

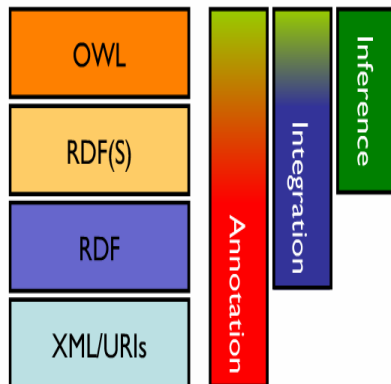
# Intelligente Systeme im WWW: Semantic Web

## RDF und RDF Schema

Dr. Raphael Volz  
Institut AIFB, Universität Karlsruhe (TH)

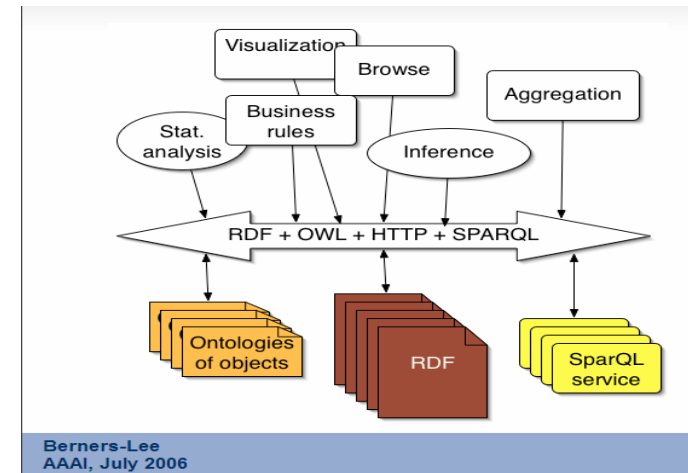
*(Most) Material courtesy  
Prof. Dr. Heiner Stuckenschmidt  
Uni Mannheim*

### Semantic Web - High-Level Objectives



- **Annotation**
  - Associating metadata with resources
- **Integration**
  - Integration of information over several data sets (“Web of Data”)
- **Inference**
  - Reasoning over Data
  - Could be light-weight (RDF Schema)
  - Could be heavy-weight (OWL)

### Further Uses of RDF...



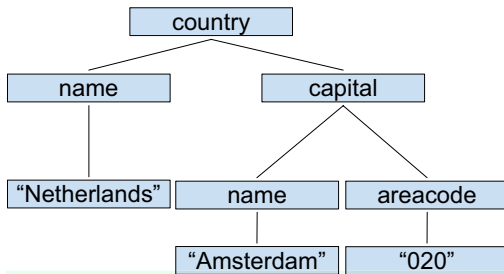
### Inhalt

- Kurz: XML
- Was ist RDF
- RDF im Detail
  - Datenmodell
  - Syntax
- RDF Schema
  - Vokabulary
  - Schema Syntax und Inferenz
- Sesame: eine RDF Datenbank

## Erinnerung: XML

```
<country name="Netherlands">
  <capital name="Amsterdam">
    <areacode>020</areacode>
  </capital>
</country>
```

- Syntax:
  - Elemente,
  - Attribute
  - Daten
- Datenmodell:
  - Baumstruktur



Slide 5

## Probleme von XML

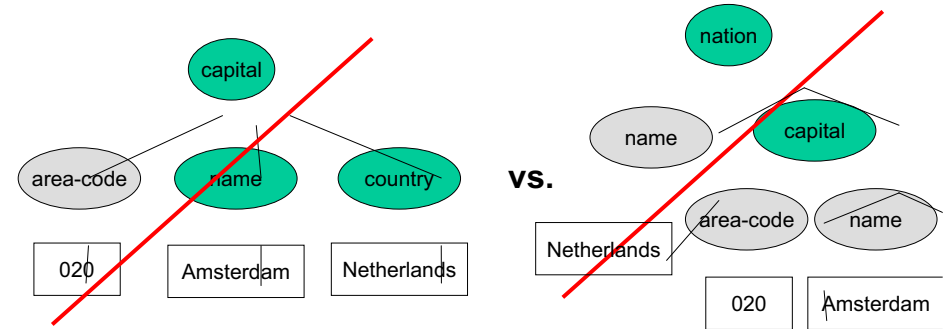
- Keine Festlegung auf eindeutige Struktur:
  - **ist country:**
    - Ein Object?
    - Eine Klasse?
    - Ein Attribut?
    - Eine Relation?
  - **Was bedeutet die Schachtelung?**
- Keine Festlegung auf ein eindeutiges Vokabular
  - Ist **country** das gleiche wie **nation**?

```
<capital areacode="020">
  <name>Amsterdam</name>
  <country>Netherlands</country>
</capital>
```

```
<nation name="Netherlands">
  <capital name="Amsterdam">
    <areacode>020</areacode>
  </capital>
</nation>
```

