

Wissensrepräsentation Vorlesung

Sommersemester 2008

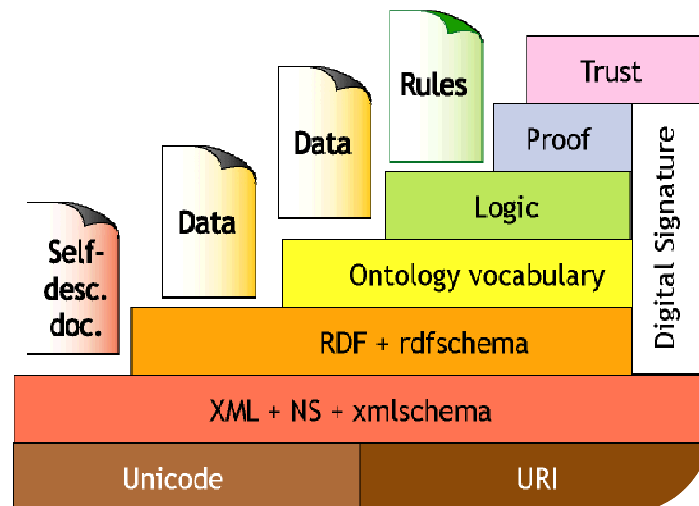
11. Sitzung

Dozent

Nino Simunic M.A.

Computerlinguistik, Campus DU

Übersicht



Ontologiesprache RDF, und RDF Schema

XML

- Universelle Meta-Sprache zur Definition von Markup.
- Einheitliches Framework, Werkzeuge wie Parser, um Daten und Metadaten zwischen Applikationen auszutauschen.
- Aber: XML stellt keine Mittel zur Verfügung, um über die Semantik der Daten zu sprechen.
 - Interpretation der Daten per se nicht eindeutig
 - Verschachtelungen bspw. Können von unterschiedlichen Programmen unterschiedliche verstanden werden.



XML Beispiel

David Billington is a lecturer of Discrete Mathematics.

Verschiedene Möglichkeiten, den Satz in XML auszudrücken:

```
<course name="Discrete Mathematics">
  <lecturer>David Billington</lecturer>
</course>

<lecturer name="David Billington">
  <teaches>Discrete Mathematics</teaches>
</lecturer>

<teachingOffering>
  <lecturer>David Billington</lecturer>
  <course>Discrete Mathematics</course>
</teachingOffering>
```



XML Beispiel

David Billington is a lecturer of Discrete Mathematics.

```
<course name="Discrete Mathematics">
  <lecturer>David Billington</lecturer>
</course>

<lecturer name="David Billington">
  <teaches>Discrete Mathematics</teaches>
</lecturer>
```

Umgedrehte Verschachtelung, jedoch identische Beschreibung der selben Information.

Kein Standard, wie Tag-Schachtelung zu interpretieren ist.



RDF

- RDF im Kern ein OAV Daten-Model
 - *Object-Attribute-Value triple*
 - Object-Attribut-Wert Tripel
 - ⇔ *statement* (Statement, Aussage)
- Abstraktes Datenmodell benötigt für die Darstellung konkrete Syntax
- RDF verwendet i.A. XML-Syntax
 - [0]ther syntactic representations of RDF, not based on XML, are also possible; **XML-based syntax is not a necessary component of the RDF model.**



RDF

- RDF is Domänen-unbabhängig
- Benutzer definiert eigene TErminologie in der als **RDF Schema (RDFS)** genannten Sprache.
- XML *versus* RDF Schema
 - XML Schema constrains the *structure of XML documents*, whereas **RDF Schema defines the vocabulary** used in RDF data models.



RDFS

- In RDFS wird definiert:
 - Vokabular
 - Erlaubte Eigenschaften und Werte der Objekte
 - Beziehungen zwischen Objekten
- Lecturer is a subclass of academic staff member.*
- Bedeutet, dass alle Dozente gleichzeitig zu den akademischen angestellten gehören.
- Wichtig: Bedeutung von “is a subclass of” ist im Gegensatz zu XML fix!
 - It is not up to the application to interpret this term; its **intended meaning** must be respected by all RDF processing software.



Wichtigkeit von RDFS: Beispiel

```
<academicStaffMember>Grigoris Antoniou</academicStaffMember>

<professor>Michael Maher</professor>

<course name="Discrete Mathematics">
  <isTaughtBy>David Billington</isTaughtBy>
</course>
```

- Fragment einer XML-Datei.
- **Xpath** Beispielanfrage `//academicStaffMember`.
- Ergebnis: **Grigoris Antoniou**
- Korrekt vom XML-Standpunkt aus.



Wichtigkeit von RDFS: Beispiel

```
<academicStaffMember>Grigoris Antoniou</academicStaffMember>
<professor>Michael Maher</professor>
<course name="Discrete Mathematics">
  <isTaughtBy>David Billington</isTaughtBy>
</course>
```

Resultate Michael Maher, und David Billington sind als Antworten prinzipiell denkbar, weil:

- Alle Professoren sind akademische Angestellte.
- Veranstaltungen werden nur von akademischen Angestellten durchgeführt.

Derartige Informationen verwenden das **semantische Modell einer spez. Domäne**. Weder in XML, noch in in RDF darstellbar, sondern in **RDFS**.

RDFS macht **semantische Informationen maschinell verfügbar**, in Einklang mit der Vision des Semantic Webs.



Fundamentale Konzepte in RDF: *resources, properties, statements*

- **Resource**: Ding, über das wir reden wollen.
 - Autoren, Bücher, Publisher, Orte, Personen, Hotels, Räume, Suchanfragen, ...
- Jede *resource* besitzt ein **URI**
 - **Universal Resource Identifier**
 - Z.B. **URL** (**U**nified **R**esource **L**ocator, Web Adresse) oder eine andere Form eines **eindeutigen Identifiers**
 - **Identifier erlaubt nicht zwingend Zugriff auf die resource.**
- **URI Schemata existieren auch für** diverse Objekte wie Telefonnummern, ISBN Nr., geographische Orte.

