

Sprachen und Tools des Semantic Web

RDF und RDF/XML (Fortsetzung)

Zusammenfassung und ergänzende Bemerkungen (1)

- XML Dokumente
 - Prolog: XML Deklaration
 - XML Elemente
 - Werden mittels „<tag> *content* </tag>“ bzw. <tag/> (kein Content) spezifiziert
 - *Content* kann weitere Elemente beinhalten oder einfache Datentypen (Strings)
 - Attribute eines Elements mittels „<tag att₁=val₁ ... att_n=val_n> *content* </tag> bzw. „< tag att₁=val₁ ... att_n=val_n/>“ spezifiziert
 - Bestehen aus **Attributname** und **Attributwert**
 - Wohlgeformte und gültige (valid) Dokumente
 - DTDs und XML Schema sind zwei Ansätze, weitere Einschränkungen des XML Dokuments zu definieren
 - XML Applikation

Zusammenfassung und ergänzende Bemerkungen (2)

● URIs und Namespaces

- URIs sind eindeutige Identifikatoren für Ressourcen
 - Ressourcen können auch XML Dokumente bzw. Fragmente daraus sein
- Sowohl Tag, Attributnamen und Attributwerte basieren auf URIs
 - Dadurch weltweit eindeutige Unterscheidung (zumindest prinzipiell) möglich
 - Durch „<... xmlns:np=URIRef ...>“ kann ein Namespace Präfix zu weiteren Verwendung innerhalb eines XML Elements definiert werden
 - Tags bzw. Attributnamen erhalten durch „<np:tag ... np:att=val ...>“den entsprechenden Präfix (**QName**)

NICHT
Attributwerte



Zusammenfassung und ergänzende Bemerkungen (3)

• URIs und Namespaces (Forts.)

- Durch „<... xmlns=*URIRef* ...>“ wird ein Default-Namespace definiert
 - Alle Tags und Attribute ohne expliziten Präfix sind innerhalb des entsprechenden Elements mit dem Default-Namespace „gepräfixed“
- Attributwerte können **nicht** mit Namespace Präfix versehen werden
 - In „<... *np:att=URIRef* ...>“ muss *URIRef* entweder absolut („http://www.../x/y/z“) spezifiziert werden oder wird relativ zu aktuellen „*xml:base*“ interpretiert
 - Alternative (**neu**) Verwendung von XML Entities:
<!DOCTYPE *Root-Name* [
 <!ENTITY *entityName* "http://www.../x/z#" >
]>
 - Spezifikation einer *URIRef* als Wert eines Attributs:
"&*entityName*;*appendString*"

Prolog

Zusammenfassung und ergänzende Bemerkungen (4)

• URIs und Namespaces (Forts.)

• Beispiel

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY xsd http://www.w3.org/2001/XMLSchema#>
]>
<rdf:RDF
  xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:xsd=http://www.w3.org/2001/XMLSchema#
  xmlns=http://www.iis.uni-hildesheim.de/rdf#>
  <rdf:Description rdf:about=
    "http://www.iis.uni-hildesheim.de/STSW">
    <dozent rdf:datatype="&xsd:String">
      Martin Schaaf
    </dozent>
  </rdf:Description>
</rdf:RDF>
```

I.d.R. wie
Namespacepräfix

Zusammenfassung und ergänzende Bemerkungen (5)

- RDF und RDF/XML
 - Allgemeines Modell zur Beschreibung/Charakterisierung von Ressourcen
 - RDF Statements sind Tripel: (Subjekt, Prädikat, Objekt)
 - RDF/XML: XML-basiertes Serialisierungsformat für RDF
 - RDF Dokumente immer eingeleitet durch `<rdf:RDF>` Element
 - Inkl. Namespace Deklarationen

Zusammenfassung und ergänzende Bemerkungen (6)

- RDF und RDF/XML
 - Statements werden durch `<rdf:Description>` eingeleitet
 - Durch das Attribut `<rdf:description ... rdf:ID=URIRef .../>` werden Elemente als Ressourcen identifizierbar gemacht
 - `<rdf:Description rdf:ID="Person123">`
 - Spezifikation eines Statements über eine Ressource, die zugleich definiert wird
 - `rdf:ID=URIRef` kann nur einmal im Dokument (`xml:base`) verwendet werden ([Unterschied zu `rdf:about`](#))
 - „#“ wird automatisch konkateniert ([Unterschied zu `rdf:about`](#))

Zusammenfassung und ergänzende Bemerkungen (7)

● RDF und RDF/XML

- Durch das Attribut `<rdf:description ... rdf:about=URIRef ...>` werden existierende Ressourcen referenziert bzw. noch nicht existierende definiert und referenziert
- Properties (Prädikat) sind Sub-Elemente eines durch `<rdf:Description>` (`rdf:Statement` siehe hinten) eingeleiteten Statements
 - Mehrere erlaubt
- Property-Values als Literal oder Ressource
 - `<dozent rdf:datatype="xsd:String">`
Martin Schaaf
`</dozent>`
 - `<dozent rdf:resource="#home/schaaf"/>`

Zusammenfassung und ergänzende Bemerkungen (8)

- RDF und RDF/XML
 - Property-Values können mehrere Werte haben
 - `<rdf:Bag>`, `<rdf:Seq>`, `<rdf:Alt>`

RDF/XML: RDF Type (1)

- RDF Typen

- Durch das `<rdf:type>` Property (vordefiniert) können Ressourcen typisiert werden
- Wert eines `<rdf:type>` Properties stellt eine Klasse dar, welche das Subjekt als Element enthält
- Mehrere Typen sind erlaubt

Reguläres
Property

