

## Intelligente Systeme im WWW

Dr. Anupriya Ankolekar, Dr. Pascal Hitzler, Dr. York Sure, Markus Krötzsch  
Sommersemester 2006

### Übung 1: XML und RDF (8.5.2006)

---

*Wichtige Mitteilungen zum Inhalt und Ablauf der Vorlesung werden mit Hilfe unserer Mailingliste [iswww@aifb.uni-karlsruhe.de](mailto:iswww@aifb.uni-karlsruhe.de) bekannt gegeben. Es wird empfohlen, sich dafür via <http://www.aifb.uni-karlsruhe.de/mailman/listinfo/iswww> anzumelden (siehe auch Vorlesungswebseite).*

**Aufgabe 1.1** Entscheiden Sie, ob die folgenden XML-Dokumente (bzw. Teile davon) *wohlgeformt* sind und schlagen Sie gegebenenfalls eine Korrektur vor.

(a) Ein Auszug aus einer früheren Startseite des AIFB ([www.aifb.uni-karlsruhe.de](http://www.aifb.uni-karlsruhe.de)):

```
<table width="100%">
<tr>
<td width="33%" align="left">
<a href="mailto:webmaster@aifb.uni-karlsruhe.de">
  <font class="footnote">&copy; AIFB 2001-2005</font></a></td>
<td width="33%" align="center">
&nbsp;<td width="33%" align="right">
<a href="http://www.aifb.uni-karlsruhe.de/about.html">
  <font class="footnote">About our SEmantic portAL</font></a></td>
</tr>
</table>
```

**Lösung** Falsch. Ein schließendes `<td>`-Tag fehlt und `&nbsp;` ist eventuell ungültig (falls keine entsprechende Entity definiert wurde<sup>1</sup>; für XHTML sind die bekannten HTML-Entities natürlich korrekt definiert). Inzwischen sind die Webseiten des AIFB übrigens in einem besseren Zustand.

(b) Ein vollständiges HTML-Dokument:

---

<sup>1</sup>In XML kann man Entities als Abkürzungen für Zeichenreihen verwenden. Dazu muss im Kopf des Dokumentes eine Angabe der Art `<!DOCTYPE bla [!ENTITY nbsp "Inhalt">]` gemacht werden. Ein Beispiel wird in der Lösung zu Aufgabe 1.6 (b) gegeben.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC
"-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1/transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Kontrollieren Sie Ihr HTML!</title></head>
<body>
  <h3>HTML-Check des W3C</h3>
  <b>Kostenloser HTML-Check:<br/><i>http://validator.w3.org/</b></i>
</body></html>

```

**Lösung** Falsch. Reihenfolge der Tags nicht korrekt.

(c) Dieses Dokument hält sich nicht an die Empfehlung, alle Tags klein zu schreiben:

```

<Liste laenge=4>
  <Listeneintrag pos=1>
    <Element/>Lorem ipsum<element/>
  </Listeneintrag>
  <listeneintrag pos=3/>
</Liste>

```

**Lösung** Falsch. Attribute müssen in Anführungszeichen stehen und XML ist case-sensitive.

**Aufgabe 1.2** Auf den Folien zur Vorlesung wurden bereits einige Kriterien für ein wohlgeformtes XML-Dokument genannt: jedes XML-Dokument enthält genau eine Wurzel und auf jedes öffnende Tag folgt ein entsprechendes schließendes Tag. Finden Sie, ausgehend von den obigen Beispielen, drei weitere Kriterien.

**Lösung** Geöffnete Tags werden in umgekehrter Reihenfolge geschlossen. Attributwerte stehen in Anführungszeichen. Groß- und Kleinschreibung der Tags wird beachtet. Nur zulässige Zeichen werden verwendet.

**Aufgabe 1.3** Geben Sie eine natürlichsprachliche Beschreibung für die folgende DTD an.