-11-

UNIVERSITÄT
DUISBURG
ESSEN

Fundamentale Konzepte in RDF: *resources, **properties**, statements*

- **Properties** (Eigenschaften): Spezielle Art von *resource*.
 - Beziehungen zwischen Objekten,
 - "geborenIn", "alter", "titel", etc.
- Properties in RDF auch via URIs referenziert (i.A. via URLs).
- URIs zum Verweis auf 'Dinge' wichtig: Globale, weltweite, eindeutiges Schema zur Benennung (von 'Dingen')
 - Reduziert das Homonym-Problem drastisch

-12-

UNIVERSITÄT
DUISBURG
ESSEN

Fundamentale Konzepte in RDF: *resources*, *properties*, **statements**

- Statements beschreiben die *properties* einer *resource*.
- Ein Statement ist ein OAV Tripel:
resource, property, value
- **Werte** sind **entweder** weitere **resource**, oder **Literal**. (Grob: Literale sind atomare Werte (Strings / Zeichenketten))

-13-

UNIVERSITÄT
DUISBURG
ESSEN

Drei Perspektiven auf RDF (1)

Beispiel-Aussage:

David Billington is the owner of the Web page <http://www.cit.gu.edu.au/~db>.

Als OAV Tripel / statement:

```
(  "David Billington",  
  http://www.mydomain.org/site-owner,  
  http://www.cit.gu.edu.au/~db      )
```

- **(*x*, *P*, *y*)** vorstellbar als logische Formel ***P*(*x*, *y*)**
- Das binäre Prädikat ***P*** verbindet Objekt ***x*** mit Objekt ***y***.
- RDF bietet ausschließlich binäre Prädikate (*properties*).
- Anmerkung: *Property* **'site-owner'**, und eines der zwei Objekt werde via URLs identifiziert. Das andere wird schlicht mit einer Zeichenkette identifiziert.

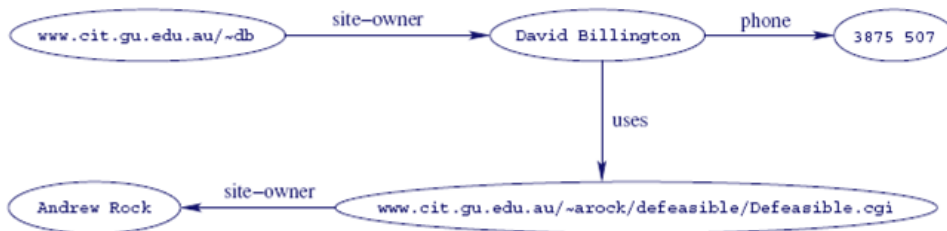
-14-

UNIVERSITÄT
DUISBURG
ESSEN

Drei Perspektiven auf RDF (2)



Figure 3.1 Graph representation of triple



-15-

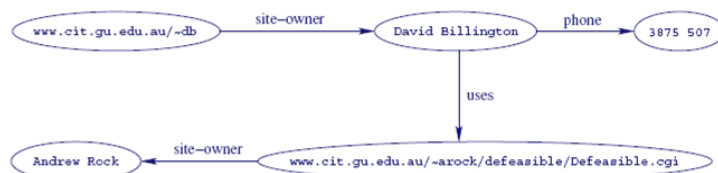
UNIVERSITÄT
DUISBURG
ESSEN

Drei Perspektiven auf RDF (2)

Die zweite Perspektive ist Graph-basiert:



Figure 3.1 Graph representation of triple

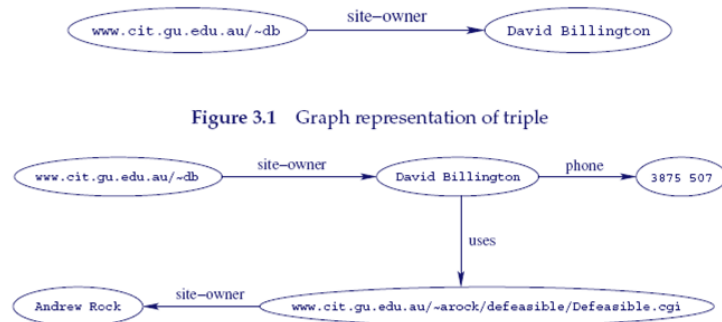


- Gerichteter Graph
- Benannte Knoten und Kanten
- Richtung der Kanten: Von der resource (Subjekt der Aussage) zum Wert (Objekt der Aussage).

-16-

UNIVERSITÄT
DUISBURG
ESSEN

Drei Perspektiven auf RDF (2)



- Graphen sinnvoll fürs intellektuelle Interpretation.
- Semantic Web fordert maschinell verwertbare Repräsentationen:
→ Dritte Repräsentation möglich: Auf Basis von XML ...

-17-

UNIVERSITÄT
DUISBURG
ESSEN

Drei Perspektiven auf RDF (3): XML Basis

- Ein RDF Dokument wird mittels XML Element mit Tag **rdf:RDF** beschrieben.
- Inhalt dieses Elements ist eine Menge von **descriptions**, die das **rdf:Description** Tag verwenden.
- Jede description macht eine Aussage über die *resource*. Kann auf drei Arten identifiziert werden:
 - **about Attribut**, referenziert bestehende *resource*
 - **ID Attribut**, erstellt neue *resource*
 - ohne Bezeichnung, erstellt anonyme *resource*

-18-

UNIVERSITÄT
DUISBURG
ESSEN

Erstes Statement:

Erste Zeile spezifiziert, dass wir XML verwenden. **Muss** stets angegeben werden.

```
<?xml version="1.0" encoding="UTF-16"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mydomain="http://www.mydomain.org/my-rdf-ns">
```

Das `rdf:Description` Element macht eine Aussage über die *resource* `http://www.cit.gu.edu.au/~db`:

```
<rdf:Description rdf:about="http://www.cit.gu.edu.au/~db">
  <mydomain:site-owner>
    David Billington
  </mydomain:site-owner>
</rdf:Description>
</rdf:RDF>
```

Innerhalb der *Beschreibung*:

- Verwendung der *property* als Tag
- *Inhalt* ist der *Wert* der *property*



Serialisierung

- Beschreibungen bei XML-Umsetzung in bestimmter Reihenfolge:
 - **XML Syntax imposes a *serialization*.**
- *Reihenfolge bei abstraktem RDF Model jedoch irrelevant.*
 - Graphen-basiertes Modell ist das 'echte' Datenmodell von RDF.
 - XML is "nur" eine mögliche, serielle Repräsentation.