Slide 6

## Eindeutige Strukturen



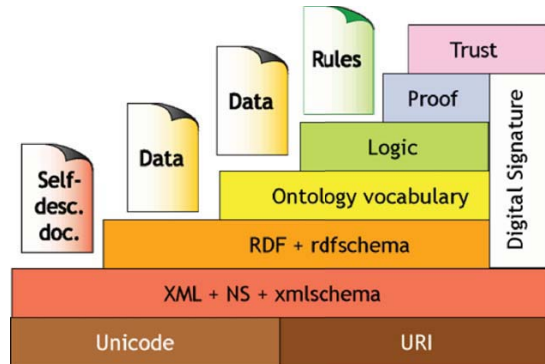
Slide 7

## Was ist RDF?

- RDF
  - “Resource Description Framework”
  - W3C Recommendation (<http://www.w3.org/RDF>)
- RDF ist ein Datenmodell
  - Für **Metadaten** (daten über Daten)
  - RDF beschreibt die Bedeutung von Informationen in Maschinenlesbarer Form

Slide 8

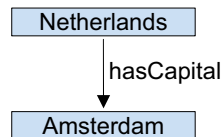
## Die Rolle von RDF im Semantic Web



Slide 9

## RDF im Detail: Das Datenmodell

- statements sind (subject, predicate, object) *triple*:
  - (Netherlands, hasCapital, Amsterdam)

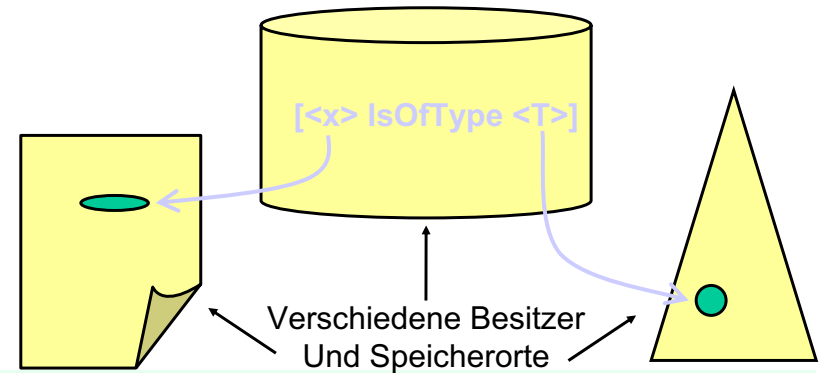


- statements beschreiben *resources (Object)*
- Eine resource kann alles sein, was eine URI besitzt:
  - Ein Dokument, ein Bild, ein Teil eines XML Dokuments
    - <http://www.cs.vu.nl/index.html>
  - ein Buch in einer Bibliothek:
    - [isbn://5031-4444-3333](http://isbn://5031-4444-3333)

Slide 10

## URI als Referenzen

- URIs können auf Informationen in unterschiedlichen Modellen verweisen:



Slide 11

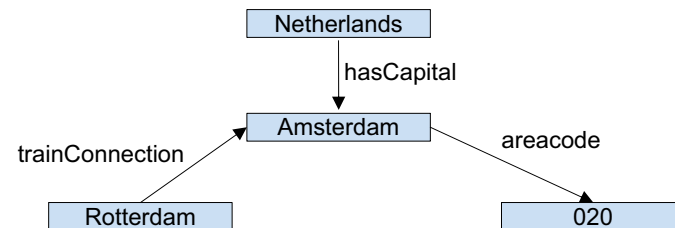
## URIs

- URI = Uniform Resource Identifier
- “The generic set of all names/addresses that are short strings that refer to resources”
- URLs (Uniform Resource Locators) sind spezielle URIs, die Adressen von Webseiten angeben.
- In RDF werden häufig URLs verwendet – oft mit direkten Verweisen auf Teile des Dokuments:
  - <http://somedomain.com/some/path/to/file#fragmentID>