```

<!DOCTYPE kochbuch [
<!ELEMENT kochbuch (rezept)+ >
<!ELEMENT rezept (rezepttyp, titel, zutat+, arbeitsschritt+) >
<!ELEMENT titel (#PCDATA) >
<!ELEMENT zutat (#PCDATA) >
<!ELEMENT arbeitsschritt (#PCDATA) >
<!ATTLIST arbeitsschritt nummer CDATA #REQUIRED >
<!ELEMENT rezepttyp EMPTY >
<!ATTLIST rezepttyp name CDATA #REQUIRED >
]>

```

**Lösung** Ein Kochbuch besteht aus ein oder mehreren Rezepten. Ein Rezept hat einen bestimmten Typ und einen Titel, man braucht mindestens eine Zutat und benötigt einen oder mehrere Schritte. Jeder Schritt hat eine Nummer. Und ein Rezepttyp hat einen Namen.

**Aufgabe 1.4** Ist der folgende Ausschnitt einer XML Datei zur DTD aus der vorigen Aufgabe konform? Begründen Sie Ihre Entscheidung.

```
<kochbuch>
  <rezept>
    <rezepttyp name="Torten"/>
    <titel>Schwarzwälder Kirschtorte</titel>
    <arbeitsschritt nummer="19">Schokostreusel draufmachen</arbeitsschritt>
    <zutat>Sahne</zutat>
    <zutat>Schokostreusel</zutat>
  </rezept>
</kochbuch>
```

**Lösung** Nicht konform, da Reihenfolge von arbeitsschritt und zutat vertauscht wurden.

**Aufgabe 1.5** Geben Sie die korrekte Übersetzung der folgenden Elemente von XML Schema nach DTD an.

- (a) `<element name="foo" minOccurs="0" maxOccurs="unbounded">`
- (b) `<element name="bar" minOccurs="0">`
- (c) `<element name="baz" maxOccurs="unbounded">`

**Lösung**

- (a) `foo*`
- (b) `bar?`
- (c) `baz+`

**Aufgabe 1.6** Betrachten Sie die folgende XML-Datei:

```
<bib>
  <book year="1994">
    <title>TCP/IP Illustrated</title>
    <author><last>Stevens</last><first>W.</first></author>
    <publisher>Addison-Wesley</publisher>
    <price> 65.95</price>
  </book>
  <book year="1992">
    <title>Advanced Programming in the Unix environment</title>
```

```

    <author><last>Stevens</last><first>W.</first></author>
    <publisher>Addison-Wesley</publisher>
    <price>65.95</price>
</book>
<book year="2000">
    <title>Data on the Web</title>
    <author><last>Abiteboul</last><first>Serge</first></author>
    <author><last>Buneman</last><first>Peter</first></author>
    <author><last>Suciu</last><first>Dan</first></author>
    <publisher>Morgan Kaufmann Publishers</publisher>
    <price> 39.95</price>
</book>
<book year="1999">
    <title>The Economics of Technology and Content for Digital TV</title>
    <editor>
        <last>Gerbarg</last><first>Darcy</first>
        <affiliation>CITI</affiliation>
    </editor>
    <publisher>Kluwer Academic Publishers</publisher>
    <price>129.95</price>
</book>
</bib>

```

- (a) Geben Sie eine DTD an, für die diese XML-Datei gültig ist.  
(b) Geben Sie ein XML Schema an, für die diese XML-Datei gültig ist.

**Lösung** (a) Eine der möglichen Lösungen lautet:

```

<!ELEMENT bib (book* )>
<!ELEMENT book (title, (author+ | editor+ ), publisher, price )>
<!ATTLIST book year CDATA #REQUIRED >
<!ELEMENT author (last, first )>
<!ELEMENT editor (last, first, affiliation )>
<!ELEMENT title (#PCDATA )>
<!ELEMENT last (#PCDATA )>
<!ELEMENT first (#PCDATA )>
<!ELEMENT affiliation (#PCDATA )>
<!ELEMENT publisher (#PCDATA )>
<!ELEMENT price (#PCDATA )>

```

- (b) Die Lösung zu (b) ist als gesonderte XML-Datei auf den Webseiten der Vorlesung verfügbar.

**Aufgabe 1.7** Beschreiben Sie die folgenden RDF-Begriffe in eigenen Worten. Veranschaulichen Sie die Definitionen durch Beispiele.

(a) Klasse

**Lösung** Klassen oder Konzepte können als allgemeine Objektdefinitionen angesehen werden. Sie beschreiben eine Klasse von Objekten, nicht ein spezifisches Objekt. *Fluggesellschaft* ist ein Beispiel für eine allgemeine Klasse.

(b) Instanz

**Lösung** Im Gegensatz zu Klassen stellen Instanzen ein konkretes Objekt dar. *Lufthansa* wäre eine Instanz von *Fluggesellschaft*. In RDF(S) können manchmal auch Klassen als Instanzen behandelt werden, um über die Klasse als solche Aussagen zu machen. In den meisten Anwendungen überwiegen aber Instanzen, die keine Klassen sind.

(c) Property

**Lösung** Eine Property ist eine nicht taxonomische Relation. Ein Beispiel wäre *betreiben* im Sinne von Lufthansa betreibt das Flugzeug *Karlsruhe*.

Die folgenden Aufgaben sind Hausaufgaben und werden kurz am Anfang der nächsten Übung besprochen. Fragen zu den Aufgaben können an die Mailingliste gestellt werden.

**Aufgabe 1.8** Entscheiden Sie, ob folgende Aussagen zutreffen. Begründen Sie Ihre Antwort und schlagen Sie gegebenenfalls eine Korrektur vor.

- (a) RDF separiert Inhalt von Struktur.
- (b) RDF Schema definiert einfache Ontologien.
- (c) Bei RDF könnte man wegen ihrer Aufgabe und Bedeutung Proxy-Ressourcen auch als Surrogat-Ressourcen (also „Aushilfsressourcen“) bezeichnen.
- (d) RDF Bags unterscheiden sich von RDF Sequenzen durch ihre Kardinalität.
- (e) Um den Satz

„Steffen sagt, dass Sowas Buch *Knowledge Representation* eine gute Einführung in die Wissensrepräsentation ist“

vollständig in RDF zu repräsentieren, benötigt man Reifikation.

- (f) Will man Datentypen für RDF-Dokumente festlegen, benötigt man ein RDF Schema.
- (g) In RDF Schema ist Mehrfachvererbung nicht erlaubt.
- (h) In RDFS kann man ausdrücken, dass zwei Klassen keine gemeinsamen Individuen haben, wie z.B. die Klassen *Mann* und *Frau*.
- (i) Die `rdfs:subclassOf`-Property beschreibt eine Teil-Ganzes-Beziehung (Wie in „Der Reifen ist ein Teil des Autos.“).
- (j) Das folgende RDF ist korrekt.

```
<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='http://iswww.edu/test1#flugzeug' />
</rdf:RDF>
```

**Aufgabe 1.9** Betrachten Sie die folgende RDF-Repräsentation:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:iswww="http://iswww.edu/test1#"
>

<rdf:Description rdf:about="http://iswww.edu/test1#Deutschland">
  <rdf:type rdf:resource="http://iswww.edu/test1#land"/>
</rdf:Description>

<rdf:Description rdf:about="http://iswww.edu/test1#hauptstadt_von">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
    ns#Property"/>
  <rdfs:domain rdf:resource="http://iswww.edu/test1#stadt"/>
  <rdfs:range rdf:resource="http://iswww.edu/test1#land"/>
</rdf:Description>

<rdf:Description rdf:about="http://iswww.edu/test1#Land">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:label xml:lang="en">Country</rdfs:label>
</rdf:Description>

<rdf:Description rdf:about="http://iswww.edu/test1#berlin">
  <rdfs:label xml:lang="en">Berlin</rdfs:label>
  <rdf:type rdf:resource="http://iswww.edu/test1#stadt"/>
  <iswww:hauptstadt_von rdf:resource="http://iswww.edu/test1#deutschland"/>
</rdf:Description>

<rdf:Description rdf:about="http://iswww.edu/test1#stadt">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:label xml:lang="en">http://iswww.edu/test1#City</rdfs:label>
</rdf:Description>

</rdf:RDF>
```

- (a) Beschreiben Sie in natürlicher Sprache, was durch diese Repräsentation ausgedrückt wird.
- (b) Erstellen Sie eine graphische Darstellung der obigen RDF-Repräsentation.