Reifikation (engl. *reification*) ~ *Vergegenständlichung*

- In RDF sind Aussagen über Aussagen darstellbar:

`Grigoris believes that David Billington is the creator of the Web page http://www.cit.gu.edu.au/~db.`

- Beschreibung von Glauben/Vetruen in andere Aussagen
- Lösung: Eindeutiger Identifizierer für jede Aussage zur Referenzierung der Aussage.

-21-

UNIVERSITÄT
DUISBURG
ESSEN

Reifikationsmechanismus in RDF. Kern-Idee:

- Einführen eines Objekts, z.B. **belief1**
- **belief1** mit allen (drei) Teilen via *Subjekt, Prädikat, Objekt* verbinden

`Grigoris believes that David Billington is the creator of the Web page http://www.cit.gu.edu.au/~db.`

- Subjekt von belief1 ist David Billington,
- Prädikat ist der Erschaffer
- Objekt ist <http://www.cit.gu.edu.au/~db>.
 - Leider umständig, da nur Tripel in RDF möglich.
 - Identifier kann nicht direkt and Tripel gehängt werden (wäre dann ein Quadrupel)

-22-

UNIVERSITÄT
DUISBURG
ESSEN

Datentypen

(Telefonnummer) "3875507"

- RDF-lesendes Programm kann so (^) nicht wissen, ob es als Literal oder Zahl interpretiert werden soll.
- Wie bei viele anderen Programmen: Explizite Angabe des Datentyps nötig (String / Integer / ...)
 - So auch in RDF
- Getypte Literale (*typed literals*) stellen diese explizite Information (Datentyp) bereit.



Datentypen in RDF

- Getyptes Literal, um bspw. die "27" als Ganzzahl 27 interpretieren zu lassen:

```
( "David Billington",  
  http://www.mydomain.org/age,  
  "27"^^http://www.w3.org/2001/XMLSchema#integer )
```
- Zweierlei:
 - ^^ -Notation, um den Typ des Literals anzugeben.
 - Verwendung eines **vordefinierten XML Schema Datentyps**.
- Prinzipiell auch benutzerdefinierte, externe Datentypen ladbar. XML Schema ist jedoch i.A. *völlig* ausreichend.
 - Boolean, Integer and Float, Zeit, Datum, ...



Kritik an RDF

- Binärität der *properties*.
- >2 Argumente kann jedoch mittels mehrere binärer Prädikate wie im Bsp. simuliert werden.

Prädikat *referee* mit drei Argumenten:

referee(X, Y, Z)

Bedeutung, intuitiv:

X ist ein Schiedsrichter in einem Schachspiel zwischen Spielern *Y* und *Z*.

-25-

UNIVERSITÄT
DUISBURG
ESSEN

Hilfs-Resource *chessGame*

- Einführen der neuen Ressource *chessGame*, und den binären Prädikaten *ref*, *player1*, *player2*
- Repräsentation von *referee(X, Y, Z)* wie folgt:



Figure 3.3 Representation of a tertiary predicate

-26-

UNIVERSITÄT
DUISBURG
ESSEN

Kritik an RDF

- Lösung okay, jedoch wären dreistellige Prädikate natürlicher und einfacher.
- Problem: *Properties* sind spezielle Ausprägungen von Ressourcen.
 - Folglich: *properties* per se könn(-t-)en als Objekte in einem OAV Tripel (*statement*) verwendet werden. Eher ungewöhnlich für Modellierungssprachen.
- Reifikation: Mächtiger Mechanismus, jedoch für die Basis-Ebene des Semantic Webs zu komplex.
- XML-basierte Syntax ist gut geeignet für maschinelle Verarbeitung, jedoch eher mühsam zu "lesen" für den Menschen.

-27-

UNIVERSITÄT
DUISBURG
ESSEN

XML-basierte Syntax von RDF

- Ein RDF Dokument besteht aus einem **rdf:RDF** Element, dessen **Inhalt** eine Menge von Beschreibungen ist.

```
<!DOCTYPE rdf [  
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">  
>  
<rdf:RDF  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"   
  xmlns:xsd=http://www.w3.org/2001/XMLSchema#  
  xmlns:uni="http://www.mydomain.org/uni-ns#" >  
  
    ...Beschreibungen ...  
  
</rdf:RDF>
```

-28-

UNIVERSITÄT
DUISBURG
ESSEN

Beispiel

```
<!DOCTYPE rdf [  
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">  
<rdf:RDF  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"   
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"   
  xmlns:uni="http://www.mydomain.org/uni-ns# ">  
  
  <rdf:Description rdf:about="123456">  
    <uni:name>Martin Mumpitz</uni:name>  
    <uni:title>Professor</uni:title>  
    <uni:age rdf:datatype="xsd:integer">27</uni:age>  
  </rdf:Description>  
  
  <rdf:Description rdf:about="kurs987">  
    <uni:courseName>Wissensrepräsentation</uni:courseName>  
    <uni:isTaughtBy>Martin Mumpitz</uni:isTaughtBy>  
  </rdf:Description>  
  
</rdf:RDF>
```



