



Semantic Web

Vorlesung

Dr. Harald Sack

Hasso-Plattner-Institut für Softwaresystemtechnik

Universität Potsdam

Wintersemester 2008/09



<http://sw0809.blogspot.com/>

Blog zur Vorlesung: <http://sw0809.blogspot.com/>

Semantic Web - Vorlesungsinhalt

2

1. Einführung
2. **Die Sprachen des Semantic Web**
3. Wissensrepräsentation
4. Web of Trust
5. Ontology Engineering
6. Semantic Web Anwendungen

1

2

3

4

4.12.2008 – Vorlesung Nr. 5

6

7

8

9

10

11

12

13

2. Die Sprachen des Semantic Web

2.1 Uniform Resource Identifier - URI

2.2 Universelle Vokabularien mit XML

2.3 Einfache semantische Strukturen mit RDF und RDFS

2.3.0 Problem der Wissensrepräsentation

2.3.1 RDF Sprachstandard

2.3.2 RDF Schema

2.4 Abfragesprache SPARQL

2.5 Semantik im WWW - μ formats, RDFa, GRDDL

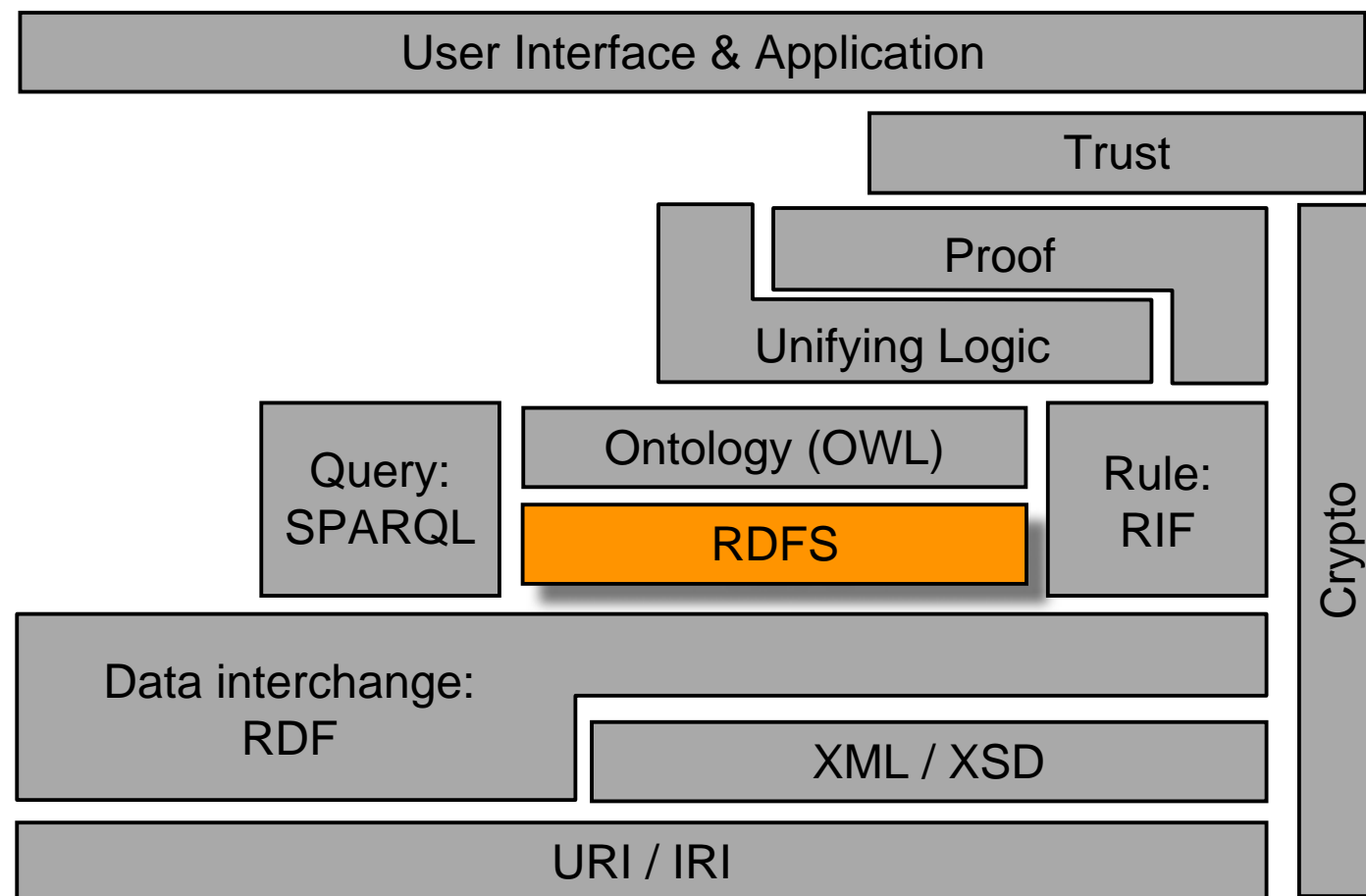
2.6 Warum RDF/RDFS noch nicht ausreicht

2. Die Sprachen des Semantic Web

2.3 RDF und RDFS / 2.3.2 RDF Schema

4

Semantic Web Architecture



2. Die Sprachen des Semantic Web

2.3 RDF und RDFS / 2.3.2 RDF Schema

5

Motivation für RDF Schema

- Programm kann RDF-Daten definieren und nutzen....vorausgesetzt, dem Programm ist bekannt, welche Terme benutzt werden können...
 - z.B.
 - name, titel, jahr, ...
 - name, blog, telefonnr...
 - author, cites, ...

- Sind alle bekannt?
Sind alle korrekt?
Gibt es (logische) Beziehungen zwischen den Termen?

- Wir benötigen eine entsprechende Datendefinition → **RDF Schema**
 - offiziell: “**RDF Vocabulary Description Language**”

2. Die Sprachen des Semantic Web

2.3 RDF und RDFS / 2.3.2 RDF Schema

6

RDF Schema

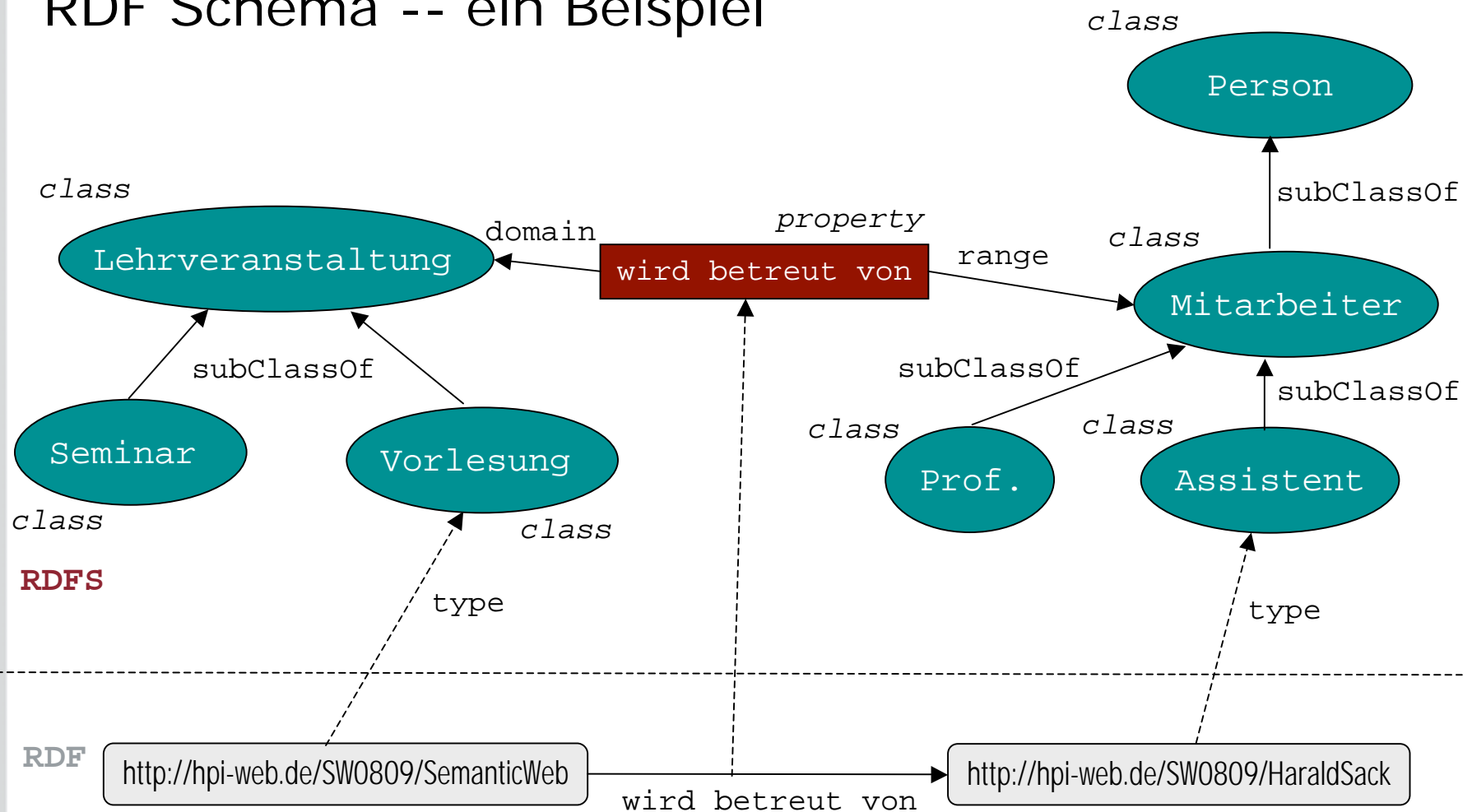
- RDFSchema spezifiziert ein Datenmodell, über das RDF-Statements entworfen werden können
 - Abstrakter Datentyp (Klasse)
 - Hierarchisches Klassenmodell und Vererbung
 - Syntax für gemeinsamen Datenaustausch
- = „*Beschreibungssprache zur Definition strukturierter Vokabularien...*“
- RDFSchema erlaubt:
 - Klassendefinitionen
 - Klasseninstantiierung in RDF via `<rdf:type>`
 - Festlegung von Eigenschaften und Restriktionen
 - Festlegung von Hierarchiebeziehungen
 - Subklassen und Superklassen

