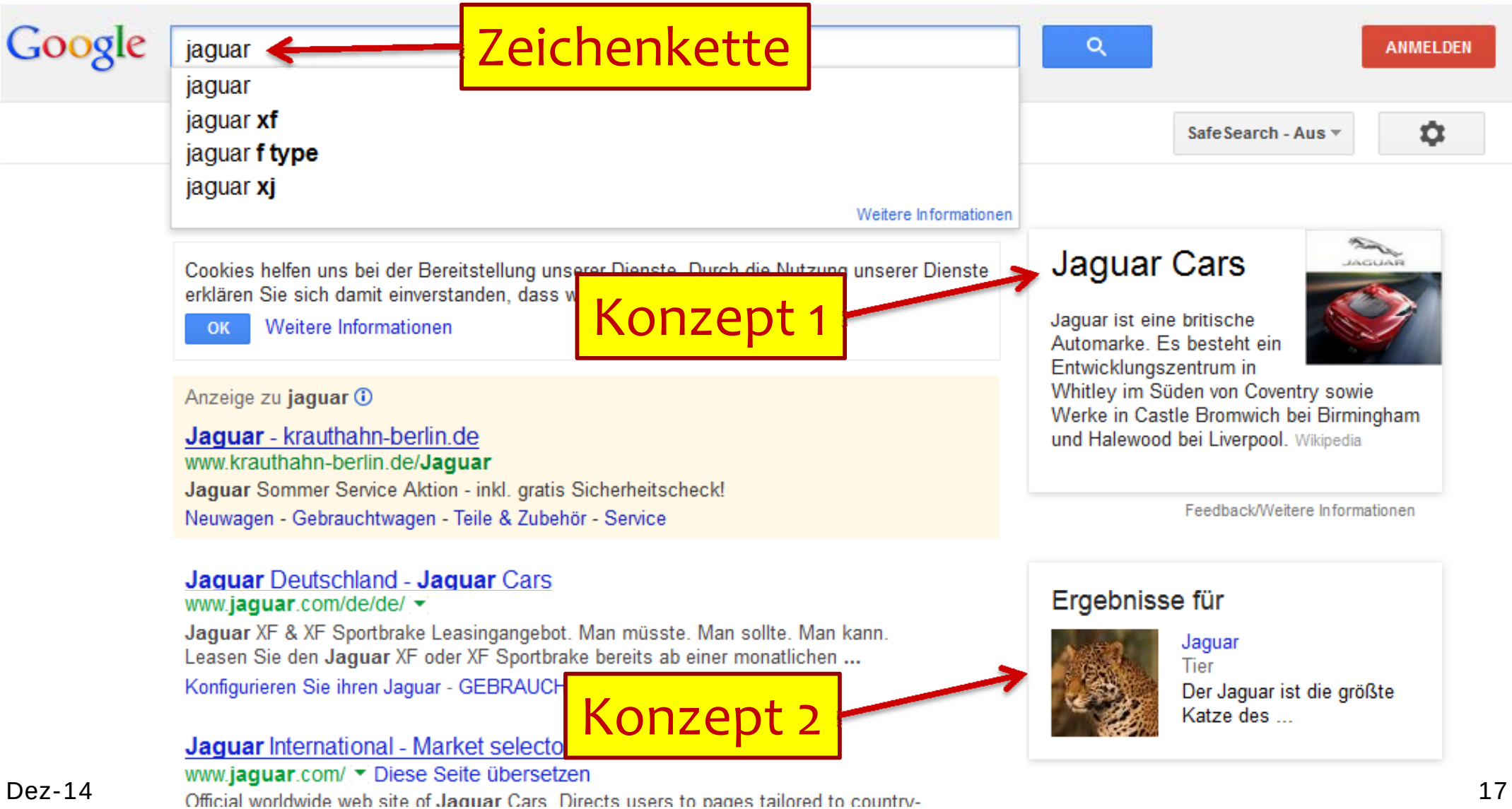
Text
- URL

Thing: description, image, name, url
- CreativeWork: about, accountablePerson, aggregateRating, alternativeHeadlin
  contentRating, contributor, copyrightHolder, copyrightYear, creator, dateC
  headline, inLanguage, interactionCount, isFamilyFriendly, keywords, ment
  thumbnailUrl, version, video
  - Article: articleBody, articleSection, wordCount
    - BlogPosting
    - NewsArticle: dateline, printColumn, printEdition, printPage, pri
    - ScholarlyArticle
  - Blog: blogPosts
  - Book: bookEdition, bookFormat, illustrator, isbn, numberOfPages
  - ItemList: itemListElement, itemListOrder
  - Map
  - MediaObject: associatedArticle, bitrate, contentSize, contentURL, duration, embedURL, encodesCreativeWork, encodingFormat, expires, height, interactionCount, offers, playerType, regionsAllowed, requiresSubscription, uploadDate, width
    - AudioObject: transcript
    - ImageObject: caption, exifData, representativeOfPage, thumbnail
    - MusicVideoObject
    - VideoObject: caption, productionCompany, thumbnail, transcript, videoFrameSize, videoQuality
  - Movie: actors, director, duration, musicBy, producer, productionCompany, trailer
  - MusicPlaylist: numTracks, tracks
    - MusicAlbum: byArtist
  - MusicRecording: byArtist, duration, inAlbum, inPlaylist
  - Painting
  - Photograph
  - Recipe: cookTime, cookingMethod, ingredients, nutrition, prepTime, recipeCategory, recipeCuisine, recipeInstructions, recipeYield, totalTime
  - Review: itemReviewed, reviewBody, reviewRating

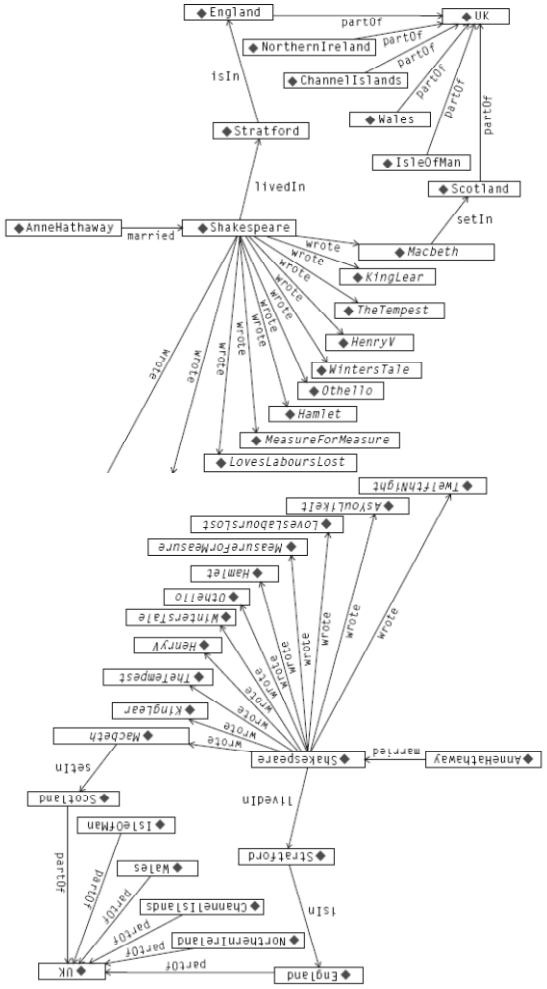
Zitat:
"… tags, that webmasters can use to markup their pages in ways recognized by major search providers. Search engines including Bing, Google, Yahoo! and Yandex rely on this markup to improve the display of search results, making it easier for people to find the right web pages."