In XML: Namespaces nur zur **Disambiguierung**

In RDF: Externe Namespaces sind **resources** definierende Dokumente

Können ins RDF-Dokument importiert werden: Bspw. zur Wiederverwendung

```
<!DOCTYPE rdf [  
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">  
<rdf:RDF  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"   
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"   
  xmlns:uni="http://www.mydomain.org/uni-ns# ">  
  
  <rdf:Description rdf:about="123456 ">  
    <uni:name>Martin Mumpitz</uni:name>  
    <uni:title>Professor</uni:title>  
    <uni:age rdf:datatype="xsd:integer">27</uni:age>  
  </rdf:Description>  
  
  <rdf:Description rdf:about="kurs987">  
    <uni:courseName>Wissensrepräsentation</uni:courseName>  
    <uni:isTaughtBy>Martin Mumpitz</uni:isTaughtBy>  
  </rdf:Description>  
  
</rdf:RDF>
```



Property elements / values

```
<rdf:Description rdf:about="kurs987">
  <uni:courseName>Wissensrepräsentation</uni:courseName>
  <uni:isTaughtBy>Martin Mumpitz</uni:isTaughtBy>
</rdf:Description>
```

Inhalte von `rdf:Description` Elementen werden als **property elements** bezeichnet.

Die zwei Elemente `uni:courseName` und `uni:isTaughtBy` definieren beide **property-values** für `kurs987`.
Entspricht also zwei *statements*.

-31-

UNIVERSITÄT
DUISBURG
ESSEN

rdf:datatype

```
<rdf:Description rdf:about="123456">
  <uni:name>Martin Mumpitz</uni:name>
  <uni:title>Professor</uni:atitle>
  <uni:age rdf:datatype="&xsd;integer">27</uni:age>
</rdf:Description>
```

Attribut `rdf:datatype="&xsd;integer"` zur Angabe des Datentyps des Werts der *age* property.

- Selbst wenn diese Eigenschaft (in RDFS) Range als `"&xsd;integer"` definiert hat, ist die Angabe des Typs jedesmal nötig, wenn die *property* so oder ähnlich verwendet wird.
- Grund: Garantie, dass ein RDF Prozessor den korrekten Typ zuweisen kann, selbst wenn er das RDF Schema noch nicht gesehen hat. Durchaus sehr wahrscheinliche Sachlage im WWW.

-32-

UNIVERSITÄT
DUISBURG
ESSEN

Das `rdf:resource` Attribut

Vorangehendes Beispiel suboptimal mit Bezug auf formaler Darstellung: Beziehung zwischen Kurs und Dozent war nur implizit durch die Namensgebung:

```
<rdf:Description rdf:about="123456 ">
  <uni:name>Martin Mumpitz</uni:name>
  <uni:title>Professor</uni:title>
</rdf:Description>

<rdf:Description rdf:about="kurs987">
  <uni:courseName>Wissensrepräsentation</uni:courseName>
  <uni:isTaughtBy>Martin Mumpitz</uni:isTaughtBy>
</rdf:Description>
```

Für die Maschine ist identischer Name "Zufall" :
Prof. **Martin Mumpitz** und unterrichtender **Martin Mumpitz**
könnten unterschiedliche Personen sein.

Benötigt: Formale Spezifikation des Faktums, dass (z.B.) Dozent
von **kurs987** Angestellter mit Nummer **123456** ist, dessen Name
Martin Mumpitz lautet.



Das `rdf:resource` Attribut

Umsetzung des Effekts via `rdf:resource` Attribut:

```
<rdf:Description rdf:about= "123456">
  <uni:name>Martin Mumpitz</uni:name>
  <uni:title>Professor</uni:title>
</rdf:Description>

<rdf:Description rdf:about="kurs987">
  <uni:courseName>Wissensrepräsentation</uni:courseName>
  <uni:isTaughtBy rdf:resource="123456"/>
</rdf:Description>
```



Das **rdf:resource** Attribut und die Raute

Wird bspw. die Nummer 123456
via ID-Attribut zugewiesen (eben: **about**):

Symbol vor 123456 im Wert der **rdf:resource**

```
<rdf:Description rdf:about="kurs987">
  <uni:courseName>Wissensrepräsentation</uni:courseName>
  <uni:isTaughtBy rdf:resource="#123456"/>
</rdf:Description>

<rdf:Description rdf:ID="#123456">
  <uni:name>Martin Mumpitz</uni:name>
  <uni:title>Professor</uni:title>
</rdf:Description>
```

Das **rdf:resource** Attribut und die Raute

- Analog gilt für extern definierte Ressourcen:
<http://www.someDomain.org/xyz-ns/#vUSA1234>
... als Beispiel-Wert für **rdf:about**
- <http://www.someDomain.org/xyz-ns/#vUSA1234>
ist die URI, an dem **vUSA1234** definiert wurde und
zu finden ist.

Verschachtelte Beschreibungen (*engl. nested descriptions*)

```
<rdf:Description rdf:about="kurs987">
```

```
  <uni:courseName>Wissensrepräsentation</uni:courseName>
```

```
  <uni:isTaughtBy>
```

Beschreibungen können innerhalb von Beschreibungen definiert werden:

```
    <rdf:Description rdf:about= "123456">
```

```
      <uni:name>Martin Mumpitz</uni:name>
```

```
      <uni:title>Professor</uni:title>
```

```
    </rdf:Description>
```

```
  </uni:isTaughtBy>
```

```
</rdf:Description>
```

kurs997 kann noch immer auf die neue Ressource **123456** verweisen:
Gültigkeitsbereich / Sichtbarkeit (scope) ist global (,und nicht auf die Schachtel in der Verschachtelung beschränkt)!



Das **rdf:type** Element

- Bisher nur zwei Kategorien: Kurse, Dozent.
 - Für uns klar, jedoch bisher nicht formal deklariert.
 - Somit keine maschinelle Interpretation dieses Sachverhalts.
- Umsetzung in RDF: **rdf:type** beim Erzeugen eines Erstellen Statements!

```
<rdf:type rdf:resource="&uni;lecturer"/>
```

```
<rdf:type rdf:resource="&uni;course"/>
```



Das `rdf:type` Element

```
<rdf:Description rdf:about="kurs987">
  <rdf:type rdf:resource="&uni;course"/>
  <uni:courseName>Wissensrepräsentation</uni:courseName>
  <uni:isTaughtBy rdf:resource="123456"/>
</rdf:Description>

<rdf:Description rdf:about="123456">
  <rdf:type rdf:resource="&uni;lecturer"/>
  <uni:name>Martin Mumpitz</uni:name>
  <uni:title>Professor</uni:title>
</rdf:Description>
```

`rdf:type` gestattet das Einführen einer gewissen Struktur in RDF Dokumente. Mehr Strukturierung via RDF Schema.