2. Die Sprachen des Semantic Web

2.3 RDF und RDFS / 2.3.2 RDF Schema

7

RDF Schema -- ein Beispiel



2. Die Sprachen des Semantic Web

2.3 RDF und RDFS / 2.3.2 RDF Schema

8

RDFS Sprachkomponenten

■ Klassen

- **rdfs:Resource**
jede Entität in einem RDF-Modell ist Instanz dieser Klasse
- **rdf:Property**
Basisklasse für Eigenschaften
- **rdfs:Class**
Klassenkonzept, legt ein abstraktes Objekt fest und dient in Verbindung mit `rdf:type` zur Erzeugung von Instanzen
- **rdfs:Literal**
Klasse für Literalwerte, also Zeichenketten, ...
- zusätzlich noch
rdfs:Datatype, **rdfs:XMLLiteral**, **rdfs:Container**,
rdfs:ContainerMembershipProperty

2. Die Sprachen des Semantic Web

2.3 RDF und RDFS / 2.3.2 RDF Schema

9

RDFS Sprachkomponenten

■ Eigenschaften

□ **rdfs:subClassOf**

transitive Eigenschaft zur Festlegung von Vererbungshierarchien von Klassen

□ **rdfs:subPropertyOf**

transitive Eigenschaft zur Festlegung von Vererbungshierarchien von Eigenschaften

□ **rdfs:domain**

legt Anwendungsbereich einer Eigenschaft in Bezug auf eine Klasse fest

□ **rdfs:range**

legt Wertebereich einer Eigenschaft fest

RDFS Sprachkomponenten

■ Eigenschaften

□ **rdfs:subClassOf**

transitive Eigenschaft zur Festlegung von Vererbungshierarchien von Klassen

$$(1) \forall i, c_1, c_2 : T(i, \text{rdf:type}, c_1) \wedge T(c_1, \text{rdfs:subClassOf}, c_2) \\ \Rightarrow T(i, \text{rdf:type}, c_2)$$

$$(2) \forall c_1, c_2, c_3 : T(c_1, \text{rdfs:subClassOf}, c_2) \wedge T(c_2, \text{rdfs:subClassOf}, c_3) \\ \Rightarrow T(c_1, \text{rdfs:subClassOf}, c_3)$$

$$(3) \forall c : \neg T(c, \text{rdfs:subClassOf}, c)$$

RDFS Sprachkomponenten

▪ Eigenschaften

□ **rdfs:subPropertyOf**

transitive Eigenschaft zur Festlegung von Vererbungshierarchien von Eigenschaften

$$(1) \forall s, p_1, o, p_2 : T(s, p_1, o) \wedge T(p_1, \text{rdfs:subPropertyOf}, p_2) \\ \Rightarrow T(s, p_2, o)$$

$$(2) \forall p_1, p_2, p_3 : T(p_1, \text{rdfs:subPropertyOf}, p_2) \wedge T(p_2, \text{rdfs:subPropertyOf}, p_3) \\ \Rightarrow T(p_1, \text{rdfs:subPropertyOf}, p_3)$$

$$(3) \forall p : \neg T(p, \text{rdfs:subPropertyOf}, p)$$

2. Die Sprachen des Semantic Web

2.3 RDF und RDFS / 2.3.2 RDF Schema

13

RDFS Sprachkomponenten:

■ Weitere Eigenschaften

□ **rdfs:seeAlso**

stellt Beziehung einer Ressource zu einer anderen her, die diese erklärt

□ **rdfs:isDefinedBy**

subproperty zu `rdf:seeAlso`, stellt Beziehung von Resource zu deren Definition (auch in RDFS) her

□ **rdfs:comment**

Kommentar, gewöhnlich in Textform

□ **rdfs:label**

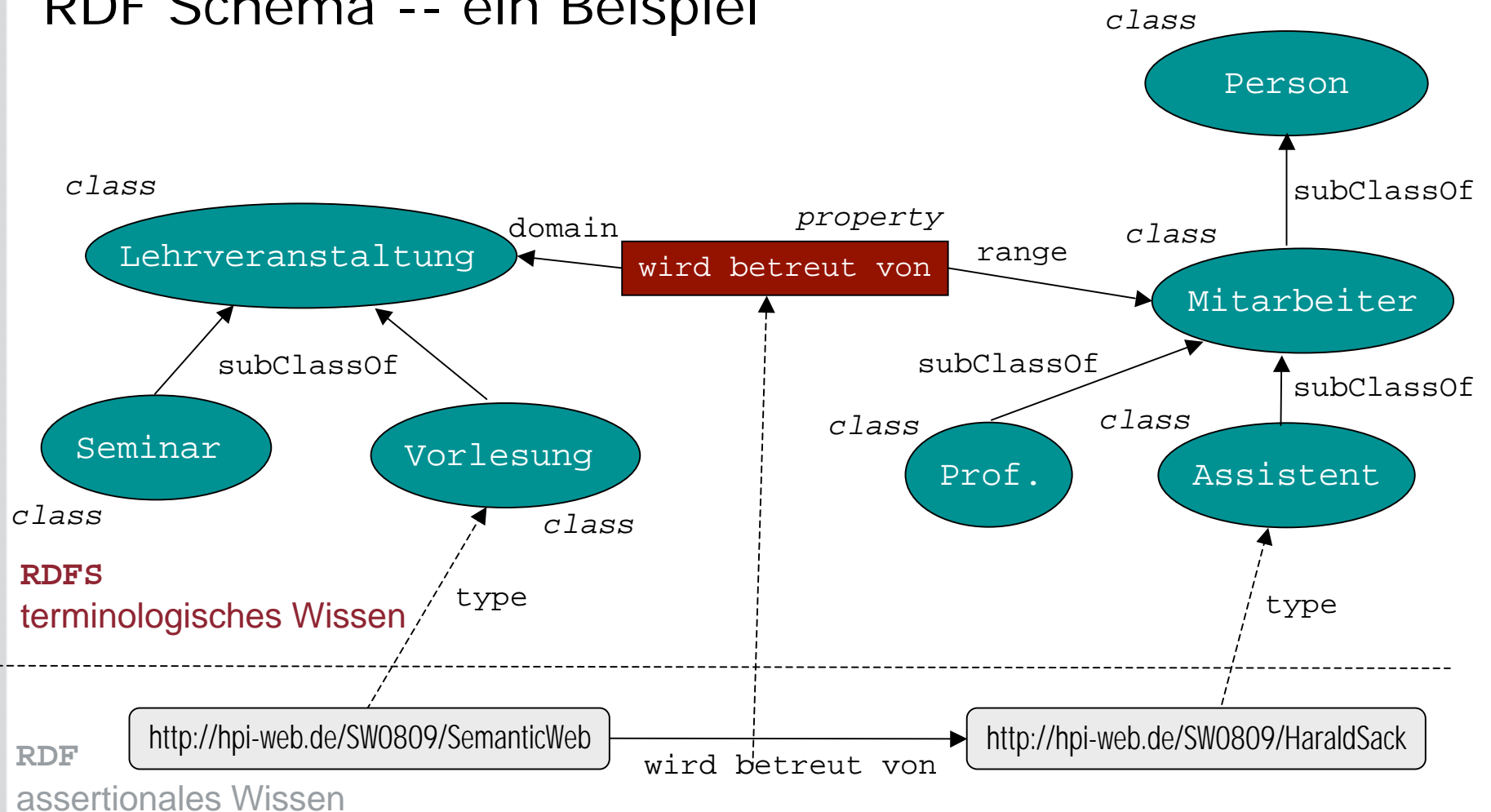
„lesbarer“ Name einer Resource (im Gegensatz zu ID)