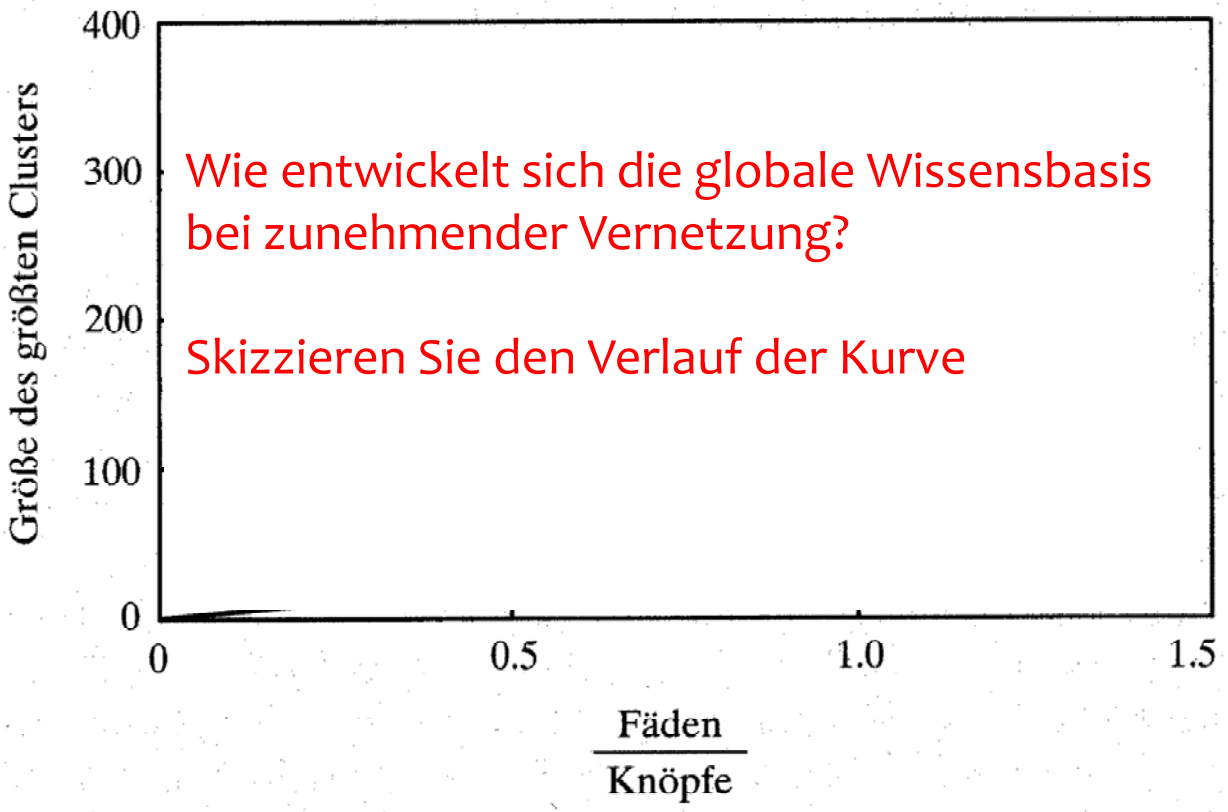
http://schema.org/docs/full.html

Von der Stringsuche zur semantischen Suche



Vernetzung: Phasenübergang nach Stuart Kauffmann





Wie entwickelt sich die globale Wissensbasis bei zunehmender Vernetzung?

Skizzieren Sie den Verlauf der Kurve

kristallisiert. (Bei diesem Experiment schwankt die Anzahl der Fäden zwischen 0 und 720, während die Anzahl der Knöpfe mit 400 konstant bleibt.)

[Kauffmann, Der Öltropfen im Wasser]

Vernetzung: Phasenübergang nach Stuart Kauffmann

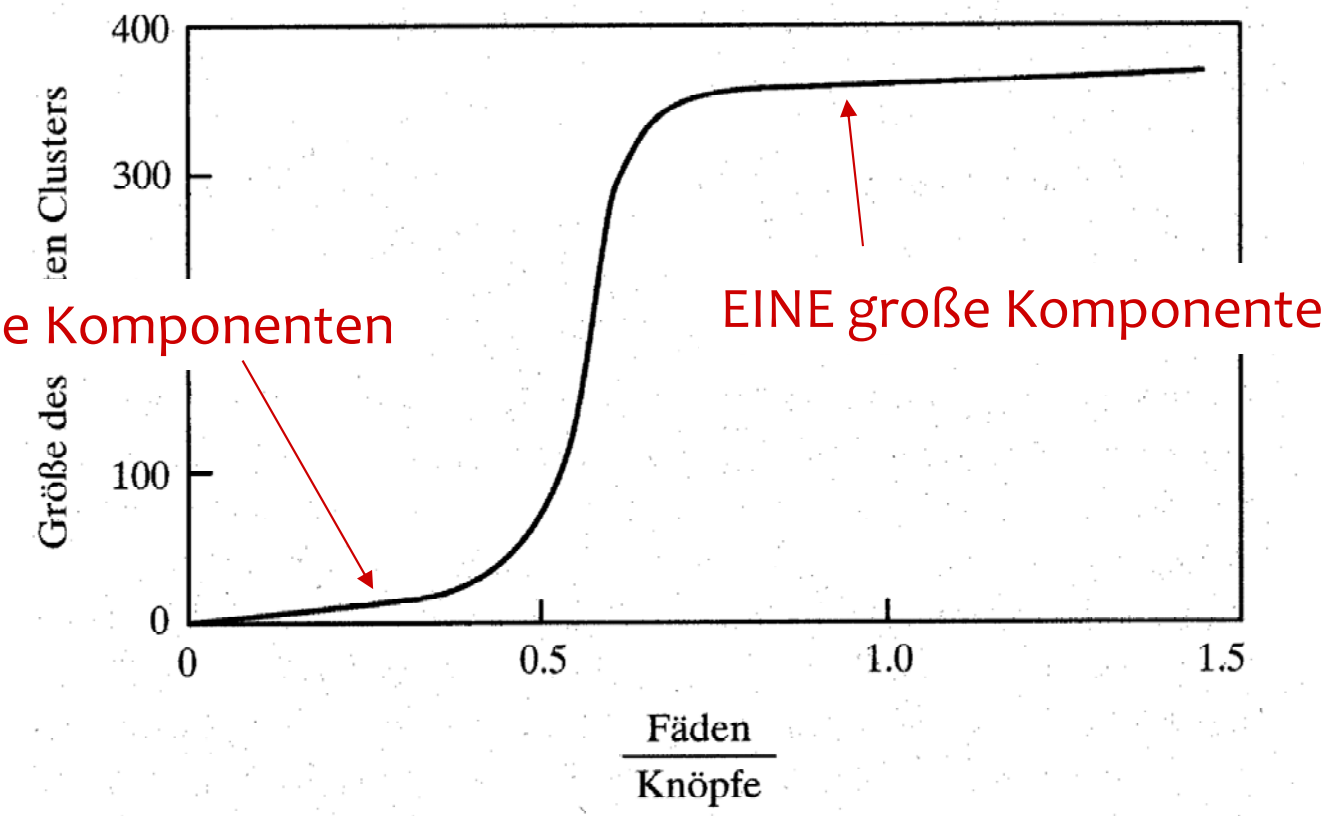


Abbildung 3.4: *Ein Phasenübergang. Sobald das Verhältnis von Fäden (Kanten) zu Knöpfen (Knoten) in einem Zufallsgraphen den Wert von 0,5 überschreitet, nimmt die Größe des zusammenhängenden Clusters langsam zu, bis er einen »Phasenübergang« erreicht und sich zu einer riesigen Komponente kristallisiert. (Bei diesem Experiment schwankt die Anzahl der Fäden zwischen 0 und 720, während die Anzahl der Knöpfe mit 400 konstant bleibt.)*

[Kauffmann, Der Öltropfen im Wasser]

- Für die 1. Vision der **Vernetzung** genügt die Wissensrepräsentation in semantischen Netzen (W3C-Standards RDF und RDFS)

- Für die 2. Vision der **maschinellen Inferenz** sind Beschreibungslogiken notwendig (W3C-Standards RDF, RDFS, OWL und OWL 2)

Next

- Ontologie und die beiden Visionen des ‚Semantic Web'
- RDF, RDFS
- Web Ontology Language OWL
  - OWL lesen – Sprachkonstrukte
  - Beispiel einer Ontologie: FOAF

PR

RDF – Ressource Description Framework

Was lässt sich in RDF beschreiben?