Abgekürzte Syntax

Syntax kann verkürzt werden, wenn:

1. **Childless property elements** within description elements may be replaced by XML attributes, as in XML.
2. For **description elements with a typing element** we can **use the name specified in the `rdf:type` element instead of `rdf:Description`**.

Beispiel

```
<rdf:Description rdf:ID="kurs987">
  <rdf:type rdf:resource="&uni;course"/>
  <uni:courseName>Wissensrepräsentation</uni:courseName>
  <uni:isTaughtBy rdf:resource="#123456"/>
</rdf:Description>
```

1. **Childless property elements** within description elements may be replaced by XML attributes, as in XML.

```
<rdf:Description rdf:ID="kurs987" uni:courseName="Wissensrepräsentation">
  <rdf:type rdf:resource="&uni;course"/>
  <uni:isTaughtBy rdf:resource="#123456"/>
</rdf:Description>
```

-41-

UNIVERSITÄT
DUISBURG
ESSEN

Beispiel

```
<rdf:Description rdf:ID="kurs987" uni:courseName="Wissensrepräsentation">
  <rdf:type rdf:resource="&uni;course"/>
  <uni:isTaughtBy rdf:resource="#123456"/>
</rdf:Description>
```

2. For **description elements with a typing element** we can **use the name specified in the `rdf:type` element instead of `rdf:Description`**.

```
<uni:course rdf:ID="kurs987" uni:courseName="Wissensrepräsentation">
  <uni:isTaughtBy rdf:resource="#123456"/>
</uni:course>
```

**Äquivalente
Repräsentationen!**

-42-

UNIVERSITÄT
DUISBURG
ESSEN

Container Elemente in RDF

- Container Element zur Sammlung einer Menge von Ressourcen oder Attributen
 - Über die als *Ganzes* eine Aussage gemacht werden soll.
 - Z.B. Aussage über alle Veranstaltungen eines bestimmten Dozenten.
 - **rdf:Bag**, **rdf:Seq**, **rdf:Alt**

-43-

UNIVERSITÄT
DUISBURG
ESSEN

Drei Typen von Container Elementen in RDF

- **rdf:Bag**
 - Unsortierter Container, Reihenfolge bleibt nicht erhalten.
- **rdf:Seq**
 - Sortierter Container – Ordnung bleibt dem Einfügen angepasst.
 - Beispiele: Agenda-Einträge, alphabetisierte Liste der Angestellten, Wörter eines Satzes, ...
- **rdf:Alt**
 - Container für eine Menge von Alternativen.
 - Beispiele: Übersetzungen zu einem Dokument in verschiedene Sprachen, Datei-Mirrors, mehrere Dozenten für eine Veranstaltung ...

-44-

UNIVERSITÄT
DUISBURG
ESSEN

rdf:Alt Beispiel

Angenommen, der Kurs CIT1111 wird entweder von von Grigoris Antoniou oder David Billington gehalten:

```
<uni:course rdf:about="CIT1111"
  uni:courseName="Discrete Mathematics">
  <uni:lecturer>
    <rdf:Alt>
      <rdf:li rdf:resource="949352"/>
      <rdf:li rdf:resource="949318"/>
    </rdf:Alt>
  </uni:lecturer>
</uni:course>
```

-45-

UNIVERSITÄT
DUISBURG
ESSEN

rdf:Bag Beispiel

Die Container Elemente haben optionale ID Attribute, über die der Container referenziert werden kann.

```
<uni:lecturer rdf:about="949318"
  uni:name="David Billington"
  uni:title="Associate Professor">
  <uni:coursesTaught>
    <rdf:Bag rdf:ID="DBcourses">
      <rdf:_1 rdf:resource="CIT1111"/>
      <rdf:_2 rdf:resource="CIT3112"/>
    </rdf:Bag>
  </uni:coursesTaught>
</uni:lecturer>
```

-46-

UNIVERSITÄT
DUISBURG
ESSEN

Reifikation

- Erinnerung: Statements über Statements.
- Bedingung: Statement muss über Identifier referenzierbar sein.
- RDF erlaubt diese Referenzierung via Reifikationsmechanismus
 - Statement wird zu Ressource.

```
<rdf:Description rdf:about="949352">  
  <uni:name>Grigoris Antoniou</uni:name>  
</rdf:Description>
```

Reifikation

???

-47-

UNIVERSITÄT
DUISBURG
ESSEN

Reifizieren in RDF

```
<rdf:Description rdf:about="949352">  
  <uni:name>Grigoris Antoniou</uni:name>  
</rdf:Description>
```

Reifikation

rdf:subject, rdf:predicate,
rdf:object, um Teile des Statements gezielt
anzusprechen.