Slide 12

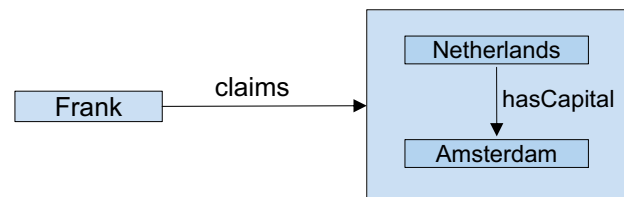
## Verknüpfung von Statements

- Das subject eines Statements kann das object (oder auch predicate) eines anderen statements sein
- Mengen solcher Statements bilden semantische Netzwerke:



Slide 13

## Reification: Ausagen über Aussagen



“Frank claims that the Netherlands has a capital called Amsterdam”

Slide 14

## RDF syntax: XML

- RDF hat eine spezielle XML-basierte Syntax
  - Jedes Description Element beschreibt eine resource
  - Jedes geschachtelte Element bezeichnet eine property
  - Das Attribut des property elements bezeichnet das object:

```

<Description about="http://www.countries.org/countries#Netherlands">
  <hasCapital resource="http://www.cities.org/cities#Amsterdam"/>
</Description>
<Description about="http://www.cities.org/cities#Amsterdam">
  <areacode>020</areacode>
</Description>
  
```

Slide 15

## RDF/XML syntax

- Wie in XML gibt es unterschiedliche Arten das gleiche aufzuschreiben, aber das Datenmodell bleibt das gleiche !

```

<Description about="http://www.countries.org/countries#Netherlands">
  <hasCapital resource="http://www.cities.org/cities#Amsterdam"/>
</Description>
<Description about="http://www.cities.org/cities#Amsterdam">
  <areacode>020</areacode>
</Description>
  
```

```

<Description about="http://www.countries.org/countries#Netherlands">
  <hasCapital resource="http://www.cities.org/cities#Amsterdam"/>
</Description>
<Description about="http://www.cities.org/cities#Amsterdam"
  areacode="020"/>
  
```

```

<Description about="http://www.countries.org/countries#Netherlands">
  <hasCapital resource="http://www.cities.org/cities#Amsterdam">
    <areacode>020</areacode>
  </hasCapital>
</Description>
  
```

Slide 16

## RDF/XML syntax: namespaces

- wie in XML werden namespaces eingesetzt, um tagnamen zu disambiguieren
- RDF eigene tags haben einen festgelegten Namespace, der Bezeichner ist standardmäßig 'rdf'

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:geo="http://www.geography.org/schema.rdf#"
  xmlns:words="http://www.dictionary.org/schema.rdf#">

  <rdf:Description rdf:about="#Netherlands">
    <geo:hasCapital rdf:resource="#Amsterdam"/>
    <words:hasCapital> N </words:hasCapital>
  </rdf:Description>
  <rdf:Description rdf:about="#Amsterdam">
    <geo:areacode>020</geo:areacode>
  </rdf:Description>
</rdf:RDF>
```

Slide 17

## RDF(S) Syntax: Turtle

- Die XML Syntax ist nur EINE mögliche Form, RDF Modelle zu beschreiben.
- Hier eine andere:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix geo: <http://www.geography.org/schema.rdf#>.
@prefix words: <http://www.dictionary.org/schema.rdf#>.
@prefix countries: <http://www.countries.org/countries#>.
@prefix cities: <http://www.cities.org/cities#>.

countries:Netherlands a geo:Country;
                       geo:hasCapital cities:Amsterdam;
                       words:hasCapital "N".
cities:Amsterdam geo:areacode "020".
```