1) Beziehungen zwischen Instanzen

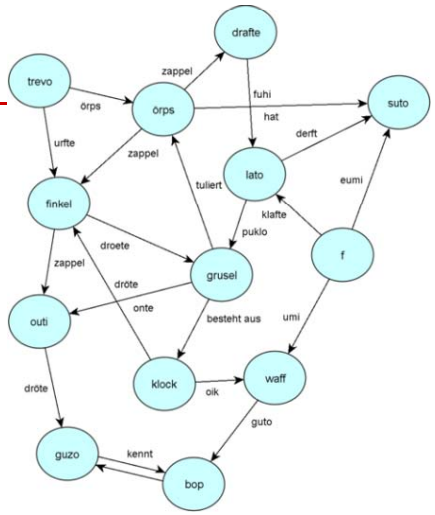
- In DL: eine Assertion – die **Rollenzuweisung R(b,c)**
- Die Instanz b steht zur Instanz c in der Rolle R
- **Beispiel:** $istVaterVon(paul, anna)$ ← $Beschreibungslogischer\ Ausdruck$
  - Die Instanz paul steht zur Instanz anna in der Relation istVaterVon

2) Klassenzugehörigkeit (=Instanziierung)

- Konstrukt **rdf:type**
- Beziehung = Relation = Rolle ‚instance_of',
- In DL: eine Assertion – die **Konzeptzuweisung C(a)**
- **Beispiel:** $Professor(uhlig)$
  - Die Instanz uhlig gehört zur Klasse Professor

Das war's für RDF.

RDF



Was lässt sich in RDF beschreiben?

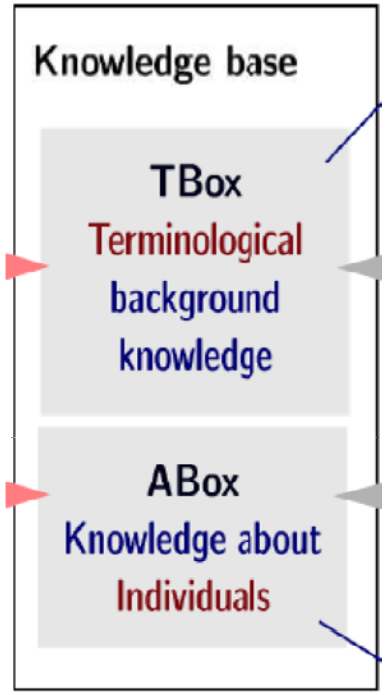
- **R(b,c) und C(a)**
- Beschreibung eines konkreten Zustands in der Anwendungsdomäne mit Hilfe von Konzepten und Rollen aus der TBox - die Verbindung zwischen Begriffswelt und realer (wahrgenommener) Welt

Wie wird dieser Teil des Wissens bezeichnet?

- Die ABox einer Ontologie lässt sich in RDF beschreiben

- Erst RDFS ermöglicht einfache Konstrukte in der TBox

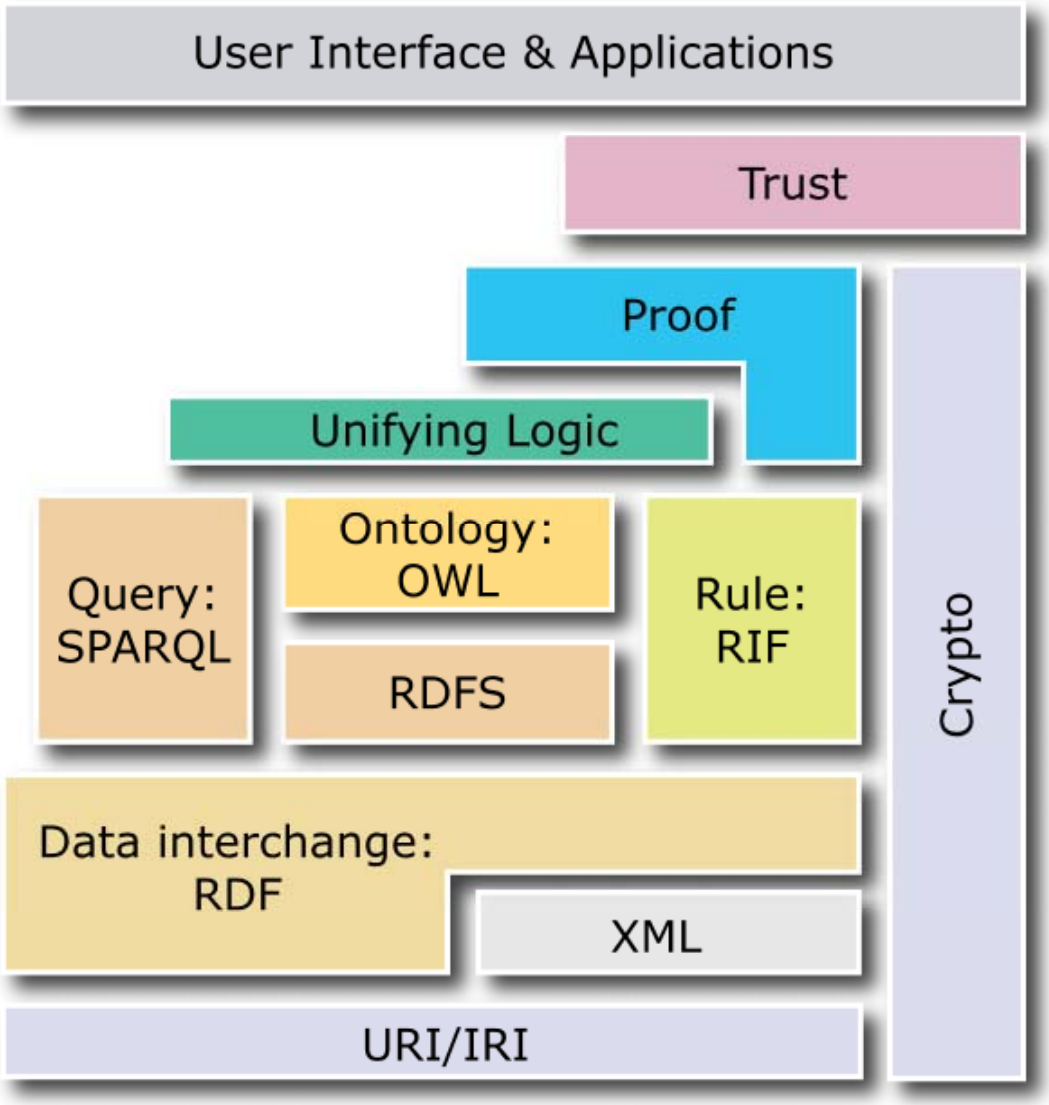
  –> Terminologisches Wissen



Beschreibungsschichten im Semantic Web

- RDF: Repräsentation von Fakten über Instanzen – nicht über Klassen
- RDFS: einfache Aussagen über Klassen, Rollen – mit Semantik
- OWL und OWL2: komplexe Aussagen über Klassen und Rollen

Versuchen Sie nicht, diese Abbildung zu interpretieren



RDFS - RDF Vocabulary Description Language

- oder kurz: RDF-Schema
- http://www.w3.org/2000/01/rdf-schema#

Welche zusätzliche Ausdruckskraft bringt RDFS?