2. Die Sprachen des Semantic Web

2.3 RDF und RDFS / 2.3.2 RDF Schema

14

RDF Schema -- ein Beispiel



```
<?xml version="1.0" encoding="utf-8"?>

<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://hpi-web.de/WS0809">

  <rdfs:Class rdf:ID="Lehrveranstaltung">
    <rdfs:ClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdfs:Class>
  <rdfs:Class rdf:ID="Vorlesung">
    <rdfs:SubClassOf rdf:resource="#Lehrveranstaltung"/>
  </rdfs:Class>
  <rdfs:Class rdf:ID="Seminar">
    <rdfs:SubClassOf rdf:resource="#Lehrveranstaltung"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="Person">
    <rdfs:SubClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdfs:Class>
  <rdfs:Class rdf:ID="Mitarbeiter">
    <rdfs:SubClassOf rdf:resource="#Person"/>
  </rdfs:Class>
  <rdfs:Class rdf:ID="Assistent">
    <rdfs:SubClassOf rdf:resource="#Mitarbeiter"/>
  </rdfs:Class>
  ...
```

2. Die Sprachen des Semantic Web

2.3 RDF und RDFS / 2.3.2 RDF Schema

```
<rdf:Property rdf:ID="wirdBetreutVon">  
  <rdfs:SubPropertyOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>  
  <rdfs:domain rdf:resource="#Lehrveranstaltung"/>  
  <rdfs:range rdf:resource="#Mitarbeiter"/>  
</rdf:Property>
```

```
<rdf:Description rdf:about="http://hpi-web.de/WS0809/semanticweb">  
  <rdf:type rdf:resource="#Vorlesung"/>  
</rdf:Description>  
<rdf:Description rdf:about="http://hpi-web.de/WS0809/HaraldSack">  
  <rdf:type rdf:resource="#Assistent"/>  
</rdf:Description>
```

```
<rdf:Description rdf:about="http://hpi-web.de/WS0809/semanticweb">  
  <wirdBetreutVon rdf:resource="http://hpi-web.de/WS0809/HaraldSack"/>  
</rdf:Description>  
...  
</rdf:RDF>
```

2. Die Sprachen des Semantic Web

2.3 RDF und RDFS / 2.3.2 RDF Schema

17

RDFS Zusammenfassung:

- RDFSchema spezifiziert ein Datenmodell, über das RDF-Statements entworfen werden können

- Mehr als XML:
 - (kleine) ontologische Einigung auf Modellierungsprimitive
 - Möglichkeit eigene Vokabulare zu definieren

- Nächste Schritte:
 - Logik
 - Regeln

1

2

3

4

4.12.2008 – Vorlesung Nr. 5

6

7

8

9

10

11

12

13

2. Die Sprachen des Semantic Web

2.1 Uniform Resource Identifier - URI

2.2 Universelle Vokabularien mit XML

2.3 Einfache semantische Strukturen mit RDF und RDFS

2.4 Abfragesprache SPARQL

2.5 Semantik im WWW - μ formats, RDFa, GRDDL

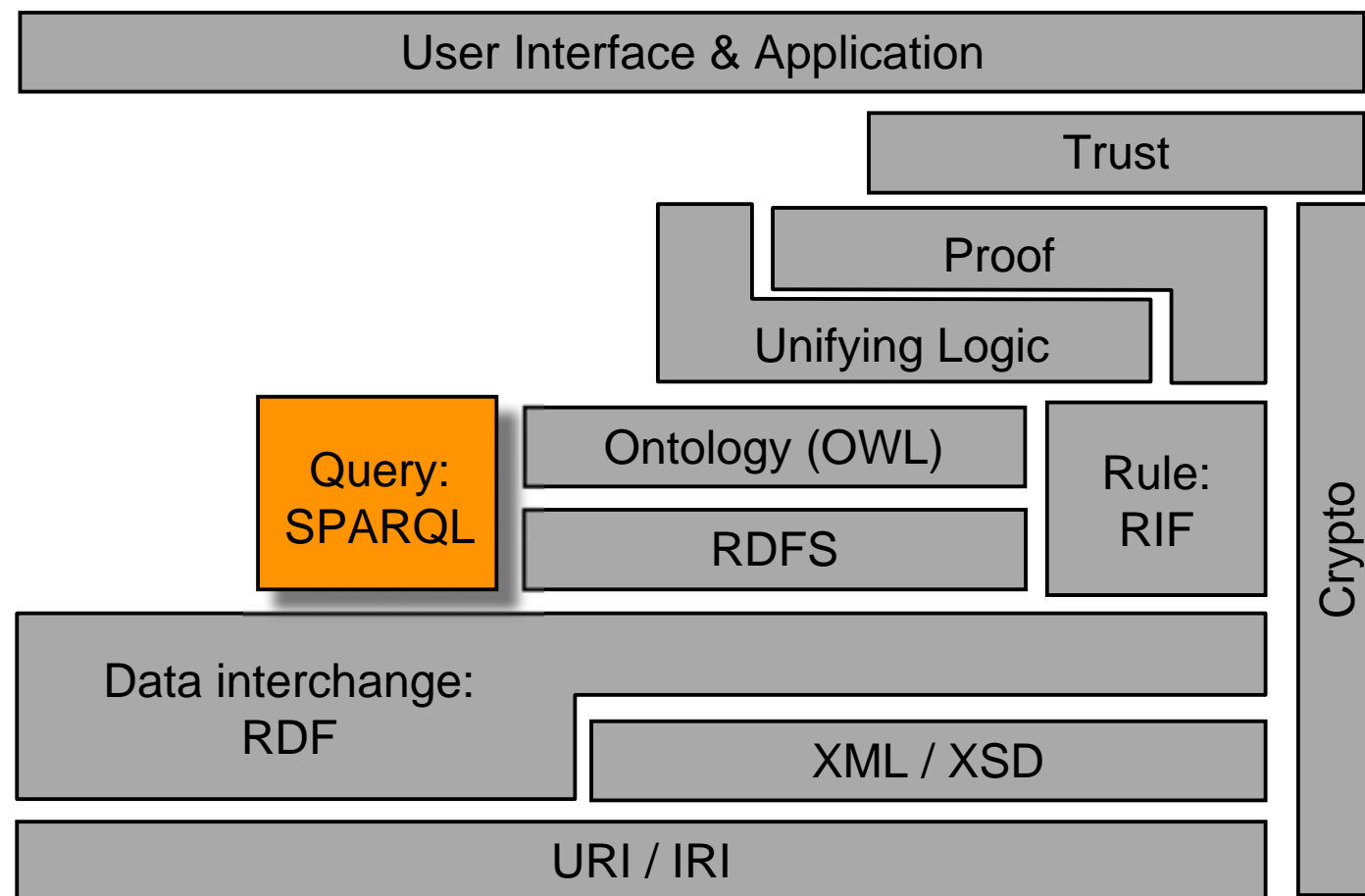
2.6 Warum RDF/RDFS noch nicht ausreicht

2. Die Sprachen des Semantic Web

2.3 RDF und RDFS / 2.3.2 RDF Schema

19

Semantic Web Architecture



2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

20

Eine Abfragesprache für RDF

- Wie gelangt man an das Wissen aus RDF/RDFS Wissensbasen?
 - Manuelles Parsen/Auslesen und Kombinieren der RDF-Tripel sehr aufwändig
- Können Sie sich eine relationale Datenbank ohne SQL vorstellen?
- *SPARQL Protocol and RDF Query Language* ist
 - eine Abfragesprache zur Traversierung von RDF-Graphen
 - ein Protokoll-Layer, um SPARQL, z.B. via http zu nutzen
 - ein XML-Rückgabeformat für SPARQL-Anfragen
 - ein W3C Standard (seit Januar 2008)
 - angelehnt an die Abfragesprache SQL

