XML-Praxis

XML – Extensible Markup Language

Jörn Clausen
joern@TechFak.Uni-Bielefeld.DE

Übersicht

- Woher? Wohin? Warum?
- Bestandteile von XML
- XML-Dokumente erstellen und bearbeiten

Was ist XML?

- Daten sind strukturiert (Texte, Bilder, Meßergebnisse, . . .)
- maschinelle Verarbeitung erfordert Kenntniss der Strukturen
- gesucht: Formalismus, um beliebige Strukturen zu beschreiben
- XML kann textuelle Daten strukturieren
- standardisierte Methoden zur Verarbeitung von XML

Ursprünge: SGML

- Standard Generalized Markup Language (ISO 8879:1986)
- *keine* Markup-Sprache, sondern Grammatik-Sprache
- maßgeschneidertes Vokabular für unterschiedliche Anwendungen
- Problem: komplexe Spezifikation, Parser schwer zu implementieren
- kommerzielle Produkte, vor allem im Verlagswesen
- Instanz + DTD + SGML Declaration
- 1989: Hypertext Markup Language (HTML), World Wide Web
- Anfang/Mitte 1990er Jahre: Browser Wars

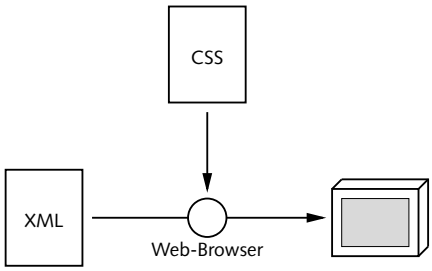
Ausweg: XML

- Entwicklung ab 1996 durch WWW Consortium
- einfache Spezifikation, Parser leicht zu implementieren
- *extensible*: Spracherweiterungen möglich/erwünscht
- DTD optional, Instanz kann *stand alone* sein
- Sprachumfang kann wachsen, Bedürfnissen angepaßt werden
- aber immer noch Gefahr von Wildwuchs

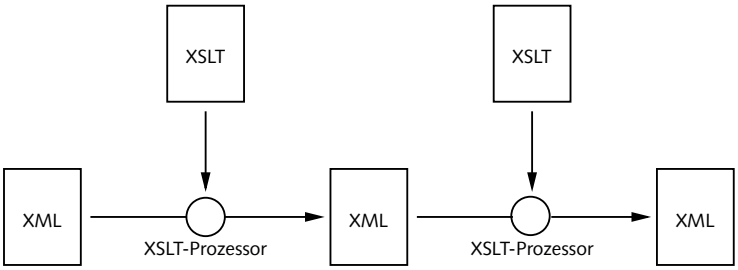
XML – Und dann?

- XML zur Datenrepräsentation
- „Darstellung“ sekundäres Problem
- XML muß weiterverarbeitet werden