RDFS: Klassen

Klassen (Konzepte) mit DL-Semantik

- **rdfs:class** Die Klasse aller Klassen (Etwas ist eine Klasse, = Konzept)
  - Beispiel: *Professor* ist eine Klasse
  - Genauer: *Professor* ist eine Instanz von rdfs:class (und damit eine Klasse)

- **rdf:type** Instanziierung, möglich ist sogar Mehrfachklassifikation
  - Beispiel: *fred_schulze* ist eine Instanz eines *Professors*

- **rdfs:subClassOf** The subject is a subclass of a class. (Abstraktionsrelation is_a, Teilmenge, DL: Konzeptinklusion, Mehrfachvererbung erlaubt)
  - Beispiel: *Professor* ist eine Unterklasse von *Lehrender*

Diese Liste wurde erzeugt mit der SPARQL-Anfrage:
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> select ?label ?comment from <http://www.w3.org/2000/01/rdf-schema>
{?EtwasAusRDFS rdfs:label ?label . ?EtwasAusRDFS rdfs:comment ?comment} limit 50

RDFS: Properties

Properties (Beziehungen, Relationen, Rollen):

- **rdf:property** (etwas ist eine Relation)
  - Beispiel: *istVaterVon* ist eine Rolle

- **rdfs:domain** A domain of the subject property, Klassenvorgabe für die linke Seite einer Rolle, Definitionsbereich einer Rolle
  - Beispiel: Die Rolle *istVaterVon* geht immer von einer *Person* aus
- **rdfs:range** A range of the subject property, Klassenvorgabe für die rechte Seite einer Rolle, Wertebereich einer Rolle
  - Beispiel: Die Rolle *istVaterVon* zeigt immer auf ein *Kind*

PR

- **rdfs:subPropertyOf** The subject is a subproperty of a property, Spezialisierung von Relationen, Teilmenge, Rolleninklusion
  - Beispiel: Die Rolle *istVaterVon* ist Subproperty der Rolle *istVorfahreVon*

->Übung

Diese Liste wurde erzeugt mit der SPARQL-Anfrage:
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> select ?label ?comment from <http://www.w3.org/2000/01/rdf-schema>
{?EtwasAusRDFS rdfs:label ?label . ?EtwasAusRDFS rdfs:comment ?comment} limit 50

RDFS

Andere:

- rdfs:label A human-readable name for the subject.
- rdfs:comment A description of the subject resource.
- rdfs:seeAlso Further information about the subject resource. (Assoziationsrelation)
- rdfs:isDefinedBy
  - The definition of the subject resource,
  - Verweis auf eine externe Definition, die nicht in RDF vorliegt,
  - subProperty of seeAlso
- rdfs:resource The class resource, everything.
- rdfs:member A member of the subject resource.
- rdfs:Literal The class of literal values, eg. textual strings and integers.
- rdfs:Datatype The class of RDF datatypes.
- rdfs:Container The class of RDF containers.
- rdfs: ContainerMembershipProperty The class of container membership properties, rdf:_1, rdf:_2, ..., all of which are sub-properties of 'member'.

Diese Liste wurde erzeugt mit der SPARQL-Anfrage:
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> select ?label ?comment from <http://www.w3.org/2000/01/rdf-schema>
{?EtwasAusRDFS rdfs:label ?label . ?EtwasAusRDFS rdfs:comment ?comment} limit 50

Beispiel rdfs:subClassOf

- TBox:
- Es gibt die 5 Klassen: Motorvehicle, Truck, Van, PassengerVehicle, Minivan
- Es gibt 5 Subklassenbeziehungen: Truck is_a MotorVehicle u.s.w.

Instanziierung: Motorvehicle is_a rdfs_class

Klassenhierarchie, Abstraktion

Mehrfachvererbung

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xml:base="http://example.org/schemas/vehicles">

        <rdf:Description rdf:ID="MotorVehicle">
                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        </rdf:Description>

        <rdf:Description rdf:ID="PassengerVehicle">
                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
                <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
        </rdf:Description>

        <rdf:Description rdf:ID="Truck">
                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
                <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
        </rdf:Description>

        <rdf:Description rdf:ID="Van">
                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
                <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
        </rdf:Description>
        <rdf:Description rdf:ID="MiniVan">
                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
                <rdfs:subClassOf rdf:resource="#Van"/>
                <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
        </rdf:Description>
</rdf:RDF>
```

RDFS: Zwei Relationen definieren

Relation

Vorwärtsdeklaration

```
<rdf:Property rdf:ID="registeredTo">
        <rdfs:domain rdf:resource="#MotorVehicle"/>
        <rdfs:range rdf:resource="#Person"/>
</rdf:Property>
```

Abstrakte Rolle (zeigt auf eine Klasse)

Vorwärtsdeklaration

```
<rdf:Property rdf:ID="rearSeatLegRoom">
        <rdfs:domain rdf:resource="#PassengerVehicle"/>
        <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
```

Konkrete Rolle (zeigt auf einen vordefinierten Datentyp)

Klasse

```
<rdfs:Class rdf:ID="Person"/>
```

Datentyp

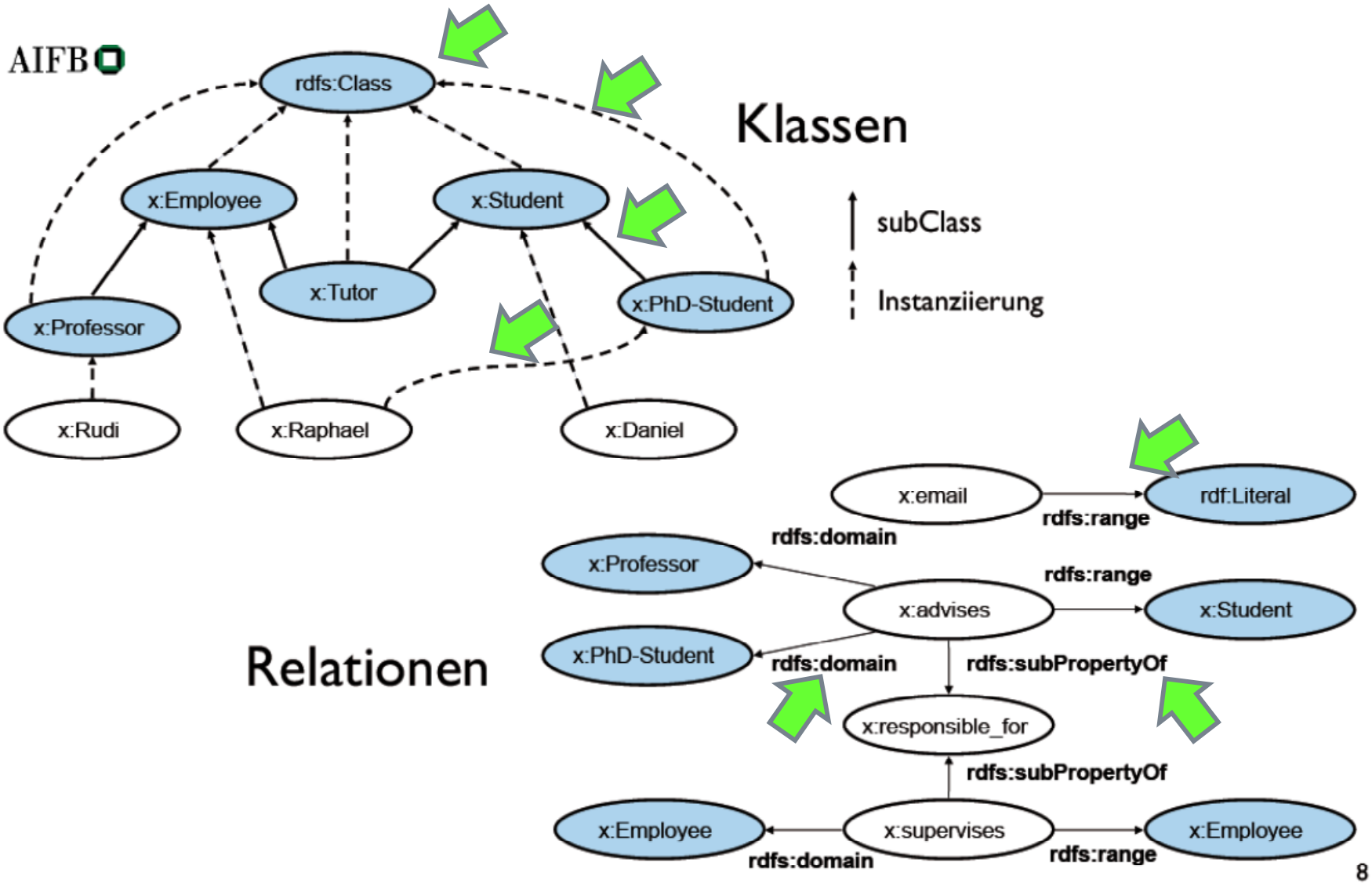
```
<rdfs:Datatype rdf:about="&xsd;integer"/>
```

Rollenhierarchie

