

Lehrveranstaltung

Semantic Web Technologien

SS 2010

HTWG Konstanz

RDFS

RDF Schema

Rückblick

Sprachen des Semantic Web – RDFS

- **XML** liefert uns ein Format zur **Serialisierung** strukturierter Daten
 - XML ist für das Semantic Web nicht ausreichend
 - Aber: Wir picken uns einige Bestandteile heraus
 - (Namespaces, DOCTYPE...)
- **RDF** liefert uns ein **Datenmodell** zur Beschreibung von **Ressourcen**
 - Modellierung der Daten als Graph – Zerlegung in **SPO Triple**
 - Mit SPO-Triplen lässt sich fast jede **Aussage** darstellen
 - Wir können **Beziehungen zwischen Ressourcen** modellieren
 - Wir können **Daten von Ressourcen** modellieren (Literale)
 - Wir können Literalen einen Typ geben! (Datum, String, Integer ...)

Was fehlt ?

Wir möchten mehrere Ressourcen eines **Typs** **zusammenfassen** können

=> Wir müssen **Typen** definieren können

=> Wir müssen Ressourcen **typisieren** (können)

– Beispiele:

- “Steffen Schlönvoigt” ist ein “Mann”
 - Wie stellen wir “ist ein” generisch dar?
 - Welchen Typ hat “Mann” ?

Sprachen des Semantic Web – RDFS

=> Wir müssen **Aussagen über Typen** treffen können

=> Typen müssen auch Ressourcen sein

- “Mann” ist Untertyp Von “Mensch”
 - Ist “Steffen Schlönvoigt” dann nicht auch ein “Mensch”?
 - Muss ich das explizit ausdrücken?
 - Für jeden Mann, für jede “Frau” einzeln?

=> Wir müssen eine **Hierarchie von Typen** aufbauen können

Wie sieht es bei **Properties** aus ?

– Beispiel:

- “Steffen” ist befreundet mit “Markus”
- “Steffen” kennt “Matthias”
 - Kennt Steffen Markus nicht?
 - Muss ich das explizit ausdrücken?

=> Wir benötigen eine **Hierarchie bei Properties**

– Ist ein “Auto” passend als Ziel der “ist befreundet mit” Beziehung?

=> Wir wollen **Aussagen über Properties** treffen

– Properties müssen **typisiert** werden

=> Wir wollen Typen für Subjekt und Objekt **eingrenzen** können

Solches Wissen nennt man “**Schemawissen**”

RDF Schema hilft uns, solches Wissen
auszudrücken

RDF Schema (RDFS)

- Gehört zum W3C Recommendation Paket zu RDF
- Ermöglicht Spezifikation von schematischem Wissen
 - Wird auch **terminologisches** Wissen genannt
 - In Ontologie: **T-Box**
- Besitzt eigenes/spezielles Vokabular
 - Vokabular ist definiert in RDF
 - Jedes RDFS Dokument ist gültiges RDF!
- Namensraum:
 - <http://www.w3.org/2000/01/rdf-schema#>
 - In der Regel abgekürzt durch **“rdfs”**

RDF Schema (RDFS)

- Vokabular ist nicht themengebunden, sondern **generisch**
- Erlaubt Spezifikation von **beliebigen** RDF-Vokabularen
- Erlaubt Spezifikation von **Semantik** von Vokabularen
 - => RDFS ist ein **Metavokabular**
- Vorteil:

Jedes mit RDFS definierte RDF-Vokabular kann von jeder Software mit RDFS Unterstützung interpretiert werden