ID des *statement* kann wie bei allen anderen
Beschreibungen (description) zur eindeutigen
Referenzierung verwendet werden.

```
<rdf:Statement rdf:about="StatementAbout949352">  
  <rdf:subject rdf:resource="949352"/>  
  <rdf:predicate rdf:resource="&uni;name"/>  
  <rdf:object>Grigoris Antoniou</rdf:object>  
</rdf:Statement>
```

-48-

UNIVERSITÄT
DUISBURG
ESSEN

RDF versus RDFS

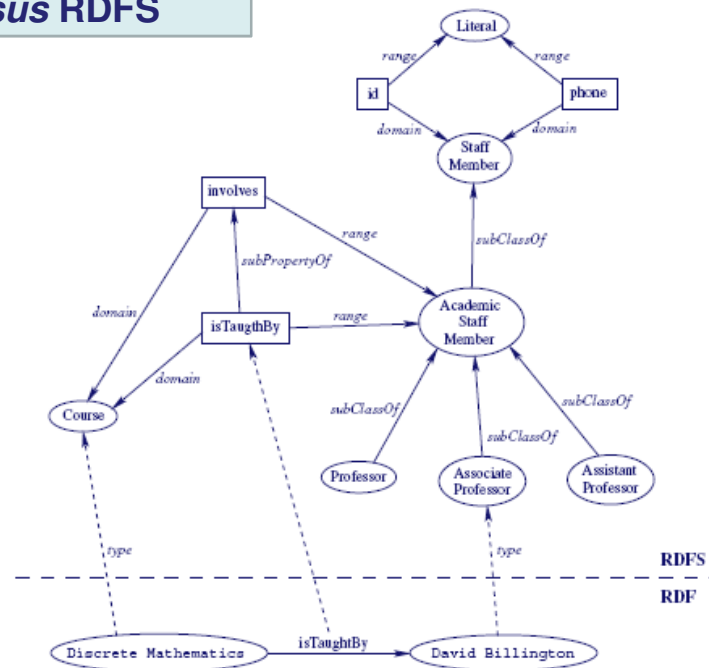


Figure 3.6 RDF and RDFS layers

-49-

UNIVERSITÄT
DUISBURG
ESSEN

RDF Schema: Grundlegendes

- RDF is eine universelle Sprache zur Beschreibung von Ressourcen mittels eigenem Vokabular.
- RDF macht keine Annahmen über spezifische Anwendungen oder Applikationen
- RDF definiert für keine Domäne Semantik!
- Ausschließlich Aufgabe des Benutzers, dieses in **RDF Schema (RDFS)** zu beschreiben.

-50-

UNIVERSITÄT
DUISBURG
ESSEN

Klassen und Eigenschaften (*properties*)

- How do we describe a particular domain? Let us consider the domain of courses and lecturers at Griffith University.
- First we have to specify the “things” we want to talk about. Here we make a first, fundamental distinction.
- On one hand, we want to talk about particular lecturers, such as David Billington, and particular courses, such as Discrete Mathematics; we have already done so in RDF. But we also want to talk about courses, first-year courses, lecturers, professors, and so on. What is the difference?
- In the first case we talk about *individual objects (resources)*, in the second we talk about *classes that define types of objects*.

-51-

UNIVERSITÄT
DUISBURG
ESSEN

Classes and Properties

- A **class** can be thought of as a **set of elements**.
 - **Individual objects** that **belong** to a **class** are referred to as ***instances*** of that class.
- *We have already* defined the relationship between instances and classes in RDF using **`rdf:type`**.
- An important use of classes is to impose restrictions on what can be stated in an RDF document using the schema.

-52-

UNIVERSITÄT
DUISBURG
ESSEN

Classes and Properties

- In programming languages, *typing* is used to prevent nonsense from being written (such as $A+1$, where A is an array; we lay down that the arguments of $+$ must be numbers).
- The same is needed in RDF. After all, we would like to disallow statements such as

Discrete Mathematics is taught by Concrete Mathematics.
Room MZH5760 is taught by David Billington.

-53-

UNIVERSITÄT
DUISBURG
ESSEN

Classes and Properties

- The first statement is nonsensical because we want courses to be taught by lecturers only. This imposes a restriction on the values of the property “is taught by”. In mathematical terms, we restrict the ***range of the property***.

Discrete Mathematics is taught by Concrete Mathematics.
Room MZH5760 is taught by David Billington.

- The second statement is nonsensical because only courses can be taught. This imposes a restriction on the objects to which the property can be applied. In mathematical terms, we restrict the ***domain of the property***.

-54-

UNIVERSITÄT
DUISBURG
ESSEN

Class Hierarchies and Inheritance

- Once we have classes we would also like to establish relationships between them.
- For example, suppose that we have classes for
 - staff members
 - assistant professors
 - academic staff members
 - administrative staff members
 - professors technical support staff members
 - associate professors

-55-

UNIVERSITÄT
DUISBURG
ESSEN

Klassenhierarchie und Vererbung

- These classes are not unrelated to each other. For example, every professor is an academic staff member. We say that “professor” is a *subclass* of “academic staff member”, or equivalently, that “academic staff member” is a *superclass* of “professor”.
- The subclass relationship defines a hierarchy of classes, as shown in figure 3.5.

-56-

UNIVERSITÄT
DUISBURG
ESSEN

Klassenhierarchie, Vererbung

Keine Einschränkung in RDFS, dass Klassen eine strikte Taxonomie einhalten.



Eine Klasse kann mehrere Superklassen haben.

Wenn eine Klasse A eine Subklasse von sowohl B1 als auch B2 ist →

Jede Instanz von A ist sowohl Instanz von B1 als auch von B2.

-57-

UNIVERSITÄT
DUISBURG
ESSEN

properties Hierarchien

- Analog zu Klassen: Auch Relationen in einer Hierarchie. Z.B.
- "is taught by" ist eine **subproperty** von "involves":
 - Wenn Kurs **c** von einem akademischen Angestellten **a** gelehrt wird, dann involviert **c** **a**.
 - Das Gegenteil ist nicht zwingend der Fall.
 - Z.B. sei **a** ein Tutor, der Hausaufgaben bewertet, aber nicht **c** lehrt

Generell gilt:

P ist eine **subproperty** von **Q**,
immer dann wenn für **Q(x, y)** auch **P(x, y)** gilt.

-58-

UNIVERSITÄT
DUISBURG
ESSEN

RDF versus RDFS

- Letzter Punkt zur Unterscheidung RDF/RDFS:
- Gegeben sei das RDF Statement

Discrete Mathematics is taught by David Billington.

- Das Schema für dieses Statement enthalte Klassen wie Dozenten, Angestellte, ..., aber auch Eigenschaften wie gelehrt von, involviert, ...