```
<rdf:Property rdf:ID="driver">
        <rdfs:domain rdf:resource="#MotorVehicle"/>
</rdf:Property>

<rdf:Property rdf:ID="primaryDriver">
        <rdfs:subPropertyOf rdf:resource="#driver"/>
</rdf:Property>
```

Ausdruckstärke von RDFS



PR

Semantische Ausdruckstärke von RDFS

1. Definition von Klassen (= Konzept) „ A ist ein Klasse“
2. Klassenhierarchie: is-a-Beziehung rdfs:subClassOf
3. Definition von Rollen („R ist eine Rolle“) mit Domain und Range
4. Rollenhierarchie/-inklusion rdfs:subProperty
5. ABox aus RDF

Fazit RDF + RDFS

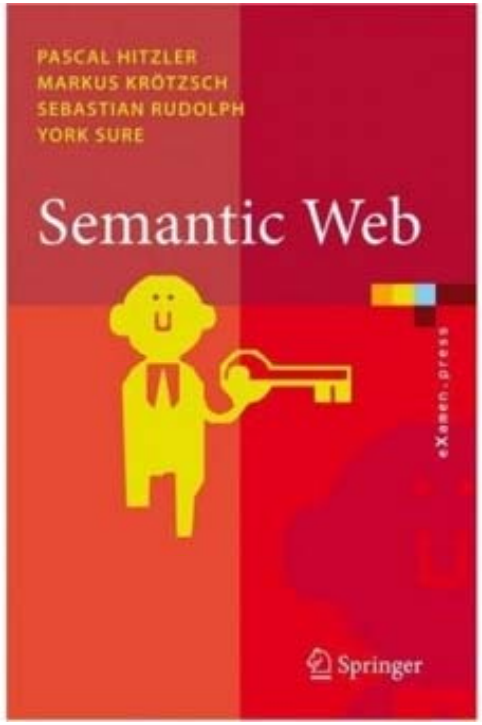
- geeignet für einfache Ontologien (wie bspw. schema.org)
- Vorteil: effizientes Schlussfolgern
- Nachteil: beschränkte Ausdruckskraft
  - Fehlend: Existenz, Kardinalität, R inverse, transitiv, symmetrisch
- Rückgriff auf mächtigere Sprachen, wie OWL

Next

- Ontologie und die beiden Visionen des ‚Semantic Web'
- RDF, RDFS
- Web Ontology Language OWL
  - OWL lesen – Sprachkonstrukte
  - Beispiel einer Ontologie: FOAF
- Von OWL zu OWL2

Nachfolgende Folien sind angelehnt an [AIFB]:

Buchempfehlung



[AIFB] Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB)
am Karlsruher Institut für Technologie (KIT):
http://semantic-web-grundlagen.de/wiki/SWebT1_WS09/10

OWL

- **W**eb **O**ntology **L**anguage

- http://www.w3.org/2002/07/owl#
- W3c Recommendation seit 2004
- Untermenge der Prädikatenlogik erster Stufe
- Drei Varianten
  - OWL Lite < OWL DL < OWL Full

- Praktisch relevant: **OWL DL**
  - ist entscheidbar
  - Entspricht Beschreibungslogik $\mathcal{SHOIN}(\mathcal{D})$

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
  - **Entscheidbar**.
  - Wird von aktuellen Softwarewerkzeugen fast vollständig unterstützt.

- **OWL Lite**
  - **Entscheidbar**.
  - Wenig ausdrucksstark.

OWL DL

Class expressions allowed in: rdfs:domain, rdfs:range, rdfs:subClassOf
owl:intersectionOf, owl:equivalentClass, owl:allValuesFrom, owl:someValuesFrom

Values are not restricted (0..N) in: owl:minCardinality, owl:maxCardinality, owl:cardinality

owl:DataRange, rdf:List, rdf:first, rdf:rest, rdf:nil

owl:hasValue (*daml:hasValue*)
owl:oneOf (*daml:oneOf*)
owl:unionOf (*daml:unionOf*), owl:complementOf (*daml:complementOf*)
owl:disjointWith (*daml:disjointWith*)

OWL Lite

owl:Ontology (*daml:Ontology*),
owl:versionInfo (*daml:versionInfo*),
owl:imports (*daml:imports*),
owl:backwardCompatibleWith,
owl:incompatibleWith, owl:priorVersion,
owl:DeprecatedClass,
owl:DeprecatedProperty

owl:Class (*daml:Class*),
owl:Restriction (*daml:Restriction*),
owl:onProperty (*daml:onProperty*),
owl:allValuesFrom (*daml:toClass*) (only with class identifiers and named datatypes),
owl:someValuesFrom (*daml:hasClass*) (only with class identifiers and named datatypes),
owl:minCardinality (*daml:minCardinality*; restricted to {0,1}),
owl:maxCardinality (*daml:maxCardinality*; restricted to {0,1}),
owl:cardinality (*daml:cardinality*; restricted to {0,1})

owl:intersectionOf (only with class identifiers and property restrictions)

owl:ObjectProperty (*daml:ObjectProperty*),
owl:DatatypeProperty (*daml:DatatypeProperty*),
owl:TransitiveProperty (*daml:TransitiveProperty*),
owl:SymmetricProperty,
owl:FunctionalProperty (*daml:UniqueProperty*),
owl:InverseFunctionalProperty (*daml:UnambiguousProperty*),
owl:AnnotationProperty

owl:Thing (*daml:Thing*)
owl:Nothing (*daml:Nothing*)

owl:inverseOf (*daml:inverseOf*),
owl:equivalentClass (*daml:sameClassAs*) (only with class identifiers and property restrictions),
owl:equivalentProperty (*daml:samePropertyAs*),
owl:sameAs (*daml:equivalentTo*),
owl:sameIndividualAs,
owl:differentFrom (*daml:differentIndividualFrom*),
owl:AllDifferent, owl:distinctMembers

RDF(S)

rdf:Property
rdfs:subPropertyOf
rdfs:domain
rdfs:range (only with class identifiers and named datatypes)
rdfs:comment, rdfs:label, rdfs:seeAlso, rdfs:isDefinedBy
rdfs:subClassOf (only with class identifiers and property restrictions)

OWL DL

OWL Lite

RDF(S)

Vokabular von OWL

owl:AllDifferent
owl:allValuesFrom
owl:AnnotationProperty
owl:backwardCompatibleWith
owl:cardinality
owl:Class
owl:complementOf
owl:DatatypeProperty
owl:DeprecatedClass
owl:DeprecatedProperty
owl:DataRange
owl:differentFrom
owl:disjointWith
owl:distinctMembers
owl:equivalentClass
owl:equivalentProperty
owl:FunctionalProperty
owl:hasValue
owl:imports
owl:incompatibleWith
owl:intersectionOf
owl:InverseFunctionalProperty
owl:inverseOf
owl:maxCardinality
owl:minCardinality
owl:Nothing
owl:ObjectProperty
owl:oneOf
owl:onProperty
owl:Ontology
owl:OntologyProperty
owl:priorVersion
owl:Restriction
owl:sameAs
owl:someValuesFrom
owl:SymmetricProperty
owl:Thing
owl:TransitiveProperty
owl:unionOf
owl:versionInfo
rdf:List
rdf:nil
rdf:type
rdfs:comment
rdfs:Datatype
rdfs:domain
rdfs:label
rdfs:Literal
rdfs:range
rdfs:subClassOf
rdfs:subPropertyOf

- DL-Semantik definiert unter: http://www.w3.org/TR/2004/REC-owl-semantics-20040210/