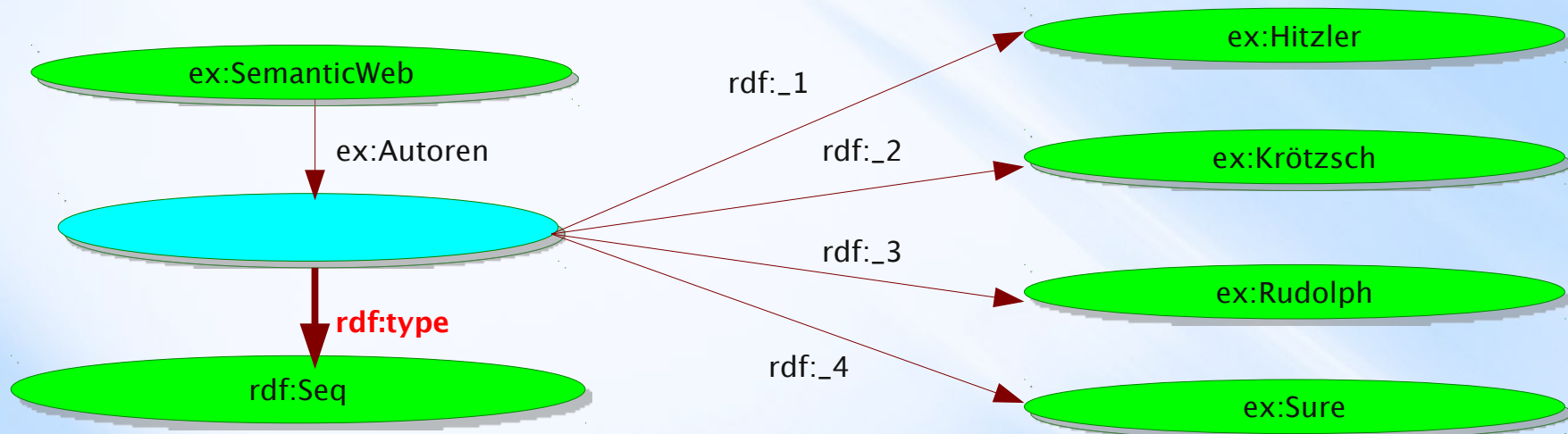
RDF Schema (RDFS)

- RDFS erlaubt die **Definition von Ontologien**
 - => RDFS ist eine Ontologiesprache (Wissensrepräsentationssprache)
- Allerdings: Funktionsumfang **eingeschränkt**
 - => Man spricht von lightweight/leichtgewichtigen Ontologien
- Wir werden uns später mit OWL, einer ausdrucksstärkeren Sprache, beschäftigen
- Aber ausdrucksstärkere Sprachen führen auch zu
 - Höherer Komplexität
 - Längerer Laufzeit der Algorithmen zum Schlussfolgern
- Mit ein bisschen Semantik kommt man schon recht weit
 - “A little semantics goes a long way”

Sprachen des Semantic Web – RDFS

Zuweisung von Instanzen zu Typen:

- RDF definiert die Eigenschaft “type”
- Bereits kennengelernt im Zusammenhang mit Listen:



- Mit “**rdf:type**” wird einem Subjekt das Objekt als Typ zugewiesen
- Das Objekt wird damit als “**Klasse**” ausgezeichnet
- Das Subjekt wird als “**Instanz**” der Klasse gesehen

Sprachen des Semantic Web – RDFS

Beispiel zu “**rdf:type**”:

```
ex:SteffenSchloenvoigt rdf:type ex:Mann .
```

– Folgendes kann daraus geschlossen werden:

- **ex:Mann** ist eine Klasse
- **ex:SteffenSchloenvoigt** ist eine Instanz, die der Klasse **ex:Mann** angehört
- Einem Individuum können **beliebig viele** Klassen zugewiesen werden
- **Unterscheidung** zwischen Klassen und Individuen syntaktisch **nicht möglich**
- Auch in der Realität ist Abgrenzung manchmal schwierig

Sprachen des Semantic Web – RDFS

- Manchmal möchten wir explizit ausdrücken, dass eine Ressource/URI eine Klasse bezeichnet

- Typ/Klasse aller Klassen:

```
rdfs:Class
```

- Beispiel:

```
ex:Mann rdf:type rdfs:Class .
```

- Wenn `rdfs:Class` Klasse aller Klassen ist, enthält sie sich auch selbst.