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

21

SPARQL

- SPARQL erlaubt
 - die Extraktion von Werten aus strukturierten und semi-strukturierten Daten
 - die Erkundung von Daten über die Abfrage unbekannter Beziehungsstrukturen
 - die Durchführung komplexer Join-Operationen über heterogene, verteilte Datenbanken in einer einzigen, einfachen Abfrage
 - die Transformation von RDF-Daten von einem Vokabular in ein anderes

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

22

SPARQL

- SPARQL Variablen werden an RDF-Terme gebunden
 - z.B. *?journal, ?disease, ?price*
- Wie in SQL erfolgt die Abfrage von Variablen über ein SELECT Statement
 - z.B. *SELECT ?article ?author ?published*
- Ein SELECT-Statement liefert Abfrageergebnisse in Form einer Tabelle

?artist	?album	?times_platinum
Michael Jackson	Thriller	27
Led Zepellin	Led Zepellin IV	22
Pink Floyd	The Wall	22

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL



23

Triple Pattern

- Ein Triple Pattern ist ein RDF-Triple, das an beliebiger Stelle (Subject, Predicate, Object) Variablen beinhalten kann.

- Beispiele:
 - Finde Länder und ihre Hauptstädte:
 - `?country geo:capital ?capital .`
 - Ausgehend von einem FOAF URI, finde Namen einer Person:
 - `<http://hpi-web.de/id#haraldsack> foaf:name ?surname .`
 - Welche Personen tragen den Nachnamen „Schmidt“?
 - `?person pers:nachname "Schmidt" .`

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

24

Einfache Abfrage Pattern

- Triple Pattern können zu komplexen Suchabfragen kombiniert werden, mit denen ein RDF-Graph auf einfache Weise traversiert werden kann.
 - Finde Länder, ihre Hauptstädte und die Bevölkerungszahl:
 - `?country geo:capital ?capital .`
`?country geo:population ?population .`
 - Ausgehend von einem FOAF URI, finde den Namen einer Person und ihrer Freunde:
 - `< http://hpi-web.de/id#haraldsack > foaf:name ?surname .`
`< http://hpi-web.de/id#haraldsack > foaf:knows ?friend .`
`?friend foaf:name ?friend_?surname .`
 - Finde alle Vorgesetzten, der 3. Managementebene
 - `?emp hr:managedBy ?first_line .`
`?first_line hr:managedBy ?second_line .`
`?second_line hr:managedBy ?third_line .`

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

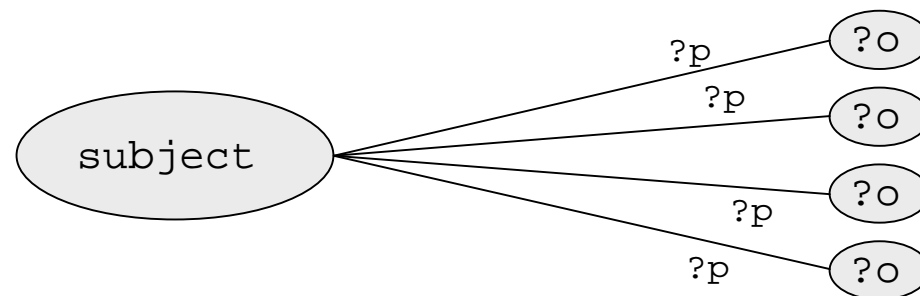
25

Allgemeines SPARQL Abfrageformat

- angelehnt an SQL

```
SELECT ?p, ?o  
WHERE { subject ?p ?o. }
```

- Triple im „WHERE“-Teil definieren Graph-Abfrage mit ?p und ?o als Variablen
- Abfrage liefert Tabelle mit passenden ?p, ?o Paaren



2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

26

SPARQL Abfrageformat

- Gebe alle Lehrveranstaltungen mit den zugehörigen Dozenten aus

```
PREFIX hpi: <http://hpi-web.de/WS0809#>
SELECT ?vorlesung ?dozent
FROM <http://hpi-web.de/WS0809.rdf>
WHERE {
    ?x hpi:lehveranstaltung ?vorlesung .
    ?y hpi:mitarbeiter ?dozent .
    ?x hpi:wirdBetreutVon ?y .
}
```

- **PREFIX** - Legt einen oder mehrere Namespaces fest
- **FROM** - gibt ein oder mehrere RDF-Quellgraphen an
- **BASE** - legt Basis-URI fest, um relative URIs zu absoluten URIs zu ergänzen

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

27

SPARQL Abfrageformat

- Gebe alle Lehrveranstaltungen mit den zugehörigen Dozenten aus, absteigend nach Dozenten geordnet, und zwar nur die Ergebnisse Nummer 11 bis 20

```
PREFIX hpi: <http://hpi-web.de/WS0809#>
SELECT ?vorlesung ?dozent
FROM <http://hpi-web.de/WS0809.rdf>
WHERE {
    ?x hpi:lehrveranstaltung ?vorlesung .
    ?y hpi:mitarbeiter ?dozent .
    ?x hpi:wirdBetreutVon ?y .
}
ORDER BY DESC (?dozent)
LIMIT 10
OFFSET 10
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

28

SPARQL Abfrageformat

- Über das Schlüsselwort FILTER kann die Ergebnismenge eingeschränkt werden

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
```

```
@prefix : <http://example.org/book/> .
```

```
@prefix ns: <http://example.org/ns#> .
```

```
:book1 dc:title "SPARQL Tutorial" .
```

```
:book1 ns:price 42 .
```

```
:book2 dc:title "The Semantic Web" .
```

```
:book2 ns:price 23 .
```

```
http://example.org/book.rdf
```

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
PREFIX ns: <http://example.org/ns#>
```

```
SELECT ?title ?price
```

```
FROM <http://example.org/book.rdf>
```

```
WHERE {
```

```
  ?x ns:price ?price .
```

```
  FILTER (?price < 30.5)
```

```
  ?x dc:title ?title .
```

```
}
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

29

SPARQL Abfrageformat

- Über das Schlüsselwort **OPTIONAL** können optionale Elemente aus dem RDF-Graphen selektiert werden

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
_:a rdf:type foaf:Person .
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@example.com> .
_:a foaf:mbox <mailto:alice@work.example> .
```

```
_:b rdf:type foaf:Person .
_:b foaf:name "Bob" .
```

<http://example.org/adressen.rdf>

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
FROM <http://example.org/adressen.rdf>
WHERE { ?x foaf:name ?name .
         OPTIONAL { ?x foaf:mbox ?mbox }
       }
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

30

SPARQL Abfrageformat

- Über das Schlüsselwort UNION können Abfragen aus mehreren Graph-Pattern kombiniert werden (ODER-Verknüpfung)

```
@prefix dc10: <http://purl.org/dc/elements/1.0/> .  
@prefix dc11: <http://purl.org/dc/elements/1.1/> .
```

```
_:a dc10:title "SPARQL Query Language Tutorial" .  
_:a dc10:creator "Alice" .
```

```
_:b dc11:title "SPARQL Protocol Tutorial" .  
_:b dc11:creator "Bob" .
```

```
_:c dc10:title "SPARQL" .  
_:c dc11:title "SPARQL (updated)" .
```

```
http://example.org/books/
```

```
PREFIX dc10: <http://purl.org/dc/elements/1.0/>  
PREFIX dc11: <http://purl.org/dc/elements/1.1/>
```