-59-

UNIVERSITÄT
DUISBURG
ESSEN

- **Blöcke:** Properties
- **Ellipsen oberhalb** gestr. Linie: **Klassen**
- **Ellipsen unterhalb** gestr. Linie: **Instanzen**

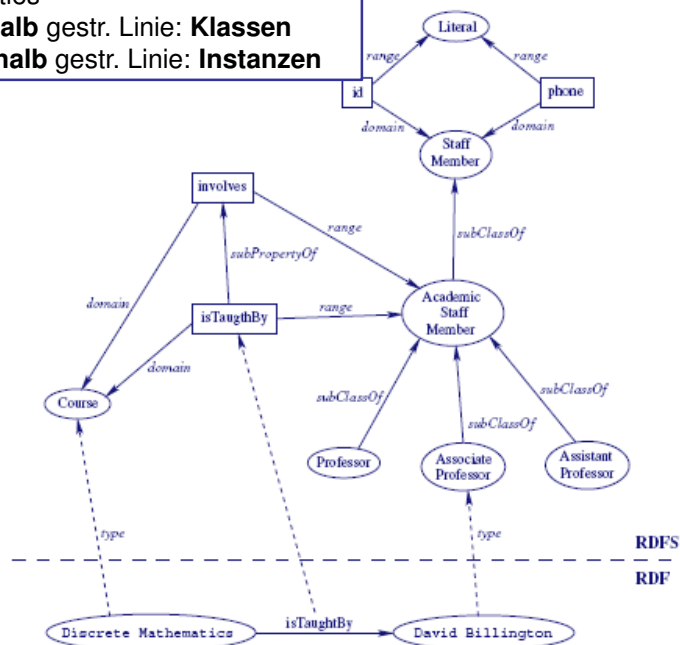


Figure 3.6 RDF and RDFS layers

-60-

UNIVERSITÄT
DUISBURG
ESSEN

▪ **RDF Schema: Modellierung der Semantik** einer Domäne.

▪ **Überraschung: RDFS wird mit RDF Mitteln umgesetzt.**

▪ **Modellerieungsprimitive** von **RDF Schema** werden definiert via **Ressourcen** und **Properties**.

▪ Ausserdem: Graph zeigt eine Klassen/Property-Hierarchie plus Instanzen.

▪ An sich aber "nur" ein gelabelter Graph, welcher in RDF kodiert werden kann.

▪ Denn: RDF erlaubt beliebige Statements zu jeder Ressource. Plus: Alles, was eine URI besitzt, kann eine Ressource sein.

▪ Beispiel: Subklassen-Relation mit RDF-Mitteln.

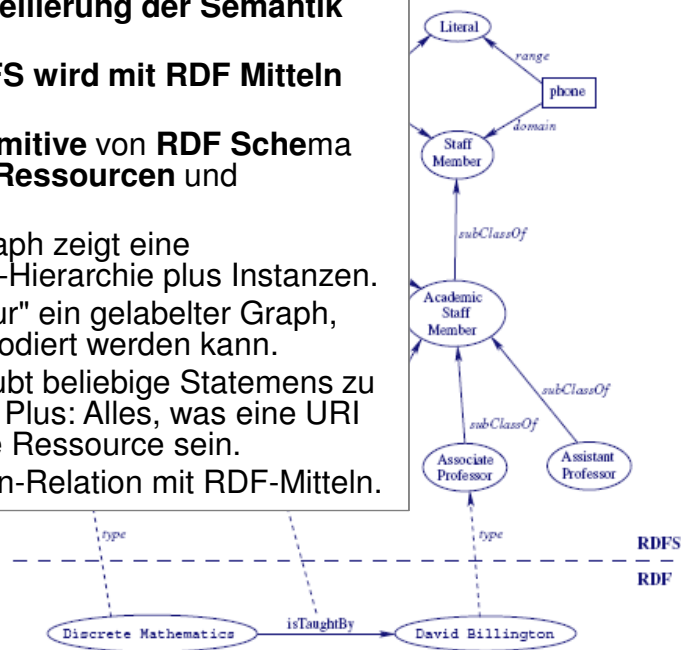


Figure 3.6 RDF and RDFS layers

-61-

UNIVERSITÄT
DUISBURG
ESSEN

Die Sprache »RDF Schema«: Klasse

- Semantik, dass die Klasse "lecturer" Subklasse von "academic staff member" ist:
 1. Definiere *resource* **lecturer**, **academicStaffMember**, **subClassOf**
 2. Definiere **subClassOf** als *property*
 3. Schreibe Tripel
(**subClassOf**, **lecturer**, **academicStaffMember**)
- Alles im Rahmen von RDF → RDFS Dokument ist "nur" ein RDF Dokument, und wir verwenden die XML-basierte Syntax von RDF.
- Zu den Modellierungsprimitiven von RDF Schema ...

-62-

UNIVERSITÄT
DUISBURG
ESSEN

RDFS Core Classes

`rdfs:Resource`, die Klasse aller *resources*.

`rdfs:Class`, die Klasse aller Klassen.

`rdfs:Literal`, die Klasse aller Literale (String). Momentan nur
Literele als "Datentypen" in RDF/RDFS.

`rdf:Property`, die Klasse aller *properties*.

`rdf:Statement`, die Klasse aller reifizierten *statements*.

Z.B., eine Klasse `lecturer` kann wie folgt definiert werden:

```
<rdfs:Class rdf:ID="lecturer">  
  . . .  
</rdfs:Class>
```



Wichtige Properties für die Definition von Beziehungen (1)

- **`rdf:type`**, which relates a resource to its class (see section 3.3.3).
 - The resource is declared to be an instance of that class.
- **`rdfs:subClassOf`**, which relates a class to one of its superclasses;
 - all instances of a class are instances of its superclass. Note that a class may be a subclass of more than one class. As an example, the class `femaleProfessor` may be a subclass of both `female` and `professor`.
- **`rdfs:subPropertyOf`**, which relates a property to one of its superproperties.