=> Es gilt immer:

```
rdfs:Class rdf:type rdfs:Class .
```

- **RDF/RDFS Standardklassen:**

- **rdfs:Class**
- **rdfs:Resource**
 - Klasse aller Ressourcen
- **rdf:Property**
 - Klasse aller Relationen
- **rdf:List, rdf:Seq, rdf:Bag, rdf:Alt, rdfs:Container**
- **rdfs:ContainerMembershipProperty**
 - Klasse aller Relationen, die eine Enthaltenseinsbeziehung darstellen
- **rdf:XMLLiteral**
 - Klasse aller Werte des Datentyps XMLLiteral
- **rdfs:Literal**
 - Klasse aller Literalwerte (enthält alle Datentypen als Sub-Class)
- **rdfs:Datatype**
 - Klasse aller Datentypen (analog zu **rdfs:Class**)
- **rdf:Statement**
 - Klasse aller reifizierten Aussagen (siehe später)

Sprachen des Semantic Web – RDFS

- Syntaktische Vereinfachungen in RDF/XML

An Stelle von

```
<rdf:Description rdf:about="&ex;SteffenSchloenvoigt">  
  <rdf:type rdf:resource="&ex;Mann" />  
</rdf:Description>
```

Kann auch

```
<ex:Mann rdf:about="&ex;SteffenSchloenvoigt" />
```

geschrieben werden.

Entsprechend gilt auch:

```
<rdfs:Class rdf:about="&ex;Mann" />
```


Sprachen des Semantic Web – RDFS

Hierarchisierung von Klassen (auch Taxonimien)

- RDFS-Vokabular definiert “**rdfs:subClassOf**”

- Beispiel:

```
ex:Mann          rdfs:subClassOf  ex:Mensch .
```

```
ex:Steffen      rdf:type          ex:Mann .
```

- RDFS-Software erkennt automatisch:

```
ex:Steffen      rdf:type          ex:Mensch .
```

- Man kann dies ausdrücken als “Mann ist eine Unterklasse von Mensch”

- “**rdfs:subClassOf**” ist **transitiv**.

- Mit zusätzlichem Triple

```
ex:Mensch       rdf:subClassOf  ex:Saeugetier .
```

- Folgt:

```
ex:Steffen      rdf:type          ex:Saeugetier .
```


Sprachen des Semantic Web – RDFS

- Weiteres zu `rdfs:subClassOf`

- `rdfs:subClassOf` ist **reflexiv**

=> Jede Klasse ist Unterklasse von sich selbst

Damit gilt implizit:

```
ex:Mann rdfs:subClassOf ex:Mann .
```

- Festlegung von **Gleichheit** zweier Klassen möglich mit gegenseitiger Unterklassenbeziehung

```
ex:Hospital rdfs:subClassOf ex:Krankenhaus .
```

```
ex:Krankenhaus rdfs:subClassOf ex:Hospital .
```

Sprachen des Semantic Web – RDFS

- RDFS hat ein “**mengentheoretisches**” Klassenmodell wie in der Mathematik
 - `rdf:type` entspricht \in
 - `rdfs:subClassOf` entspricht \subseteq
- Reflexivität und Transitivität von `rdfs:subClassOf` damit mathematisch hinterlegt

Sprachen des Semantic Web – RDFS

- Properties: (auch Relationen, Beziehungen)
 - Properties stehen in Tripeln an der Stelle des **Prädikats**
 - Sagen aus, auf welche Art Ressourcen miteinander verbunden sind
 - Mathematische Schreibweise:
 - `befreundetMit = {(Steffen,Markus),(Markus,Mario)...}`
 - Properties werden in RDFS speziellen Klassen zugeordnet
 - Klasse aller Properties in RDFS: `rdf:Property`
 - Es gilt bei Properties:
`ex:befreundetMit rdf:type rdf:Property .`

Sprachen des Semantic Web – RDFS

- Property-Hierarchien:
 - Werden in RDFS abgebildet durch Property “**rdfs:subPropertyOf**”
 - Beispiel:

```
ex:befreundetMit    rdfs:subPropertyOf    ex:kennt .
ex:Steffen          ex:befreundetMit      ex:Markus .
```

- Damit kann eine RDFS Software folgern:

```
ex:Steffen          ex:kennt              ex:Markus .
```


Sprachen des Semantic Web – RDFS

- Einschränkungen auf Properties

- Oft weiß man, dass das Subjekt oder das Objekt einer Property einer bestimmten Klasse angehört
- Den Definitionsbereich eines Property bezeichnet man als **domain**
- Den Bildbereich einer Property bezeichnet man als **range**
- Beispiel:

```
ex:employedAt    rdfs:domain    ex:Person .
ex:employedAt    rdfs:range    ex:Company .
ex:Steffen       ex:employedAt  ex:TechniData .
```