Vokabular von OWL – **mit Bezug zu DL**

owl:AllDifferent
owl:allValuesFrom
owl:AnnotationProperty
owl:backwardCompatibleWith
owl:cardinality
owl:Class
owl:complementOf
owl:DatatypeProperty
owl:DeprecatedClass
owl:DeprecatedProperty
owl:DataRange
owl:differentFrom
owl:disjointWith
owl:distinctMembers
owl:equivalentClass
owl:equivalentProperty

owl:FunctionalProperty
owl:hasValue
owl:imports
owl:incompatibleWith
owl:intersectionOf
owl:InverseFunctionalProperty
owl:inverseOf
owl:maxCardinality
owl:minCardinality
owl:Nothing
owl:ObjectProperty
owl:oneOf
owl:onProperty
owl:Ontology
owl:OntologyProperty
owl:priorVersion
owl:Restriction

owl:sameAs
owl:someValuesFrom
owl:SymmetricProperty
owl:Thing
owl:TransitiveProperty
owl:unionOf
owl:versionInfo
rdf:List
rdf:nil
rdf:type
rdfs:comment
rdfs:Datatype
rdfs:domain
rdfs:label
rdfs:Literal
rdfs:range
rdfs:subClassOf
rdfs:subPropertyOf

- Syntaxbeispiele unter: http://www.w3.org/TR/owl-ref/

DL in OWL

PR

- Die Beschreibungslogik $\mathcal{SHOIN}(\mathcal{D})$ als Basis für die Ontologiesprache OWL

$\mathcal{S}$ = $\mathcal{ALC}$ + transitive Hülle primitiver Rollen

$\mathcal{H}$ = Rollenhierarchie subProperty

$\mathcal{O}$ = Nominale = Namen von Objekten

$\mathcal{I}$ = Inverse Rolle

$\mathcal{N}$ = Number restriction

$(\mathcal{D})$ = konkrete Domäne = Datentypen

Erinnerung $\mathcal{ALC}$:

Atomic types: concept names $A, B, \ldots$
role names $R, S, \ldots$

Constructors: $\neg C$
$C \sqcap D$
$C \sqcup D$
$\exists R.C$
$\forall R.C$

DL in OWL

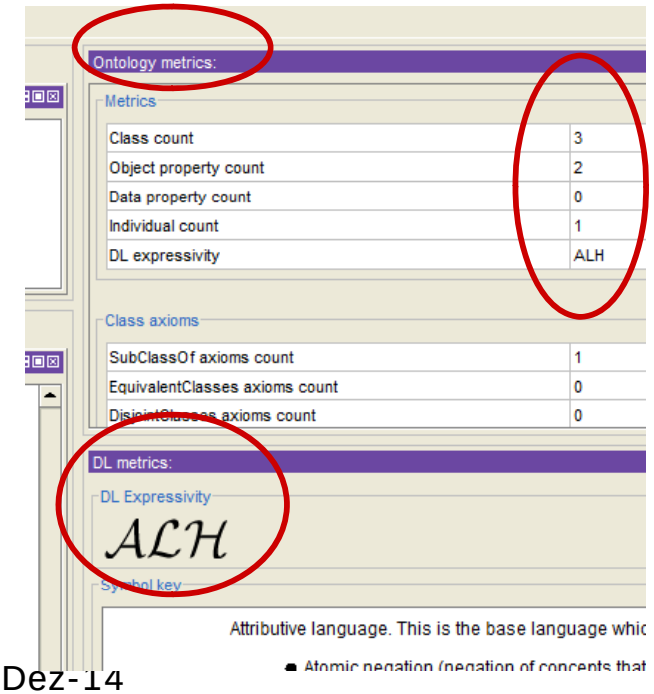
Construct Name	DL-Syntax	Semantics		
atomic concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$		
atomic role	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$		
transitive role	$R \in \mathbf{R}_+$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^+$		
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$		
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	$\mathcal{S}$	In OWL
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$		
exists restriction	$\exists R.C$	$\{x \mid \exists y.(x,y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$		
value restriction	$\forall R.C$	$\{x \mid \forall y.(x,y) \in R^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}\}$		
role hierarchy	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$	$\mathcal{H}$	In OWL
nominal	o	$o^{\mathcal{I}}$	$\mathcal{O}$	In OWL
inverse role	R^-	$\{(x,y) \mid (y,x) \in R^{\mathcal{I}}\}$	$\mathcal{I}$	In OWL
number	$\geqslant nP$	$\{x \mid \#\{y.(x,y) \in P^{\mathcal{I}}\} \geqslant n\}$	$\mathcal{N}$	In OWL
restrictions	$\leqslant nP$	$\{x \mid \#\{y.(x,y) \in P^{\mathcal{I}}\} \leqslant n\}$		
qualifying number	$\geqslant nP.C$	$\{x \mid \#\{y.(x,y) \in P^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geqslant n\}$	$\mathcal{Q}$	In OWL2
restrictions	$\leqslant nP.C$	$\{x \mid \#\{y.(x,y) \in P^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leqslant n\}$		

Nicht alle müssen verwendet werden..
Wo ist die Abbildung der Hierarchierelation is_a als Teilmenge?

Anzeige der Ausdruckskraft einer DL in Protege

Protege zeigt zu einer geladenen Ontologie den verwendeten **Umfang** der beschreibungs-logischen Konzepte: **DL-Metrik**

Beispiel



$\mathcal{AL}$	Attributive language. This is the base language which allows: • Atomic negation (negation of concepts that do not appear on the left hand side of axioms) • Concept intersection • Universal restrictions • Limited existential quatification (restrictions that only have fillers of Thing)
$\mathcal{FL}^-$	A sub-langauge of AL, which is obtained by disallowing atomic negation
$\mathcal{FL}_o$	A sub-language of FLo, which is obtained by disallowing limited existential quantification
$\mathcal{C}$	Complex concept negation
$\mathcal{S}$	An abbreviation for AL and C with transitive properties
$\mathcal{H}$	Role hierarchy (subproperties - rdfs:subPropertyOf)
$\mathcal{O}$	Nominals. (Enumerated classes or object value restrictions - owl:oneOf, owl:hasValue)
$\mathcal{I}$	Inverse properties
$\mathcal{N}$	Cardinality restrictions (owl:Cardinality, owl:minCardianlity, owl:maxCardinality)
$\mathcal{Q}$	Qualified cardinality restrictions (available in OWL 1.1)
$\mathcal{F}$	Functional properties
$\mathcal{E}$	Full existential quantification (Existential restrictions that have fillers other that owl:Thing)
$\mathcal{U}$	Concept union
(D)	Use of datatype properties, data values or datatypes

Abspeichern einer Ontologie als File

- Verschiedene Serialisierungen von OWL
  - Serialisierung **RDF/XML** <u>muss</u> von allen Tools implementiert werden

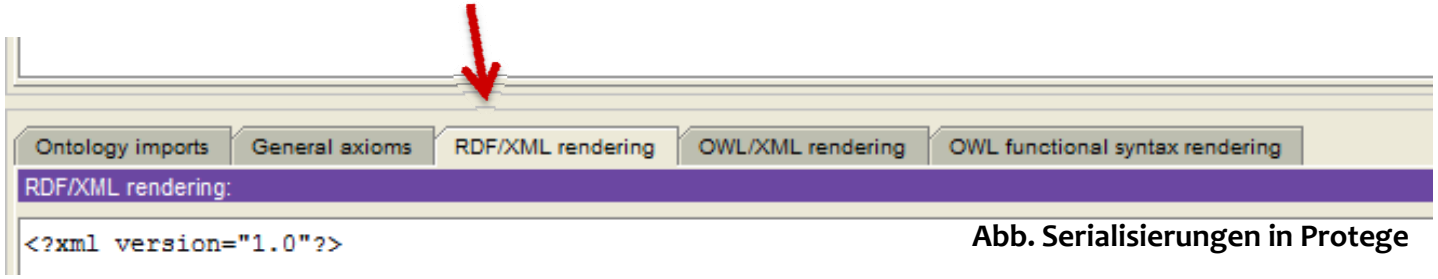


Abb. Serialisierungen in Protege

- OWL-Dokument besteht aus
  - Kopf
  - Wissensbasis
    - TBox
    - Abox

Kopf eines OWL-Dokumentes

OWL-Dokument besteht aus
- Kopf
- Wissensbasis
  - TBox
  - Abox

- **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>
```