Wichtige Properties für die Definition von Beziehungen (2)

- Alle Dozenten sind *staff members*:

```
<rdfs:Class rdf:about="lecturer">  
  <rdfs:subClassOf  
    rdf:resource="staffMember"/>  
</rdfs:Class>
```

- `rdfs:subClassOf` , `rdfs:subPropertyOf` sind transitiv
- `rdfs:Class` is Subklasse von `rdfs:Resource` (every class is a resource), und `rdfs:Resource` ist eine Instance von `rdfs:Class` (`rdfs:Resource` is the class of all resources, so it is a class!). For the same reason, every class is an instance of `rdfs:Class`



Wichtige Properties für die Einschränkung von Beziehungen

- `rdfs:domain`

specifies the domain of a property *P*, that is, the *class* of those resources that may appear as subjects in a triple with predicate *P*. If the domain is not specified, then any resource can be the subject.

- `rdfs:range`

specifies the range of a property *P*, that is, the *class* of those resources that may appear as values in a triple with predicate *P*.



Wichtige Properties für die Einschränkung von Beziehungen

Beispiel: *phone* gilt nur für *staff members*, und der Wert ist immer ein Literal.

```
<rdf:Property rdf:ID="phone">
  <rdfs:domain rdf:resource="#staffMember"/>
  <rdfs:range rdf:resource="&rdf;Literal"/>
</rdf:Property>
```

In RDF Schema zudem:

`rdfs:ConstraintResource`,
Klasse aller *constraints*

`rdfs:ConstraintProperty`,
enthält alle *properties*, die *constraints* enthalten.

Zwei Instanzen in der Sprache: **`rdfs:domain`**, und
`rdfs:range`

Definiert als Subklasse von **`rdfs:ConstraintResource`**
und **`rdf:Property`**

-67-

UNIVERSITÄT
DUISBURG
ESSEN

Beziehung zwischen den Primitiven in RDFS

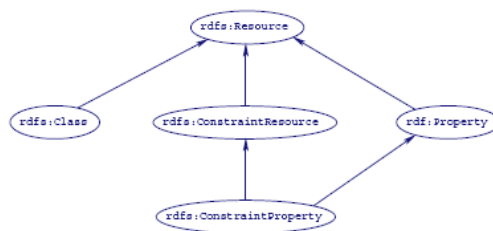


Figure 3.7 Subclass hierarchy of some modeling primitives of RDFS

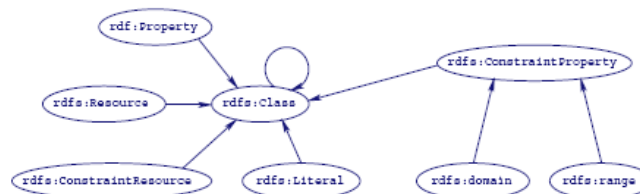


Figure 3.8 Instance relationships of some modeling primitives of RDFS

-68-

UNIVERSITÄT
DUISBURG
ESSEN

Nützliche Properties fürs Reifizieren

- **rdf:subject**
 - Bezieht sich auf das Subjekt eines reifizierten Statements
- **rdf:predicate**
 - Bezieht sich auf das Prädikat seines r. S.
- **rdf:object**
 - Bezug zum Objekt seines r. S., which relates a reified statement to its object

-69-

UNIVERSITÄT
DUISBURG
ESSEN

Container Klassen

- **rdf:Bag**
 - Klasse der *bags*
- **rdf:Seq**
 - Klasse der *sequences*
- **rdf:Alt**
 - Klasse der *alternatives*.
- **rdfs:Container**
 - Superklasse aller *container classes*, inklusive der Klassen oberhalb (Folie)

-70-

UNIVERSITÄT
DUISBURG
ESSEN

Hilfs-properties

- **rdfs:seeAlso**
 - Resource wird mit anderer Resource vernetzt, welche die erste Resource erklärt.
- **rdfs:isDefinedBy**
 - Subproperty von rdfs:seeAlso
 - Zeigt auf Ort, bei dem i.A. ein RDF schema liegt.
- **rdfs:comment.**
- **rdfs:label**
 - Menschen-freundliche Bezeichnung für eine Ressource
 - U.a. zur Anzeige von sinnvollen Knoten/Kanten-Bezeichnern bei Visualisierung, ...

-71-

UNIVERSITÄT
DUISBURG
ESSEN

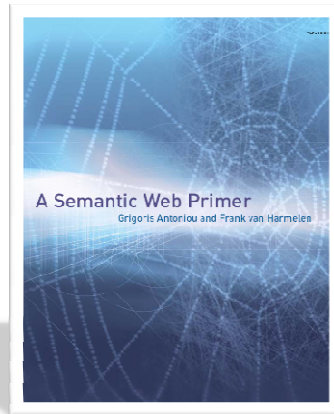
Zusammenfassung

- RDF provides a foundation for representing and processing metadata.
- RDF has a graph-based data model. Its key concepts are resource, property, and statement. A statement is a resource-property-value triple.
- RDF has an XML-based syntax to support syntactic interoperability. XML and RDF complement each other because RDF supports semantic interoperability.
- RDF has a decentralized philosophy and allows incremental building of knowledge, and its sharing and reuse.
- RDF is domain-independent. RDF Schema provides a mechanism for describing specific domains.
- RDF Schema is a primitive ontology language. It offers certain modelling primitives with fixed meaning. Key concepts of RDF Schema are class, subclass relations, property, subproperty relations, and domain and range restrictions.
- There exist query languages for RDF and RDFS. Some points that will be discussed in the next chapter:
- RDF Schema is quite primitive as a modelling language for the Web. Many desirable modelling primitives are missing.
- Therefore we need an ontology layer on top of RDF/RDFS.

-72-

UNIVERSITÄT
DUISBURG
ESSEN

Referenzen



Grigoris Antoniou, Frank van Harmelen, A. **2004**. *Semantic Web Primer*, The MIT Press, Cambridge, Massachusetts, April.

Kapitel 3