- Aus dem Beispiel kann eine RDFS-Software folgern:

- `ex:Steffen` `rdf:type` `ex:Person`
- `ex:TechniData` `rdf:type` `ex:Company`

- Kann auch für Datentypen bei Literalen verwendet werden:

- `ex:hasAge` `rdfs:range` `xsd:nonNegativeInteger`

Sprachen des Semantic Web – RDFS

- Property-Einschränkungen
 - Einzige Möglichkeit zur Definition semantischer Zusammenhänge zwischen Properties und Klassen

- **Vorsicht**

- Angaben zu Domain und Range gelten **global**
 - Also überall, wo die Property verwendet wird
- Angaben zu Domain und Range sind **konjunktiv**
- Beispiel:

```
ex:istAutorVon      rdfs:range      ex:Kochbuch .  
ex:istAutorVon      rdfs:range      ex:Maerchenbuch .
```

führt dazu, dass jedes Buch von dem jemand Autor ist, gleichzeitig Kochbuch UND Märchenbuch ist!!!

- Immer die Klasse verwenden, die am allgemeinsten ist!

Sprachen des Semantic Web – RDFS

- Offene Listen mit RDFS

- **rdfs:Container**

- Oberklasse von `rdf:Seq`, `rdf:Bag` und `rdf:Alt` (und allen weiteren Klassen, die man als Container definiert)

- **rdfs:ContainerMembershipProperty**

- Klasse aller Properties, die eine Beziehung beschreiben, welche die Semantik besitzt, dass das Subjekt im Objekt enthalten ist
 - Für alle Instanzen (Properties) gilt: Range ist `rdfs:Container`

- **rdfs:member**

- Oberproperty aller in `rdfs:ContainerMembershipProperty` enthaltenen Instanzen
 - Wenn gilt

`aProperty` `rdfs:subPropertyOf` `rdfs:member`

folgt daraus:

`aProperty` `rdf:type` `rdfs:ContainerMembershipProperty`

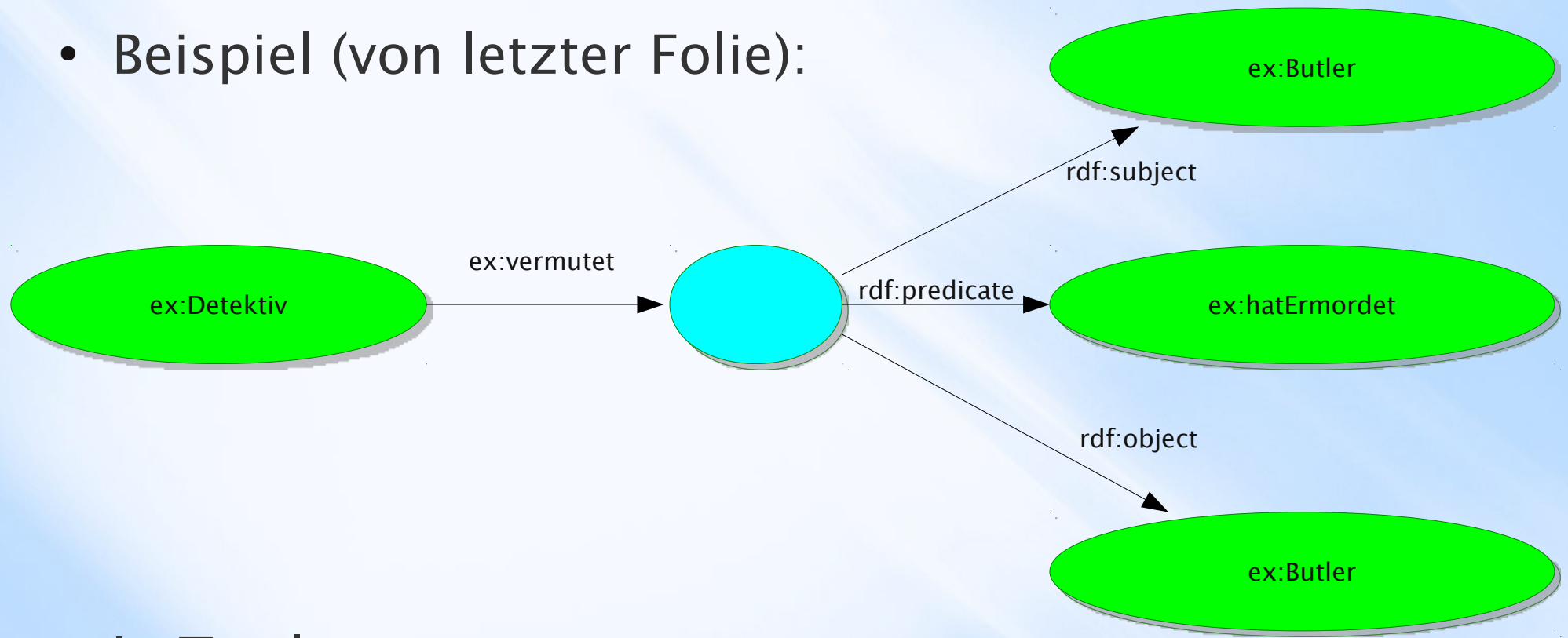
und umgekehrt

- Reifikation

- Aussagen über Triple
- Hinweiswort “dass”
- Beispiel:
 - Der Detektiv vermutet, **dass** der Butler den Gärtner ermordet hat.
- Oft verwendet, um Ungewissheiten auszudrücken
- Die “reifizierten” Triple müssen nicht unbedingt gelten
 - Beachten: Wenn ein Triple einmal definiert ist, gilt es immer
- Typisierung reifizierter Aussagen als `rdf:Statement`
- Elemente der Aussage:
 - `rdf:subject`, `rdf:predicate`, `rdf:object`
- Statements werden oft als Blank Nodes realisiert

Sprachen des Semantic Web – RDFS

- Beispiel (von letzter Folie):



- In Turtle:

```
ex:Detektiv ex:vermutet [ ex:subject ex:Butler
                          ex:predicate ex:hatErmordet
                          ex:object   ex:Gaertner ] .
```


Sprachen des Semantic Web – RDFS

- Annotations
 - Beim Programmieren kommentiert man Code, bei der Ontologierstellung annotiert man sein Vokabular und Ressourcen
 - Ziel: Erhöhung der Verständlichkeit für den Nutzer
 - Wird ebenfalls als Graph notiert – RDFS bietet spezielle Properties
 - `rdfs:label`
 - Vergabe eines lesbaren Namens für eine Ressource
 - Wird oft von Tools anstelle der (abgekürzten) URI angezeigt
 - `rdfs:comment`
 - Möglichst umfangreicher Kommentar zu einer Ressource
 - z.B. Natürlichsprachliche Definition einer neuen Klasse
 - “Die Klasse `ex:Bank` bezeichnet ein Geldinstitut”
 - `rdfs:seeAlso`, `rdfs:definedBy`
 - URIs, wo man weitere Informationen über die Ressource finden kann

Sprachen des Semantic Web – RDFS

- Beispiel zu Annotationen:

```
<rdfs:Class rdf:about="&ex;Car">
  <rdfs:label xml:lang="en">Car</rdfs:label>
  <rdfs:label xml:lang="de">Auto</rdfs:label>
  <rdfs:comment xml:lang="de">
    Ein Automobil, kurz Auto (auch Kraftwagen, früher
    Motorwagen), ist ein mehrspuriges Kraftfahrzeug, das
    von einem Motor angetrieben wird und zur Beförderung von
    Personen und Frachtgütern dient.
  </rdfs:comment>
  <rdfs:seeAlso rdf:resource="http://de.wikipedia.org/wiki/Auto" />
  <rdfs:isDefinedBy rdf:resource="http://example.org/vocabulary" />
  <rdfs:subClassOf rdf:resource="&ex;MotorVehicle" />
</rdfs:Class>
```