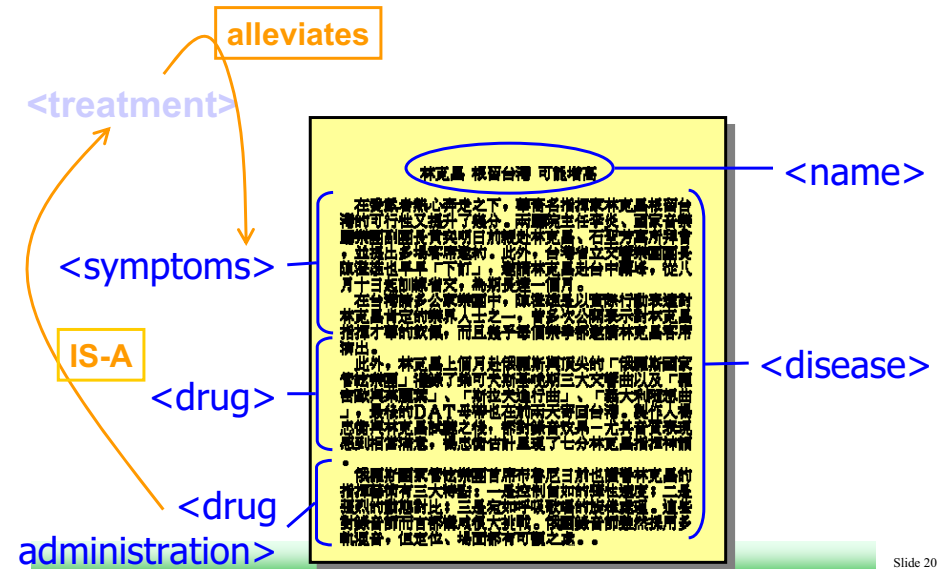
Slide 18

## Diskussion

- Vorteile von RDF :
  - Es wird explizit zwischen Objekten, Relationen und Konzepten unterschieden
  - Relationen zwischen Ressourcen werden explizit gemacht (auch über Modellgrenzen hinweg)
  - Das Datenmodelle ist invariant bzgl. unterschiedlicher Syntaktischer Varianten
- Offene Probleme
  - Es besteht keine Festlegung auf ein bestimmtes Vokabular:
    - Sind country und nation nun das gleiche ?
    - Welche Eigenschaften kann country haben ?