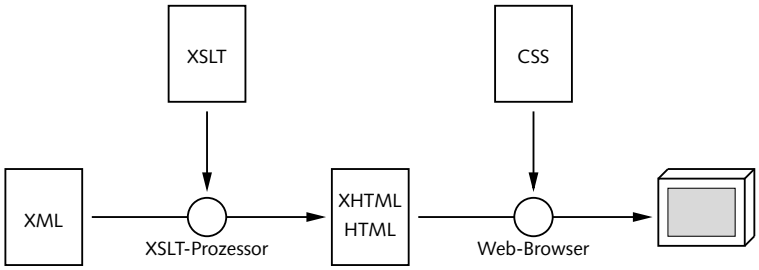
XML anzeigen



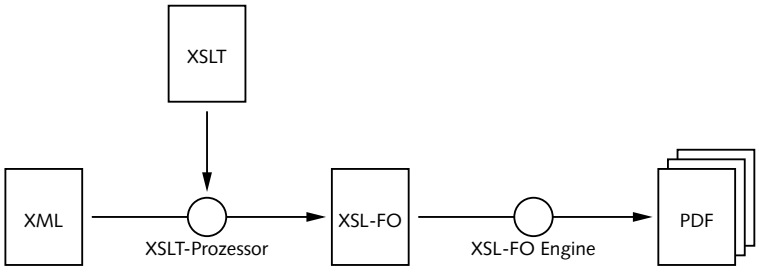
XML transformieren



XML transformieren und anzeigen



XML formatieren



XML verarbeiten

- (fast) alles ist XML
- wenige Werkzeuge nötig (XML-Parser, XML-Editor, . . .)
- wiederverwendbare Komponenten
- Textformat Unicode: portabel, einfach zu verarbeiten
- offene Standards, viele Open Source-Lösungen

Aufgabe

- Sieh Dich etwas auf den Web-Seiten des W3-Konsortiums um:

  http://www.w3.org

- Finde die Spezifikationen („Recommendations“) der folgenden Standards:
  - XML 1.0 (Third Edition)
  - XSLT 1.0
  - XPath 1.0
- Sieh Dir die Web-Seiten der „Organization for the Advancement of Structured Information Standards“ (OASIS) an:

  http://www.oasis-open.org

ein Beispiel

```
<?xml version="1.0"?>
<presentation status="draft" date="2002-10-04">
  <title>XML &amp; Friends for Dummies</title>
  <author>Joe User</author>
  <slide>
    <title toc="yes">What is XML?</title>
    <ilist>
      <item>XML is not a markup language (unlike HTML)</item>
      <item>XML instances can be <emph>well formed</emph> or
        even <emph>validating</emph></item>
      <item>XML stands for &xml;</item>
    </ilist>
  </slide>
  <slide>...</slide>
</presentation>
```

Aufbau von XML

- XML-Datei beginnt mit *XML declaration*

```
<?xml version="1.0"?>
```

- seit 4.2.2004: XML 1.1
- Empfehlung: bis auf weiteres XML 1.0 verwenden
- verwendete Kodierung

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

- sonst Unicode (UTF-8), Obermenge von ASCII
- Kodierung im Dokument nicht unproblematisch

Elemente (elements)

- öffnendes und schließendes *tag*

```
<item>XML is not a ...</item>
```

- Elemente können geschachtelt werden

```
<ilist>
  <item>XML is not a ...</item>
  <item>... <emph>well formed</emph> ...</item>
</ilist>
```

- keine Minimierungsregeln
- leeres Element

`<hr/>` statt `<hr></hr>` statt `<hr>`

Elemente, cont.

- Schachtelung muß „passen“

```
<a> <b> ... </a> </b>
```

- Groß/Klein-Schreibung relevant

```
<html> ... </HTML>
```

- XML-Dokument muß genau ein äußerstes Element enthalten

```
<?xml version="1.0"?>
<presentation>
  ...
</presentation>
<comment>
  ...
</comment>
```

Aufgaben

- Erstelle mit Hilfe des Emacs eine einfache Literaturliste in Form einer XML-Datei. Es sollen mehrere Bücher mit ihrem Titel und ihrem/n Autor(en) erfaßt werden.

  Achte auf die Dateiendung `.xml` und darauf, daß der XML-Mode verwendet wird.

  Verwende die folgenden Tastenkombinationen beim Schreiben der XML-Datei:

  - TAB-Taste: Zeile einrücken
  - `CTRL-C /`: schließendes Tag einfügen

- Überprüfe mit Hilfe der Programme `xmlwf` und `xmllint`, ob die Datei korrekt ist.