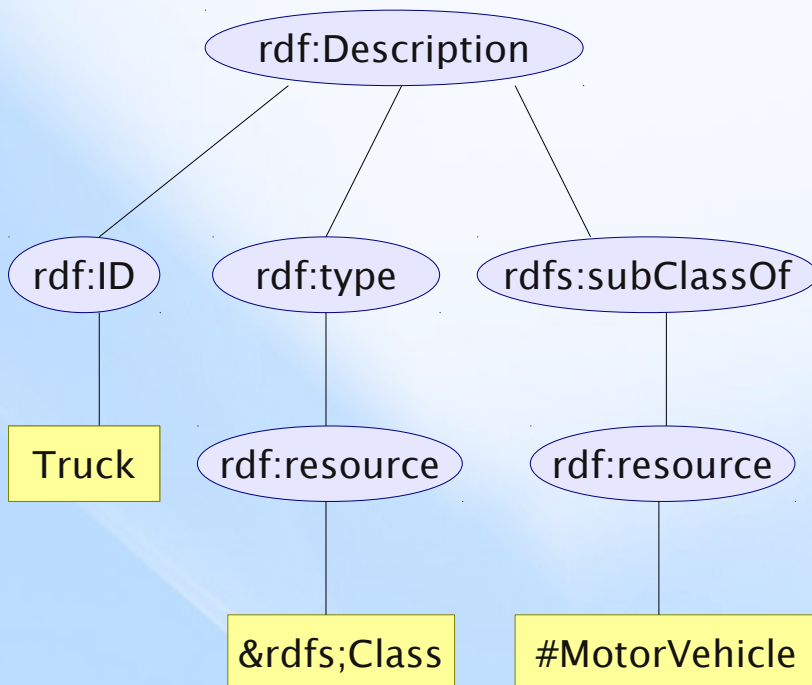
- Ontologien – A-Box / T-Box
 - Mit RDFS kann Wissen so modelliert werden, dass weiteres, implizites Wissen zu einem gewissen Teil bereits geschlussfolgert werden kann
 - Eine Ontologie ist die Beschreibung von Wissen
 - Man unterscheidet zwischen
 - A-Box: Assertionales Wissen
 - Aussagen über Individuen und deren Beziehung zueinander
 - Wird bereits durch RDF abgedeckt
 - T-Box: Terminologisches Wissen
 - Definition von Klassen, Properties und deren Beziehung untereinander
 - Wird mit RDFS modelliert

Sprachen des Semantic Web – RDFS

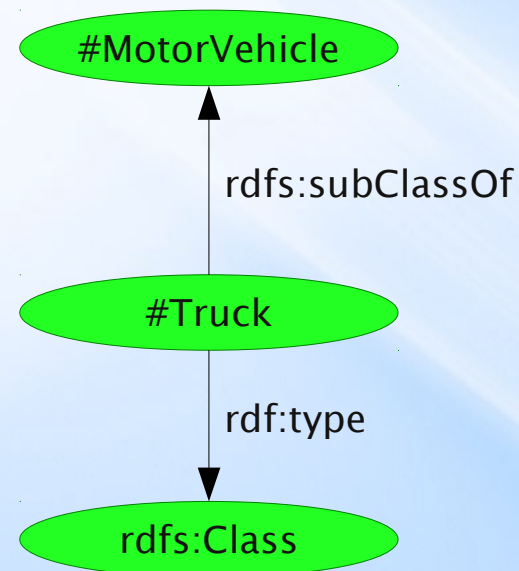
1 Dokument – 3 Interpretationen

```
<rdf:Description rdf:ID="Truck">  
  <rdf:type rdf:resource="&rdfs;Class" />  
  <rdfs:subClassOf rdf:resource="#MotorVehicle" />  
</rdf:Description>
```

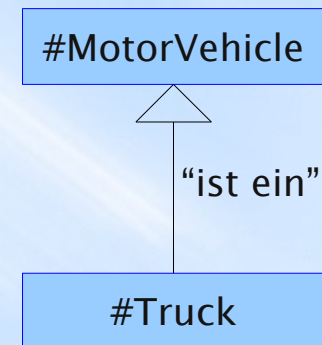
XML



RDF



RDFS



Noch Fragen ?

- Literatur:

- Buch “Semantic Web Grundlagen”, Springer Verlag 2008
Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, York Sure
ISBN: 978-3-540-33993-9
- RDF Webseite
<http://www.w3.org/RDF/>
- Resource Description Framework Schema
Dan Brickley, R.V. Guha, Brian McBride
W3C Recommendation, 10th February 2004
<http://www.w3.org/TR/rdf-schema>
- RDF Planet – Blogs zum Thema RDF / Semantic Web
<http://www.planetrdf.com/>