Slide 19

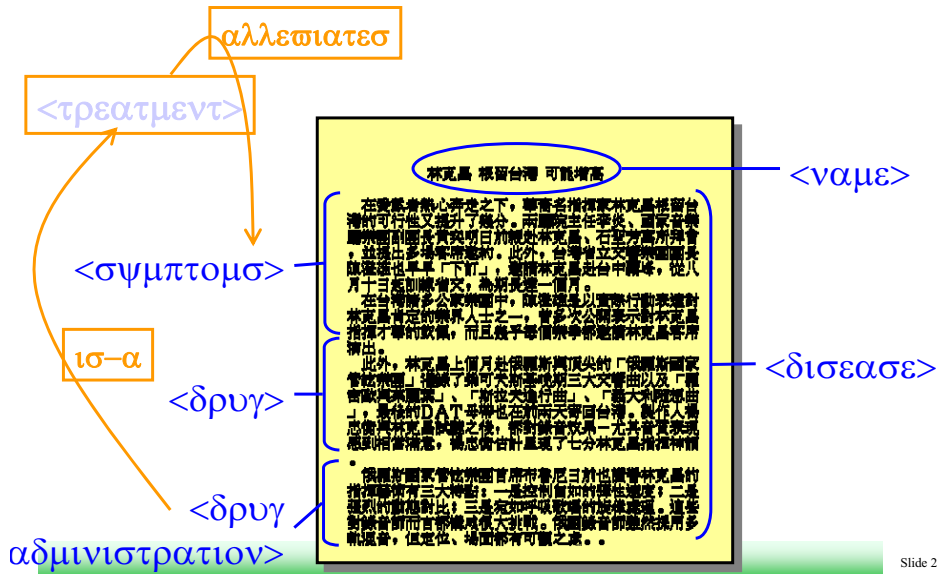
## Metadaten



Slide 20

# Metadaten in RDF

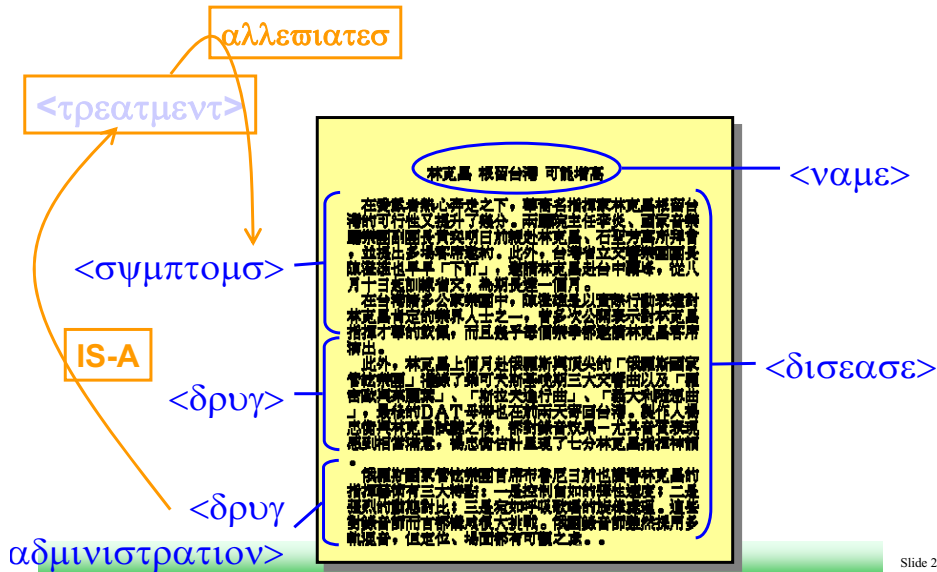
© Raphael Volz 2007



Slide 21

# RDF Schema

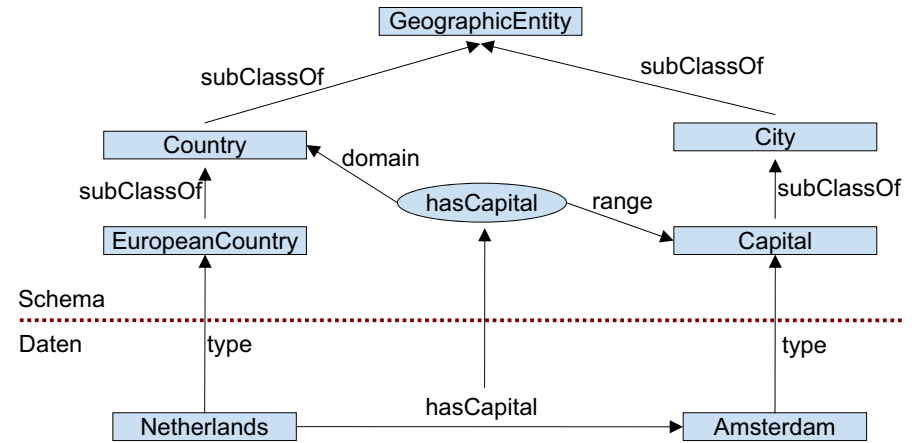
© Raphael Volz 2007



Slide 22

# RDF Schema

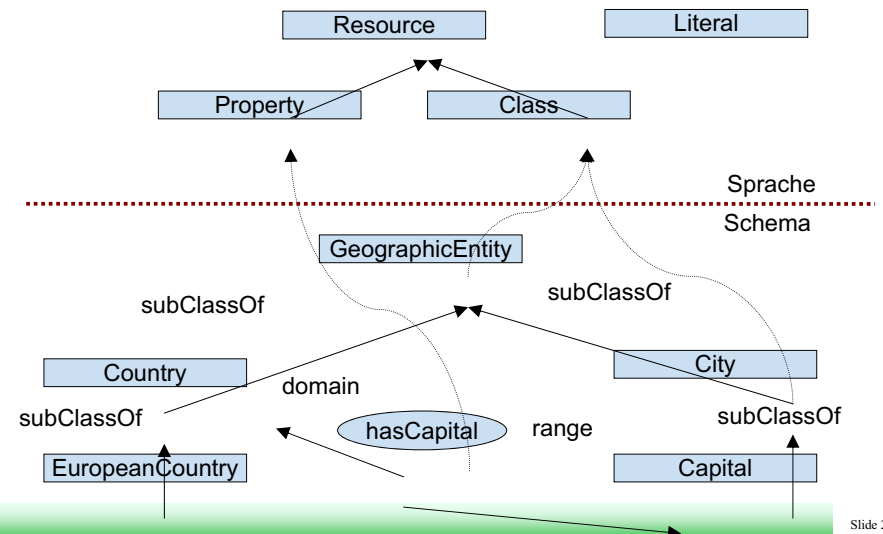
© Raphael Volz 2007



Slide 23

# RDF Schema

© Raphael Volz 2007



Slide 24

## Beobachtungen

- Relationen sind das wichtigste Instrument!
  - Relationen werden unabhängig von Klassen beschrieben
  - Unterschied zu gängigen Ansätzen im Software Engineering (z.B. UML) .
  - range/domain haben globale Bedeutung!
- Ein RDF Schema ist ein RDF Modell, in dem einige Relationen spezielle Bedeutung haben.
- Der Übergang zwischen den Metaebenen ist fließend
  - Ressourcen können gleichzeitig Instanzen und Klassen sein.