Welche anderen Ontologien sollen verwendet werden

Kopf eines OWL-Dokumentes

OWL-Dokument besteht aus
- Kopf
- Wissensbasis
  - TBox
  - Abox

- **Allgemeine Informationen zur Ontologie**

```
<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.semanticwebgrundlagen.de/foo"/>
    <owl:priorVersion rdf:resource="http://ontoware.org/projects/swrc"/>
</owl:Ontology>
```

Kopf eines OWL-Dokumentes

OWL-Dokument besteht aus

- Kopf
- Wissensbasis
  - TBox
  - Abox

- 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

OWL Wissensbasis

PR

OWL-Dokument besteht aus
- Kopf
- Wissensbasis
  - TBox
  - Abox

- Beschreibt ein kontrolliertes Vokabular mit Hilfe von
  - **Klassen** (Konzepte, Begriffe)
  - **Rollen** (Relationen, Beziehungen, Eigenschaften, Properties)
  - **Individuen** (Instanzen, Objekte der realen Welt wie Du und ich)

  - Klassen zerfallen in OWL in
    - die eigentlichen Konzepte und
    - *Datentypen*. Diese betrachten wir nicht weiter.

OWL Wissensbasis

PR

OWL-Dokument besteht aus
- Kopf
- Wissensbasis
  - TBox
  - Abox

OWL Wissensbasen bestehen aus 2 Teilen:

- **TBox**: Axiome, die die Struktur der zu modellierenden Domäne beschreiben

  - $Vater \equiv Mann \sqcap \exists hatKind.Person$
  - $Elefant \sqsubseteq Tier \sqcap Gross \sqcap Grau$
  - $transitiv(hatVorfahr)$

- **Abox**: Axiome, die konkrete Situationen (Daten) beschreiben:

  - $Vater(Paul)$
  - $hatKind(Paul, Anna)$

Auf dieser Seite sind drei Regeln versteckt, mit denen Sie schlussfolgern können

Next

- Ontologie und die beiden Visionen des ‚Semantic Web'
- RDF, RDFS
- Web Ontology Language OWL
  - OWL lesen – Sprachkonstrukte
  - Beispiel einer Ontologie: FOAF
- Von OWL zu OWL2

OWL lesen –
Ein Schnappschuss der Sprachkonstrukte

Aktuelle Version:
OWL 2 Web Ontology Language
W3C Recommendation 11 December 2012

Sprachkonstrukte in OWL

- **Konstruktoren zur Beschreibung** komplexer Konzepte und Rollen aus einfachen Konzepten und Rollen
  - 1 Konzeptkonstruktoren (Descriptions)
  - 2 Rollenkonstruktoren
- **Axiome zum Ausdruck von** Fakten über Konzepte, Rollen und Individuen
  - 3 Über Konzepte
    - **C ist Unterklasse von <Konzeptkonstruktor>**
    - **C und D sind äquivalent**
    - C und D haben keine gemeinsamen Elemente
  - 4 Über Rollen
    - R hat Domain und Range
    - R ist eine Funktion, ist transitiv
  - 5 Über Individuen
    - a ist ein C, a ist verschieden von b, Rollenzuweisung R(b,c)

Von DL zu OWL

1 OWL-DL: Konstruktion von Konzepten

Wie können komplexe Konzepte beschrieben werden

Abstrakte Syntax in OWL	DL Syntax
A	A
owl:Thing	$\top$
owl:Nothing	$\bot$
intersectionOf($C_1 \ldots C_n$)	$C_1 \sqcap \ldots \sqcap C_n$
unionOf($C_1 \ldots C_n$)	$C_1 \sqcup \ldots \sqcup C_n$
complementOf(C)	$\neg C$
oneOf($o_1 \ldots o_n$)	$\{o_1\} \sqcup \ldots \sqcup \{o_n\}$
restriction(R someValuesFrom(C))	$\exists R.C$
restriction(R allValuesFrom(C))	$\forall R.C$
restriction(R hasValue(o))	$R : o$
restriction(R minCardinality(n))	$\geqslant n\, R$
restriction(R maxCardinality(n))	$\leqslant n\, R$

Alle diese DL-Ausdrücke können für „C..“ in DL-Ausdrücken verwendet werden.

R ist eine Rolle, eine sogenannte ObjectProperty (Relation zu einem Konzept)

Konstruktoren mit Datentypen sind hier weggelassen

Details der abstrakten Syntax sind hier nicht wichtig, aber die Begriffe finden sich in der konkreten OWL-Syntax wieder

Einfache Konstruktoren

Name	DL	OWL
Konjunktion	$C \sqcap D$	owl:intersectionOf
Disjunktion	$C \sqcup D$	owl:unionOf
Negation	$\neg C$	owl:complementOf

- Beispiel einer Konjunktion: $\text{Weißwein} \sqcap \text{Burgunder}$

```
<owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Burgundy" />
    <owl:Class rdf:about="#WhiteWine" />
</owl:intersectionOf>
```

1

PR

OWL-DL: Konstruktion von Konzepten

Value restriction $\forall R.C$

- Entspricht **owl:allValuesFrom**
- Alle Individuen, die zum gerade beschriebenen Konzept in der Relation R stehen, müssen zum Konzept C gehören
- Als unäres Prädikat:

$$PredAllRC(x) \Longleftrightarrow \forall y.R(x,y) \rightarrow C(y)$$

- DL-Beispiel: Welches Konzept wird hier beschrieben?

$$\forall hatKind.Female$$

```
<owl:Restriction>
    <owl:onProperty rdf:resource="#hatKind" />
    <owl:allValuesFrom rdf:resource="#Female" />
</owl:Restriction>
```

PR

1

OWL-DL: Konstruktion von Konzepten

Exists restriction $\exists R.C$

- Entspricht **owl:someValuesFrom**
- Es gibt mindestens ein Individuum, das zum gerade beschriebenen Konzept in der Relation R steht und zum Konzept C gehört
- Als unäres Prädikat:

$$PredExistRC(x) \Longleftrightarrow \exists y.R(x,y) \wedge C(y)$$

- DL-Beispiel: Welches Konzept wird hier beschrieben?

$$\exists hatPruefer.Person$$

```
<owl:Restriction>
        <owl:onProperty rdf:resource=„#hatPruefer"/>
        <owl:someValuesFrom rdf:resource=„#Person"/>
</owl:Restriction>
```