- eine mögliche Lösung:

```
<?xml version="1.0"?>
<literature>
  <book>
    <title>The Lord of the Rings</title>
    <authors>
      <author>John Ronald Reuel Tolkien</author>
    </authors>
  </book>
  <book>
    <title>Internetworking With TCP/IP</title>
    <authors>
      <author>Douglas E. Comer</author>
      <author>David L. Stevens</author>
    </authors>
  </book>
</literature>
```

Kommentare

- Kommentare kennzeichnen:

```
<!-- fix some spelling errors here -->
```

- mehrzeilige Kommentare möglich
- `--` darf nicht im Kommentar vorkommen
- keine geschachtelten Kommentare

Attribute (attributes)

- Zusatzinformationen zu Elementen

```
<presentation status="draft" date="2002-10-04">
```

- nur im öffnenden tag
- Anführungszeichen " (double quote) oder ' (single quote)
- Attribut darf nur einmal vorkommen

```
<presentation lang="en" lang="de">
```

- Design-Frage: Wann Elemente, wann Attribute?

```
<date y="2002" m="10" d="7"/>
<date><y>2002</y><m>10</m><d>7</d></date>
```

Attribute, cont.

- Attribute sind nicht weiter strukturierbar:

```
<interval start_year="2004" start_month="10" start_day="11"
          end_year="2005" end_month="2" end_day="4"/>
```

vs.

```
<interval>
  <start year="2004" month="10" day="11"/>
  <end year="2005" month="2" day="4"/>
</interval>
```

vs.

```
<interval>
  <tstamp type="start" year="2004" month="10" day="11"/>
  <tstamp type="end" year="2005" month="2" day="4"/>
</interval>
```

Aufgaben

- Erweitere die Datei aus der letzten Aufgabe um folgende Angaben:
  - Erscheinungsjahr
  - Verlag
  - ISBN-Nummer
  - Sachgebiet
  - Zustand (neuwertig, gelesen, zerfleddert, ...)
  - persönlicher Kommentar
- Überprüfe die Datei wieder mit `xmlwf` bzw. `xmllint` auf syntaktische Korrektheit.

- eine mögliche Lösung:

```
<book isbn="0-04-823045-6" condition="used">
  <title>The Lord of the Rings</title>
  <authors>
    <author>John Ronald Reuel Tolkien</author>
  </authors>
  <publisher year="1953">George Allen and Unwin</publisher>
  <genre>english fiction</genre>
  <comment>A classic book with a classic theme.
    Better than the movie.</comment>
</book>
```

Entitäten (entities)

- Makros und Sonderzeichen
- in XML vordefinierte *entity references*

```
&amp; &lt; &gt; &apos; &quot;
```

- weitere können definiert werden
- *character references*: Zugriff auf beliebige Unicode-Zeichen

```
&#xA9; 2002 by J&#xF6;rn Clausen
```

Aufgaben

- Die Datei `entities.xml` verwendet einige character references. Sieh sie Dir mit Mozilla oder Opera an.
- Eine Übersicht über alle *code points* von Unicode findest Du unter

  http://www.unicode.org/charts/

  Versuche, weitere „exotische“ Zeichen mit Hilfe von character references einzufügen.
- Erweitere die XML-Deklaration auf

  ```
  <?xml version="1.0" encoding="ISO-8859-1"?>
  ```

  Überprüfe die Datei mit `xmllint`. Was erwartest Du für ein Ergebnis?

- Die Datei ist weiterhin valide. Die Kodierung gibt den Zeichensatz an, in dem die XML-Datei geschrieben ist, nicht den, den sie verwendet. Da durch die character references gerade auf die direkte Eingabe von “Sonderzeichen” verzichtet wurde, enthält die Datei nur ASCII-Zeichen, und ist damit ebenfalls UTF8- und ISO-8859-1-kodiert, da es sich hierbei um Obermengen von ASCII handelt.

Aufgaben

- Beschreibe die folgenden Dinge mit Hilfe von XML:
  - Fußball-Tabelle
  - Liste mit Fußball-Ergebnissen
  - Periodensystem der chemischen Elemente
  - DNA-Sequenz
  - Gedicht
  - Lebenslauf
  - Brief
  - Roman
- Wofür ist XML ungeeignet?