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

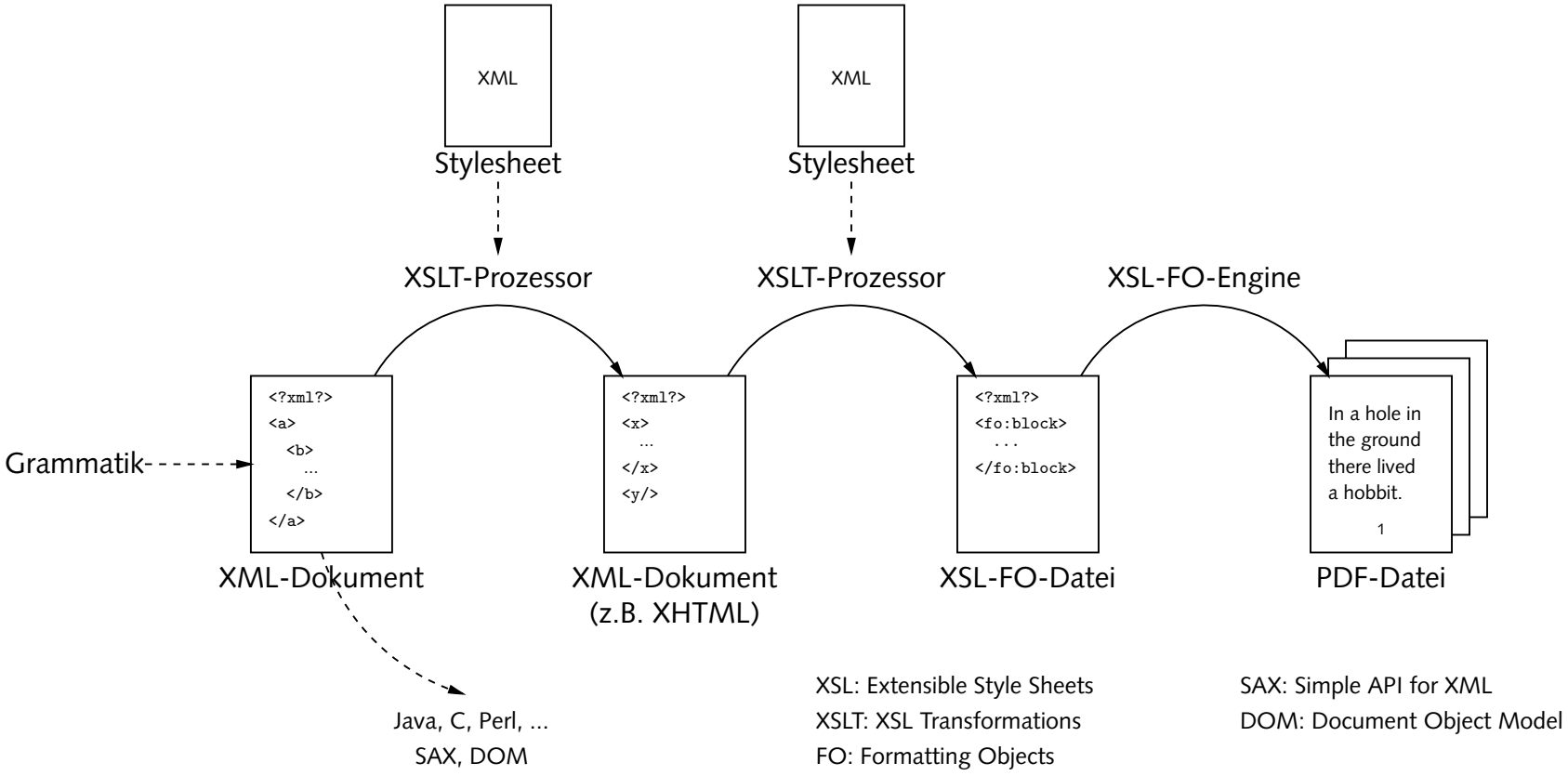
Was ist XML?

- 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

Ursprung von XML: SGML

- Standard Generalized Markup Language
- 1986 als ISO-Standard 8879 verabschiedet
- *keine* Markup-Sprache, sondern Grammatik-Sprache
- Einsatz vor allem im Verlagswesen
- 1989: Hypertext Markup Language, World Wide Web
- Problem: komplexe Regeln, Parser schwer zu implementieren
- ab 1996: Entwicklung von XML
- SGML−−, nicht HTML++

Verarbeitung von XML



Verarbeitung von XML, cont.

- (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 (Second 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"?>
  ```

- bisher nur Version 1.0 spezifiziert, neue Versionen in Planung
- 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 oberstes Element enthalten

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

Aufgaben

- Was stimmt mit den Dateien `bogus1.xml`, `bogus2.xml` und `bogus3.xml` nicht? Überprüfe sie mit Hilfe der Programme `xmlwf` und `xmllint`. Verbessere die Fehler.
- Erstelle mit Hilfe des Emacs eine einfache Telefonliste in Form einer XML-Datei. 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

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>
  ```

Aufgaben

- Erweitere die XML-Datei aus der letzten Ausgabe um folgende Angaben:
  - Anschrift
  - EMail-Adresse
  - Geburtsdatum
- Pro Person sollen mehrere Telefonnummern gespeichert werden können. Es soll zwischen Festnetz, Handy und Fax unterschieden werden.
- Die bevorzugte Form der Kontaktaufnahme (Telefon, Handy, Fax, EMail) soll kenntlich gemacht werden.

Entitäten (entities)

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

  `& < > &apos; &quot;`
- weitere können definiert werden
- *character references*: Zugriff auf beliebige Unicode-Zeichen

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

Aufgaben

- Was passiert, wenn Du die Zeichen

  `& < > ' "`

  direkt im Text (zwischen zwei Tags) oder in einem Attribut verwendest? Teste wieder mit `xmlwf` bzw. `xmllint`.

- Was passiert, wenn Du die character reference

  `&#x7;`

  verwendest?

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?