```
<rdf:Description rdf:about="#Handheld">  
  <rdf:type rdf:resource="#BTMinorDeviceClass"/>  
  <majorClass rdf:resource="#Computer"/>  
</rdf:Description>
```



RDF/XML: RDF Type (2)

- RDF definiert eine abgekürzte Schreibweise für RDF Typen
 - URIRef des `<rdf:type>` Wertes wird QName des Tags

```
<rdf:Description rdf:about="#Handheld">  
  <rdf:type rdf:resource="#BTMinorDeviceClass"/>  
  <majorClass rdf:resource="#Computer"/>  
</rdf:Description>
```

```
xmlns="http://www.iis.uni-hildesheim.de/ont/2007/5/sensor.rdf#"  
xml:base="http://www.iis.uni-hildesheim.de/ont/2007/5/sensor.rdf"
```

```
<BTMinorDeviceClass rdf:ID="Handheld">  
  <majorClass rdf:resource="#Computer"/>  
</BTMinorDeviceClass>
```

RDF/XML: RDF Type (3)

- Falls mittels `<rdf:type>` mehrere Typen definiert wurden, nur ein Typ für abgekürzte Schreibweise

```
<rdf:Description rdf:about="#Handheld">  
  <rdf:type rdf:resource="#BTMinorDeviceClass"/>  
  <rdf:type rdf:resource="#ComputerClass"/>  
</rdf:Description>
```

äquivalent

Darf nur einmal mit
rdf:ID verwendet
werden

```
<BTMinorDeviceClass rdf:ID="Handheld">  
  <rdf:type rdf:resource="#ComputerClass"/>  
</BTMinorDeviceClass>
```

RDF/XML: Reifikation

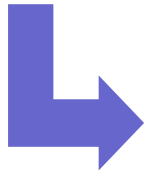
- RDF bietet auch die Möglichkeit Aussagen über Statements zu machen
 - Eher selten benutzt
 - RDF Statement wird selbst zur Ressource
 - Modellsemantik problematisch
 - **rdf:subject, rdf:predicate und rdf:object (Properties)** ermöglichen Zugriff bzw. Definition der Teile von RDF Statements
 - **rdf:Statement** statt rdf:Description

RDF/XML: Reifikation (Beispiel)

```
<rdf:Description rdf:about="#Handheld">  
  <majorClass rdf:resource="#Computer"/>  
</rdf:Description>
```

Aussage über
„Handhelds“

Aussage über
Aussage(n) über
„Handhelds“



```
<rdf:Statement rdf:ID="StatementAboutHandheld">  
  <rdf:subject rdf:resource="#Handheld"/>  
  <rdf:predicate rdf:resource="#majorClass"/>  
  <rdf:object rdf:resource="#Computer"/>  
</rdf:Statement>
```

RDF/XML: Bemerkungen

- Für RDF/XML existieren weitere abkürzende Schreibweisen
 - Nicht immer sinnvoll für Lesbarkeit, sondern für „Koexistenz“ mit HTML Code für Web Browser

RDF: Anmerkungen

- „Schwache“ Semantik
 - Ressourcen und Properties im wesentlichen Symbole (QNames und URIRefs)
 - Lediglich typisierte Literale mit definierter Semantik (falls entsprechendes Typensystem gewählt z. B. XML Schema)
 - Durch `<rdf:type>` können Ressourcen Klassen zugeordnet werden (Ressource als Subjekt), bzw. Ressourcen als Klassen spezifiziert werden (Ressource als Property Wert), aber
 - Keine weiteren Einschränkungen für Klassen möglich
- RDF ist Sprache für strukturierte Meta-Daten, nicht zur Spezifikation von Konzepten