```
SELECT ?title  
FROM <http://example.org/books.rdf>  
WHERE { { ?book dc10:title ?title } UNION {  
        ?book dc11:title ?title }  
}
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

31

SPARQL Abfrageformat

- Kombination mehrerer Datenquellen:
 - Default RDF-Graph
 - Named RDF-Graph
- Named Graph kann explizit über das Schlüsselwort GRAPH angesprochen werden

```
GRAPH <http://example.org/graph1.rdf> {  
    ?x foaf:mbox ?mbox  
}
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

32

SPARQL Abfrageformat

- Verwendung von Named Graphs

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
```

```
<http://example.org/bob> dc:publisher "Bob Hacker" .
```

```
<http://example.org/alice> dc:publisher "Alice Hacker" .
```

Default Graph

(stored at <http://example.org/dft.ttl>)

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

```
_:a foaf:name "Bob" .
```

```
_:a foaf:mbox <mailto:bob@oldcorp.example.org> .
```

Named Graph: <http://example.org/bob>

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

```
_:a foaf:name "Alice" .
```

```
_:a foaf:mbox <mailto:alice@work.example.org> .
```

Named Graph: <http://example.org/alice>

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

33

SPARQL Abfrageformat

- Verwendung von Named Graphs

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
SELECT ?who ?g ?mbox  
FROM <http://example.org/dft.ttl>  
FROM NAMED <http://example.org/alice>  
FROM NAMED <http://example.org/bob>  
WHERE  
{  
  ?g dc:publisher ?who .  
  GRAPH ?g { ?x foaf:mbox ?mbox }  
}
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

34

SPARQL Antwortformat

- SPARQL Ergebnisse werden in Form wohlgeformter und valider XML-Dokumente erbracht

```
<?xml version="1.0"?>  
<sparql xmlns="http://www.w3.org/2005/sparql-results#">  
...  
</sparql>
```

- In einem <head>-Element werden alle Variablen der SPARQL-Query aufgelistet

```
<head>  
  <variable name="x"/>  
  <variable name="hpage"/>  
  <variable name="name"/>  
  <variable name="mbox"/>  
  <variable name="blurb"/>  
</head>
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

35

SPARQL Antwortformat

- Für jedes SPARQL-Query Ergebnis steht ein `<result>`-Element

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="x"/>
    ...
  </head>
  <results>
    <result>
      <binding name="x"> ... </binding>
      <binding name="hpage"> ... </binding>
    </result>

    <result> ... </result>

    ...
  </results>
</sparql>
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

36

SPARQL Antwortformat

- Im `<binding>`-Element erfolgt die Bindung einer `<head>`-Variablen an einen Ergebniswert

```
<result>
  <binding name="x">
    <bnode>r2</bnode>
  </binding>
  <binding name="hpage">
    <uri>http://work.example.org/bob/</uri>
  </binding>
  <binding name="name">
    <literal xml:lang="en">Bob</literal>
  </binding>
  <binding name="age">
    <literal datatype="http://www.w3.org/2001/XMLSchema#integer">30</literal>
  </binding>
  <binding name="mbox">
    <uri>mailto:bob@work.example.org</uri>
  </binding>
</result>
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

37

SPARQL Abfrageformat

- Neben der SELECT-Abfrage erlaubt SPARQL noch folgende Abfrageformate:
 - ASK
 - liefert Boolesche Antwort („*hat die Abfrage ein Ergebnis?*“)
 - Antwort wie bei SELECT als XML oder JSON
 - CONSTRUCT
 - nutzt Variablenbindung, um neue RDF-Tripel zurückzuliefern
 - Antwort als RDF/XML oder Turtle
 - DESCRIBE
 - liefert Server-generiertes RDF zur Beschreibung der angefragten Ressourcen
 - Antwort als RDF/XML oder Turtle

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

38

SPARQL Konstruktion von RDF-Graphen

- CONSTRUCT definiert ein Template zur Konstruktion neuer RDF-Graphen

```
@prefix org: <http://example.com/ns#> .
```

```
_:a org:employeeName "Alice" .
```

```
_:a org:employeeid 12345 .
```

```
_:b org:employeeName "Bob" .
```

```
_:b org:employeeid 67890 .
```

```
data
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
PREFIX org: <http://example.com/ns#>
```

```
CONSTRUCT { ?x foaf:name ?name }
```

```
WHERE { ?x org:employeeName ?name }
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

39

SPARQL Konstruktion von RDF-Graphen

- Ergebnis einer CONSTRUCT-Query als serialisiertes RDF/XML

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  >
  <rdf:Description>
    <foaf:name>Alice</foaf:name>
  </rdf:Description>
  <rdf:Description>
    <foaf:name>Bob</foaf:name>
  </rdf:Description>
</rdf:RDF>
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

40

SPARQL Protokoll

- Methode zur Anfrage/Antwort von SPARQL Queries über http
- Eine SPARQL URI besteht aus drei Teilen:
 1. Die URL eines SPARQL Endpoints
(z.B. `http://example.org/sparql`)
 2. Die abzufragenden RDF-Graphen
(*Optional, als Teil des Query-Strings*,
z.B. `named-graph-uri=http://example.org/testrdf.rdf`)
 3. Der Query-String selbst
(*Als Teil des Query-Strings*, z.B. `query=SELECT...`)

```
http://example.org/sparql?named-graph-uri=http%3A%2F%2Fexample.org%2F
reviews%2Febert&query=SELECT+%3Freview_graph+WHERE+%7B%0D%0A++GRAPH+%3Frev
iew_graph+%7B%0D%0A+++++%3Freview+rev%3Arating+10+.%0D%0A++%7D%0D%0A%7D
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

41

SPARQL Protokoll -- Beispiel

- Einfache SPARQL Query

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>  
SELECT ?book ?who  
WHERE { ?book dc:creator ?who }
```

- HTTP Trace der SPARQL Query

```
GET /sparql/?query=EncodedQuery&default-graph-uri=http://www.other.example/books HTTP/1.1  
Host: www.other.example  
User-agent: my-sparql-client/0.1
```

2. Die Sprachen des Semantic Web

2.4 Abfragesprache SPARQL

42

- HTTP Trace der SPARQL Antwort

```
HTTP/1.1 200 OK
Date: Fri, 06 May 2008 20:55:12 GMT
Server: Apache/1.3.29 (Unix) PHP/4.3.4 DAV/1.0.3
Connection: close
Content-Type: application/sparql-results+xml
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="book"/>
    <variable name="who"/>
  </head>
  <results ordered="false" distinct="false">
    <result>
      <binding name="who">
        <literal>Bob Hacker</literal>
      </binding>
      <binding name="book">
        <literal>The Art of Hacking</literal>
      </binding> ...
    </result>
  </sparql>
```



2. Die Sprachen des Semantic Web

2.1 Uniform Resource Identifier - URI

2.2 Universelle Vokabularien mit XML

2.3 Einfache semantische Strukturen mit RDF und RDFS

2.3.0 Problem der Wissensrepräsentation

2.3.1 RDF Sprachstandard

2.3.2 RDF Schema

2.4 Abfragesprache SPARQL

2.5 Semantik im WWW - μ formats, RDFa, GRDDL

2.6 Warum RDF/RDFS noch nicht ausreicht

Semantische Annotation im WWW

- Prinzipiell existieren heute zwei Möglichkeiten, strukturierte Daten mit semantischen Annotationen im WWW-Dokumente explizit einzubetten



Domain-spezifische Microformate



Generisches RDFa und embedded RDF

μformats -- Microformate

- XHTML Markup, mit dem es möglich ist, (in einem begrenzten Umfang) Semantik in einem HTML-Dokument auszudrücken
- Anwendungen können leichter Daten aus HTML-Dokumenten extrahieren
- Verwendung in folgenden XHTML-Tag **Attributen**:
 - **class**
 - **rel**
 - **rev**
- Vordefinierte Standard-Microformate:
 - hCard - Personendaten (→ vCard, RFC2426)
 - hCalendar – Events
 - rel-Tag – social tagging
 - XFN – XHTML Friends Network
 - ...

2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

46

μFormats -- Microformate

- Beispiel:

```
<span class="vcard">  
  <span class="fn">Joe Blow</span>  
  <span class="org">The Example Company</span>  
  <span class="tel">604-555-1234</span>  
  <a class="url" href="http://example.com/">  
    http://example.com/</a>  
</span>
```

2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

47

μFormats -- Einsatz

- [Last.fm](#)
Nutzerprofile und Geodaten, Veranstaltungen und CD-Bewertungen
- [Mac webmail](#)
Kontaktdaten
- [LinkedIn](#)
Lebensläufe
- [Flickr](#)
Kontaktdaten, Geodaten
- [WordPress](#)
Social Networking
- [Serendipity](#)
Bookmarks

2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

48

μ Formats -- Bewertung

- Microformate nutzen vorhandene XHTML-Attribute um strukturierte Daten zu kodieren
- Mit passenden XSLT Transformationen kann Microformat/XHTML in RDF umgewandelt werden
- Neues Microformat-Vokabular muss erst externe Konsolidierungsprozedur durchlaufen, bei der jedesmal auch ein neues XSLT Stylesheet entwickelt werden muss
- Wird mehr als ein Microformat-Vokabular in einer XHTML-Seite genutzt, steigt die Komplexität rapide
- Es können Konflikte mit genutzte XHTML-Attributen auftreten

2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

49



RDFa

- RDFa = **RDF** in HTML **a**tttributes
- bringt generische RDF-Annotationen in XHTML-Dokumente

- RDFa nutzt
 - bereits vorhandene XHTML-Attribute
 - » **href**, **content**, **rel**, **rev**, und **src**

 - neue XHTML-Attribute
 - » **about**, **datatype**, **property**, **resource** und **typeof**
(aus XHTML2 Metainformation Attributes)

2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

50



RDFa

- Unterscheidet grundsätzlich 2 Varianten von RDF-Triples
 - Triple mit URI als Objekt
 - Triple mit Literal als Objekt

	Subjekt	Prädikat	Objekt
Literal als Objekt	<code>about</code>	<code>property</code>	<code>content</code> oder <code>#PCDATA</code>
URI als Objekt	<code>about</code>	<code>rel</code>	<code>href</code>

2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

51



RDFa

- `...`
...erzeugt ein temporäres Subjekt für ein/mehrere Triples
- `...`
...definiert ein verlinktes Triple
- `object`
...definiert ein unverlinktes Triple

2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

52



RDFa -- einfaches Beispiel

```
...  
<div xmlns:dc="http://purl.org/dc/elements/1.1/">  
  I'm currently reading  
  <span about="urn:ISBN:0-596-00027-8" >  
    <span property="dc:title">Programming Perl</span>  
    by  
    <span property="dc:creator">Larry Wall</span>  
  </span>.  
</div>  
...
```

2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

53



RDFa -- einfaches Beispiel

```
...  
<div typeof="foaf:Person" xmlns:foaf="http://xmlns.com/foaf/0.1/">  
  <p property="foaf:name">  
    Alice Birpemschwick  
  </p>  
  <p> Email: <a rel="foaf:mbox" href="mailto:alice@example.com">  
    alice@example.com</a>  
  </p>  
  <p> Phone: <a rel="foaf:phone" href="tel:+1-617-555-7332">  
    +1 617.555.7332</a>  
  </p>  
  <p>  
      
  </p>  
</div>  
...
```

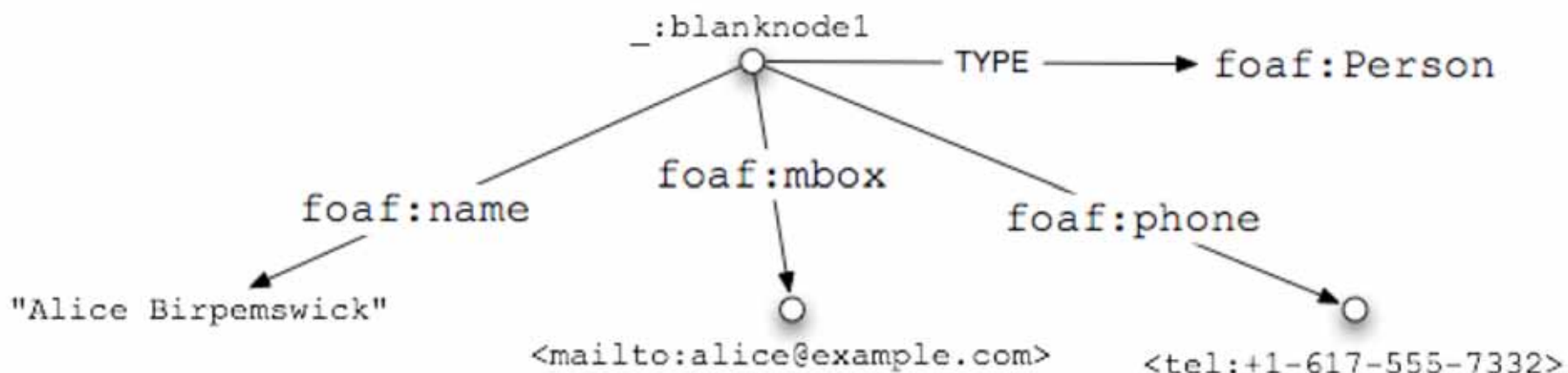
2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

54



RDFa -- einfaches Beispiel



2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

55



RDFa -- einfaches Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml" version="XHTML+RDFa 1.0"
  xml:lang="en">
  <head>...</head>
  <body>...</body>
</html>
</xml>
```

...und wie wird aus XHTML+RDFa dann RDF?

2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

56



GRDDL

- Gleaning Resource Descriptions from Dialects of Languages
- GRDDL bietet einen einfachen Weg, um RDF-Daten aus XML-Dokumenten (insbesondere XHTML) zu gewinnen
- GRDDL verbindet Transformationen (XSLT Stylesheets), um
 - RDFa
 - Microformats
 - RSS
- aus XHTML-Dokumenten herauszulösen

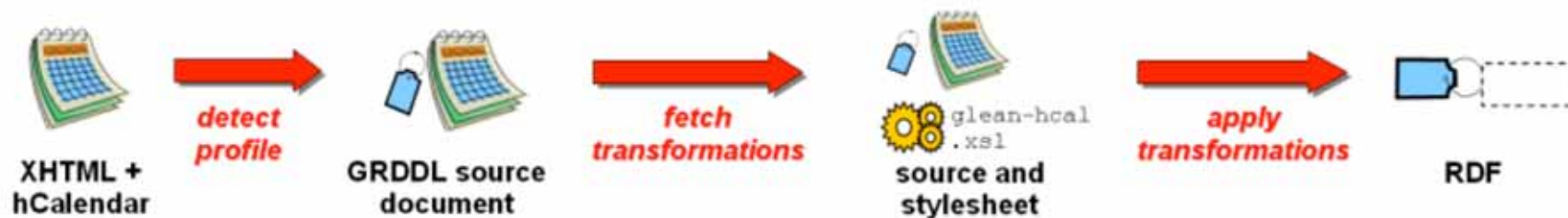
2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

57

GRDDL

- Um ein Semantische Annotationen aus einem Dokument zu extrahieren und in das RDF Format zu bringen, nutzt GRDDL folgende
- Schritte:
 1. Erkennung der GRDDL Profil Metadaten
 2. Beziehen der passenden Transformationsdaten
 3. Anwenden der Transformationen zum Konvertieren in das RDF Format