->Übung

1

OWL-DL: Konstruktion von Konzepten

(Unqualifizierte) numerische Restriktion $\geqslant nR$ und $\leqslant nR$

```
<owl:Class rdf:about=„ex:#Animal">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource=„ex:#hasGender"/>
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource=„ex:#hasHabitat"/>
      <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

$$Animal \sqsubseteq \leqslant 1 hasGender \sqcap \geqslant 1 hasGender$$

$$Animal \sqsubseteq \geqslant 1 hasHabitat$$

Ein Tier hat genau ein Geschlecht und mindestens einen Lebensraum.
Primitves Konzept ‚Animal', Konzeptinklusion.

http://protege.stanford.edu/plugins/owl/owl-library/koala.owl

Sprachkonstrukte in OWL

- **Konstruktoren zur Beschreibung** komplexer Konzepte und Rollen aus einfachen Konzepten und Rollen
  - Konzeptkonstruktoren (Descriptions)
  - 2 Rollenkonstruktoren
- **Axiome zum Ausdruck von** Fakten über Konzepte, Rollen und Individuen
  - Über Konzepte
    - **C ist Unterklasse von <Konzeptkonstruktor>**
    - **C und D sind äquivalent**
    - C und D haben keine gemeinsamen Elemente
  - Über Rollen
    - R hat Domain und Range
    - R ist eine Funktion, ist transitiv
  - Über Individuen
    - a ist ein C, a ist verschieden von b, Rollenzuweisung R(b,c)

2 OWL-DL: Konstruktion von Rollen

Abstrakte Syntax in OWL	DL Syntax
R	R
$\mathtt{inv}(R)$	R^-

->Übung

<owl:inverseOf rdf:resource="#hasMaker" />

Sprachkonstrukte in OWL

- **Konstruktoren zur Beschreibung** komplexer Konzepte und Rollen aus einfachen Konzepten und Rollen
  - Konzeptkonstruktoren (Descriptions)
  - Rollenkonstruktoren
- **Axiome zum Ausdruck von** Fakten über Konzepte, Rollen und Individuen
  - 3 Über Konzepte
    - **C ist Unterklasse von <Konzeptkonstruktor>**
    - **C und D sind äquivalent**
    - C und D haben keine gemeinsamen Elemente
  - Über Rollen
    - R hat Domain und Range
    - R ist eine Funktion, ist transitiv
  - Über Individuen
    - a ist ein C, a ist verschieden von b, Rollenzuweisung R(b,c)

3 OWL-DL: Axiome über Konzepte

Wie können Aussagen über Konzepte formuliert werden, z.B. Definition

Abstrakte Syntax in OWL	DL Syntax
Class(A partial $C_1 \ldots C_n$)	$A \sqsubseteq C_1 \sqcap \cdots \sqcap C_n$
Class(A complete $C_1 \ldots C_n$)	$A \equiv C_1 \sqcap \cdots \sqcap C_n$
EnumeratedClass(A $o_1 \ldots o_n$)	$A \equiv \{o_1\} \sqcup \cdots \sqcup \{o_n\}$
SubClassOf(C_1 C_2)	$C_1 \sqsubseteq C_2$
EquivalentClasses($C_1 \ldots C_n$)	$C_1 \equiv \cdots \equiv C_n$
DisjointClasses($C_1 \ldots C_n$)	$C_1 \sqcap C_j \sqsubseteq \bot, i \neq j$

in OWL nur paarweise, in OWL2 einfacher als Menge disjunkter Klassen

3 OWL-DL: Axiome über Konzepte

Klassenhierarchie und -definition

PR

$$Professor \sqsubseteq Fachbereichsmitglied$$

- "Jeder Professor ist Fachbereichsmitglied"
- $(\forall x)(Professor(x) \rightarrow Fachbereichsmitglied(x))$
- owl:subClassOf

$$Professor \equiv Fachbereichsmitglied$$

- "Die Fachbereichsmitglieder sind genau die Professoren"
- $(\forall x)(Professor(x) \leftrightarrow Fachbereichsmitglied(x))$
- owl:equivalentClass

3 OWL-DL: Axiome über Konzepte

Klassenhierarchie

PR

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

- Es folgt durch Inferenz, dass

$$Professor \sqsubseteq Fachbereichsmitglied$$

$$Fachbereichsmitglied \sqsubseteq Person$$

->Übung

3 OWL-DL: Axiome über Konzepte

Klassenäquivalenz

PR

- Nur damit können wir neue komplexe Klassen definieren:

```
<owl:Class rdf:about=„ #JemandNurMitToechtern“>
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty rdf:resource=„#hatKind"/>
        <owl:allValuesFrom rdf:resource=„#Female"/>
      </owl:Restriction>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  </owl:Class>
```

$$JemandNurMitToechtern \equiv \forall hatKind.Female$$

3 OWL-DL: Axiome über Konzepte

Disjunkte Klassen

$$Professor \sqsubseteq Fachbereichsmitglied$$

$$Buch \sqsubseteq Publikation$$

$$Fachbereichsmitglied \sqcap Publikation \sqsubseteq \bot$$

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

- Es folgt durch Inferenz, dass Professor und Buch

->Übung

Sprachkonstrukte in OWL

- **Konstruktoren zur Beschreibung** komplexer Konzepte und Rollen aus einfachen Konzepten und Rollen
  - Konzeptkonstruktoren (Descriptions)
  - Rollenkonstruktoren
- **Axiome zum Ausdruck von** Fakten über Konzepte, Rollen und Individuen
  - Über Konzepte
    - **C ist Unterklasse von <Konzeptkonstruktor>**
    - **C und D sind äquivalent**
    - C und D haben keine gemeinsamen Elemente
  - 4 Über Rollen
    - R hat Domain und Range
    - R ist eine Funktion, ist transitiv
  - Über Individuen
    - a ist ein C, a ist verschieden von b, Rollenzuweisung R(b,c)

4 OWL-DL: Axiome über Rollen

Abstrakte Syntax in OWL	DL Syntax
ObjectProperty(R super(R_1)...super(R_n)	$R \sqsubseteq R_i$
domain(C_1)...domain(C_m)	$\geqslant 1\, R \sqsubseteq C_i$
range(C_1)...range(C_ℓ)	$\top \sqsubseteq \forall R.C_i$
[inverseOf(R_0)]	$R \equiv R_0^-$
[Symmetric]	$R \equiv R^-$
[Functional]	$\top \sqsubseteq \leqslant 1\, R$
[InverseFunctional]	$\top \sqsubseteq \leqslant 1\, R^-$
[Transitive])	$Tr(R)$
SubPropertyOf(R_1 R_2)	$R_1 \sqsubseteq R_2$
EquivalentProperties($R_1 \ldots R_n$)	$R_1 \equiv \ldots \equiv R_n$

4

OWL-DL: Axiome über Rollen

Domain und Range (noch aus RDFS)

PR

- Beispiel: Eine Relation **hasChildren**, die als Definitionsbereich (Domain) und als Wertebereich (Range) Objekte der Klasse ‚**Animals**‘ erlaubt:

```
<owl:ObjectProperty rdf:about=„ex:#hasChildren">
    <rdfs:range rdf:resource=„ex:#Animal"/>
    <rdfs:domain rdf:resource=„ex:#Animal"/>
</owl:ObjectProperty>
```

$$ObjectProperty(hasChildren)$$
$$\top \sqsubseteq \forall hasChildren.Animal$$
$$\geqslant 1 hasChildren \sqsubseteq Animal$$

```
<owl:ObjectProperty rdf:about=„ex:#hasDegree">
    <rdfs:range rdf:resource=„ex:#Degree"/>
    <rdfs:domain rdf:resource=„ex:#Person"/>
</owl:ObjectProperty>
```

$$ObjectProperty(hasDegree)$$
$$\top \sqsubseteq \forall hasDegree.Degree$$
$$\geqslant 1 hasDegree \sqsubseteq Person$$

Aus http://protege.stanford.edu/plugins/owl/owl-library/koala.owl

->Übung

Range: Stellen wir uns ein Konzept vor, von dem Relationen hasChildren nur zu Animals führen (JemandNurMitAnimalsAlsKind). Alle Konzepte sollen in dieses Konzept gehören -> Damit treffen wir eine Aussage über hasChildren: ‚hasChildren zeigt nur auf Animals‘
Domain: Stellen wir uns die Konzepte vor, von denen mindestens eine hasChildren-Relation ausgeht . Alle diese Konzepte sollen Animals sein -> Damit Aussage über hasChildren: ‚hasChildren geht nur von Animals aus‘

- ZIEL 1. VL erreicht