Slide 25

## Die Semantik von RDF Schema

- Die Bedeutung spezieller Relationen wird im RDF Schema Standard definiert, z.B.
  - 2.3.2 rdfs:subClassOf

“This property specifies a subset/superset relation between classes. The rdfs:subClassOf property is transitive. If class A is a subclass of some broader class B, and B is a subclass of C, then A is also implicitly a subclass of C. Consequently, resources that are instances of class A will also be instances of class C, since A is a subset of both B and C. Only instances of rdfs:Class can have the rdfs:subClassOf property and the property value is always of rdf:type rdfs:Class. A class may be a subclass of more than one class.”

- Frage: gilt (A subclass A)?

Slide 26

## Formale Semantik

- Welche statements lassen sich aus einem gegebenen Modell ableiten
- Menge von einfachen Ableitungsregeln
  - Einfach zu berechnen
  - Effizient in der Praxis
  - Keine kombinatorische Explosion
- Minimale Semantik ist bindend für alle Parteien.
- Darüber hinaus können anwendungsspezifisch weitere Regeln definiert werden

Slide 27

## RDFS Semantik: Beispiel

- Netherlands **Type** EuropeanCountry  
EuropeanCountry **subClassOf** Country  
→ Netherlands **Type** Country
- aspirin alleviates headache  
alleviates **range** symptom  
→ headache **Type** symptom

Slide 28

## RDFS Semantik: Beispiel

- Νετηερλανδσ **Type** ΕυροπεανΧουντρψ  
ΕυροπεανΧουντρψ **subClassOf** Χουντρψ  
→ Νετηερλανδσ **Type** Χουντρψ
- ασπριν αλλεπιατεσ ηεαδαχηε  
αλλεπιατεσ **range** συμπτομ  
→ ηεαδαχηε **Type** συμπτομ

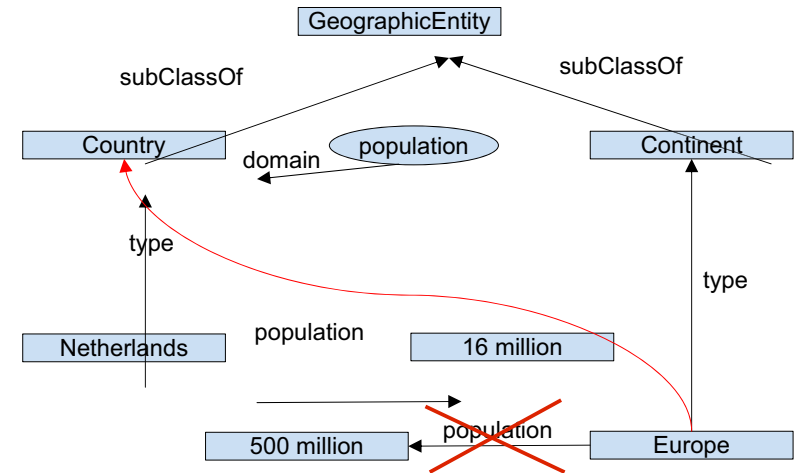
Slide 29

## RDFS Semantik: Beispiel

- $X R Y + R \text{ domain } T \rightarrow X \text{ IsOfType } T$
- $X R Y + R \text{ range } T \rightarrow Y \text{ IsOfType } T$
- $T1 \text{ SubClassOf } T2 + T2 \text{ SubClassOf } T3 \rightarrow T1 \text{ SubClassOf } T3$
- $X \text{ Type } T1 + T1 \text{ SubClassOf } T2 \rightarrow X \text{ IsOfType } T1$
- $X \text{ Type } \text{Class} \rightarrow X \text{ SubClassOf } X$

Slide 30

## Globale Einschränkungen



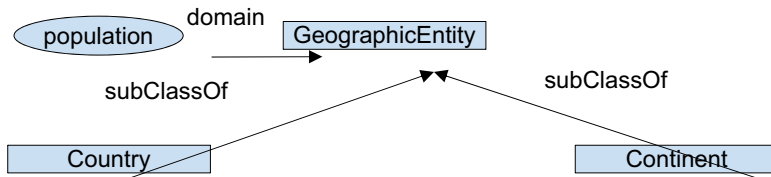
Slide 31

## Globale Einschränkungen



Slide 32

## Globale Einschränkungen



- Solution
  - moving the domain restriction 'up in the hierarchy'
  - This solves the problem, but **over-generalization** is a danger:
    - **properties get very 'loose' restrictions**
    - **properties might be used on classes for which they are not meant (but it is allowed because the restriction could not be made more specific)**