2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

58



GRDDL

GRDDL Profildaten

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head profile="http://www.w3.org/2003/g/data-view">
    ...
    <link rel="transformation" href="http://www.w3.org/2002/12/cal/glean-hcal" />
    <link rel="transformation" href="http://www.w3.org/2003/12/rdf-in-xhtml-xslts/grokFOAF.xsl" />
    <link rel="transformation" href="http://www.w3.org/2003/12/rdf-in-xhtml-xslts/grokCC.xsl" />
    <link rel="transformation" href="http://www.w3.org/2003/12/rdf-in-xhtml-xslts/grokGeoURL.xsl" />
    ...
  </head>
  ...
</html>
```

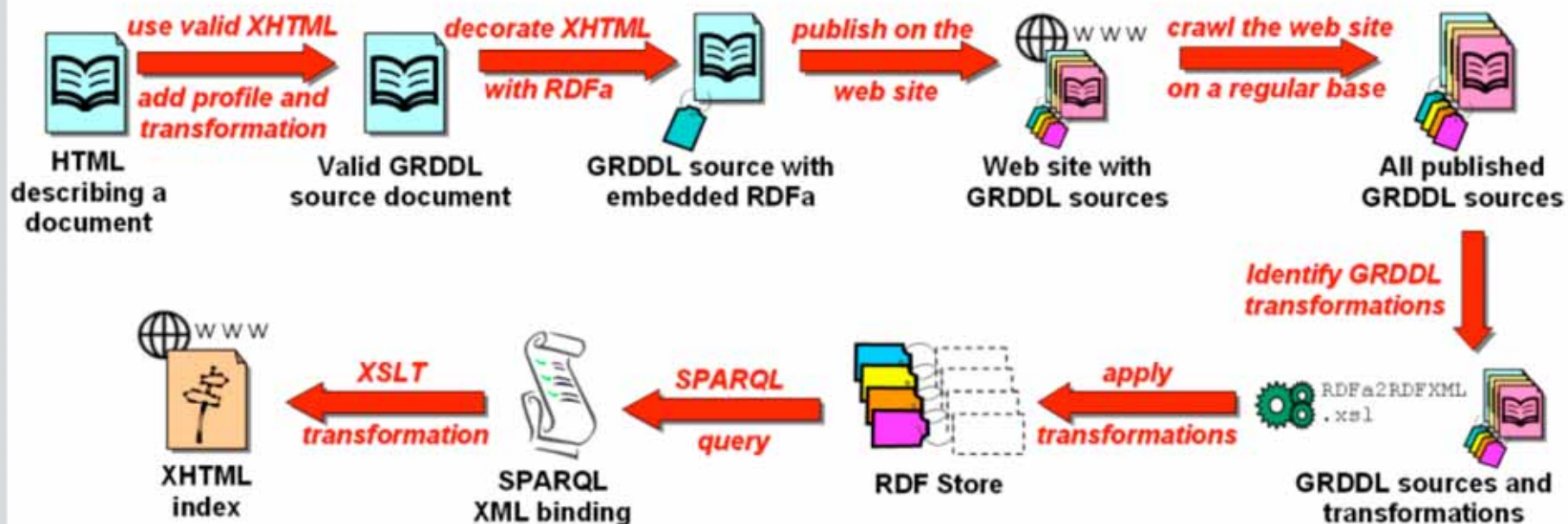
GRDDL Transformationen

2. Die Sprachen des Semantic Web

2.5 Semantik im WWW

59

GRDDL - Anwendungsbeispiel





2. Die Sprachen des Semantic Web

2.1 Uniform Resource Identifier - URI

2.2 Universelle Vokabularien mit XML

2.3 Einfache semantische Strukturen mit RDF und RDFS

2.3.0 Problem der Wissensrepräsentation

2.3.1 RDF Sprachstandard

2.3.2 RDF Schema

2.4 Abfragesprache SPARQL

2.5 Semantik im WWW - μ formats, RDFa, GRDDL

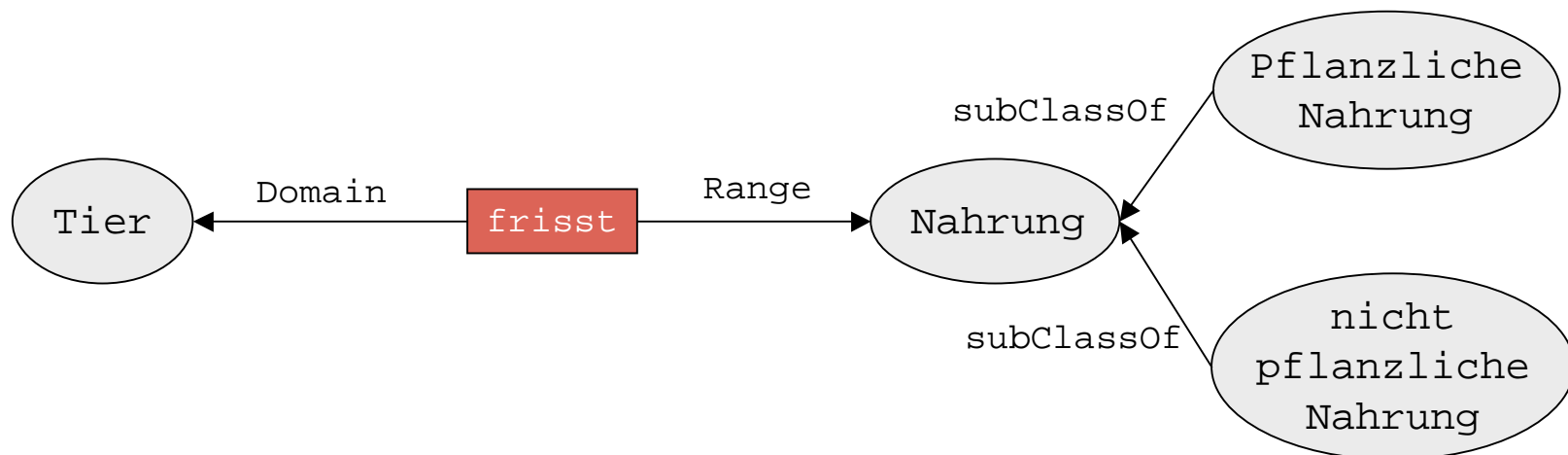
2.6 Warum RDF/RDFS noch nicht ausreicht

2. Die Sprachen des Semantic Web

2.6 Warum RDF immer noch nicht ausreicht...

61

Lokalität globaler Properties



Problem:

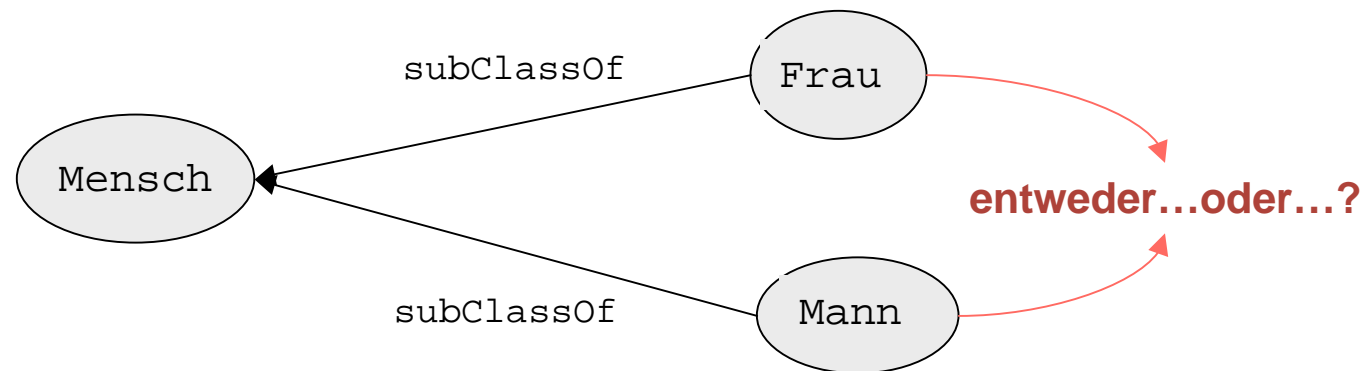
Kühe fressen nur pflanzliche Nahrung,
andere Tiere fressen nur Fleisch bzw. beides...

2. Die Sprachen des Semantic Web

2.6 Warum RDF immer noch nicht ausreicht...

62

Disjunktheit von Klassen



Problem:

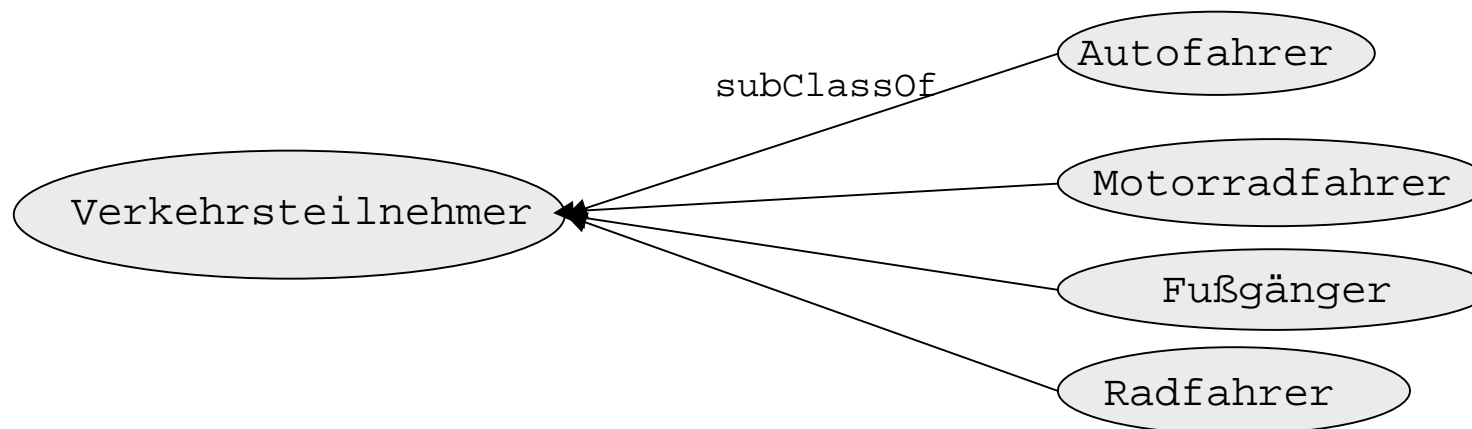
Subklassenbeziehung vermag nicht auszudrücken, dass Subklassenzugehörigkeit disjunkt sein kann

2. Die Sprachen des Semantic Web

2.6 Warum RDF immer noch nicht ausreicht...

63

Kombination von Klassen



Problem:

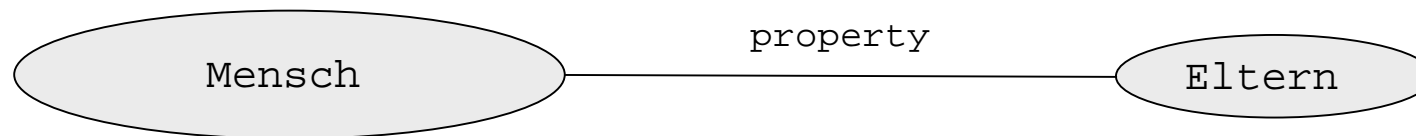
Alle Autofahrer, Motorradfahrer, Fußgänger und Radfahrer bilden zusammengenommen die Gruppe der Verkehrsteilnehmer

2. Die Sprachen des Semantic Web

2.6 Warum RDF immer noch nicht ausreicht...

64

Kardinalitätsrestriktionen



Problem:

Jeder Mensch hat in der Regel immer ZWEI Eltern

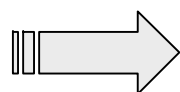
2. Die Sprachen des Semantic Web

2.6 Warum RDF immer noch nicht ausreicht...

65

Spezielle Eigenschaften von Properties

- Transitivität (z.B. „ist größer als“)
- Eineindeutigkeit (z.B. „ist Mutter von“)
- Inversivität (z.B. „ist Elternteil von“ und „ist Kind von“)



Wir benötigen zusätzliche Semantik !

2. Die Sprachen des Semantic Web

2.6 Warum RDF immer noch nicht ausreicht...

66

- Wissen muss derart repräsentiert werden, dass daraus neues Wissen hergeleitet werden kann
 - Schlussfolgerungsmechanismen (Inference Engines)

„Hans Mustermann ist ein Student“

&

**„alle Studenten sind an einer
Universität immatrikuliert“**



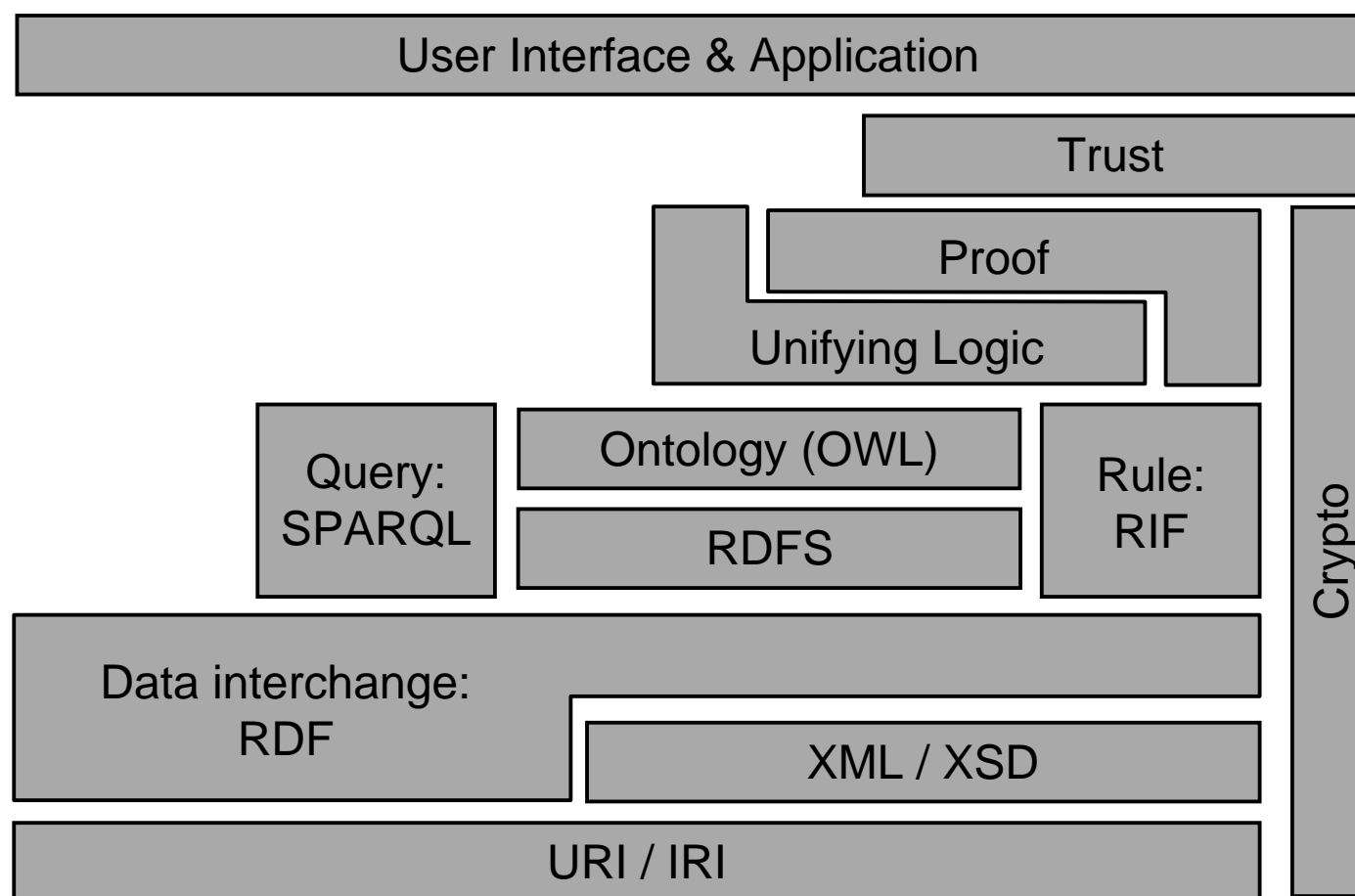
**„Hans Mustermann ist an einer
Universität immatrikuliert“**

2. Die Sprachen des Semantic Web

2.6 Warum RDF immer noch nicht ausreicht...

67

Semantic Web Architecture



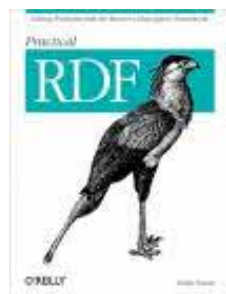
2. Die Sprachen des Semantic Web

68

Literatur



» P. Hitzler, M. Krötzsch, S. Rudolph, Y. Sure
[Semantic Web Grundlagen](#), Springer, 2008.



» S. Powers:
[Practical RDF](#), O'Reilly, 2003.