Slide 33

## RDF Schema Syntax

- Class definition

```

<rdf:Description rdf:about="#Country">
  <rdf:type rdf:resource="#Class"/>
  <rdfs:subClassOf rdf:resource="#GeographicEntity"/>
</rdf:Description>

```

- or shorter:

```

<rdfs:Class rdf:about="#Country">
  <rdfs:subClassOf rdf:resource="#GeographicEntity"/>
</rdfs:Class>

```

- Property definition

```

<rdf:Property rdf:about="#hasCapital">
  <rdfs:domain rdfs:resource="#Country"/>
  <rdfs:range rdfs:resource="#Capital"/>
</rdf:Property>

```

Slide 34

## Putting it all together

- Das schema file: <http://www.geography.org/schema.rdf>

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  <rdfs:Class rdf:about="#Country">
    <rdfs:subClassOf rdfs:resource="#GeographicEntity"/>
  </rdfs:Class>

  <rdf:Property rdf:about="#hasCapital">
    <rdfs:domain rdfs:resource="#Country"/>
    <rdfs:range rdfs:resource="#Capital"/>
  </rdf:Property>
</rdf:RDF>

```

- Das Modell: kann irgendwo anders liegen

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:geo="http://www.geography.org/schema.rdf#"
  <geo:Country rdf:about="#Netherlands">
    <geo:hasCapital rdfs:resource="#Amsterdam"/>
  </geo:Country>
  <geo:Capital rdf:about="#Amsterdam"/>
</rdf:RDF>

```

Slide 35

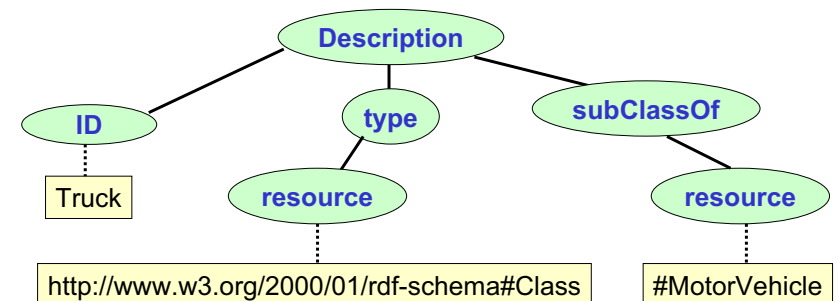
## Syntax und Interpretation

```

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

```

- Interpretation als XML:



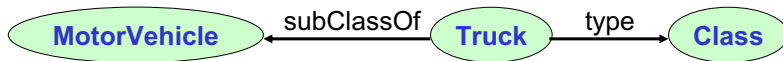
Slide 36

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

- Interpretation als RDF:
  - Anderes Datenmodell
  - **Description**, **ID** and **resource** haben eine festgelegte Bedeutung

**subject**                      **predicate**                      **object**

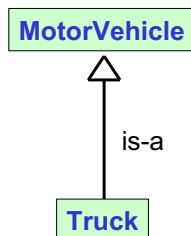
1. 'Truck'                      rdf:type                      #Class
2. 'Truck'                      rdfs:subClassOf                      #MotorVehicle



Slide 37

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

- Interpretation als RDF Schema
  - Wieder anderes Datenmodell
  - **type** and **subClassOf** werden speziell interpretiert



Slide 38

## Zusammenfassung

- RDF ist ein einfaches, Graph-basiertes Datenmodell für metadaten im Web
- RDF hat eine XML Syntax und kann so in Web Dokumente integriert werden
- Vorteile gegenüber XML
  - Datenmodell ist invariant gegenüber syntaktischen Variationen
  - Informationen aus unterschiedlichen Modellen können verbunden werden
  - Bestimmte Operationen, z.B. die Vereinigung zweier Modelle ist trivial
- RDF Schema erlaubt die Spezifikation von Hintergrundwissen
  - Vokabular: Class, SubClassOf, domain, range
  - Einfache Ableitungsregeln können implizites Wissen herleiten
  - RDF Schema ist RDF mit speziellen relationen

Slide 39