

XML-Praxis

XML-Anwendungen

Jörn Clausen

`joern@TechFak.Uni-Bielefeld.DE`

Übersicht

- DocBook
- Software-Installation dokumentieren

DocBook

- SGML/XML-Sprache für technische Dokumentationen
- Norman Walsh, OASIS
- Vorlage: man-pages (NAME, SYNOPSIS, DESCRIPTION, ...)
- Stylesheets für Transformation in HTML und XSL-FO
- leicht anpaßbar

Dokumentenstruktur

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.3//EN"
    "file:///vol/xml/share/schemas/dtds/docbook-4.3/docbookx.dtd">

<book>
  <title>XML-Praxis</title>
  <chapter>
    <title>XML - Extensible Markup Language</title>
    <section>
      <title>Was ist XML?</title>
      <para>XML ist ...</para>
      <para>...</para>
    </section>
  </chapter>
</book>
```

Aufgaben

- Öffne die Datei `manual.xml` mit dem Emacs. Erzeuge mit Hilfe des XML-Modes ein kleines Dokument. Laß Dir durch den Editor anzeigen, an welchen Stellen welche Elemente und Attribute erlaubt sind.

Erstelle mindestens zwei chapter mit jeweils zwei sections.

- Das Buch „DocBook: The Definitive Guide“ (O'Reilly) ist komplett online erhältlich:

<http://www.oreilly.com/catalog/docbook/chapter/book/docbook.html>

Eine alphabetische Liste aller Elemente findet sich in

<http://www.oreilly.com/catalog/docbook/chapter/book/refelem.html>

Aufgaben

- Wende das Stylesheet `db2html.xsl` auf `manual.xml` an. Leite die Ausgabe in eine Datei um:

```
$ xsltproc db2html.xsl manual.xml > manual.html
```

Sieh Dir die entstehende HTML-Datei mit einem Web-Browser an.

- Wende das Stylesheet `db2chunk.xsl` auf `manual.xml` an. Eine Dateiumleitung ist in diesem Fall nicht nötig.

```
$ xsltproc db2chunk.xsl manual.xml
```

Welche Dateien entstehen? Sieh sie Dir an.

- Sieh Dir die beiden Stylesheets an. Wie funktionieren sie?

Aufgaben

- Die Stylesheets lassen sich über Parameter verändern. Füge in `db2html.xsl` folgende Zeile *nach* `<xsl:import>` ein:

```
<xsl:param name="html.stylesheet" select="'manual.css'"/>
```

Sieh Dir den HTML-Code an. Was bewirkt dieser Parameter?

- Erstelle eine CSS-Datei `manual.css`, mit der Du z.B. die Formatierung der Kapitel-Überschriften veränderst.

Aufgaben

- Erweitere die Datei `db2chunk.xml` ebenfalls um den CSS-Parameter, und füge einen weiteren Parameter ein:

```
<xsl:param name="html.stylesheet" select="'manual.css' "/>  
<xsl:param name="chunk.section.depth" select="0" />
```

Was passiert, wenn Du nun die XML-Datei mit diesem Stylesheet verarbeitest?

Aufgaben

- Übersetze die XML-Datei mit dem Stylesheet `db2fo.xsl` und lenke die Ausgabe in eine Datei um:

```
$ xsltproc db2fo.xsl manual.xml > manual.fo
```

Verarbeite diese Datei mit dem Programm `xep` weiter:

```
$ xep manual.fo
```

Welche Datei entsteht dabei? Sieh sie Dir an.

Aufgaben

- Auch die Ausgabe von XSL-FO läßt sich parametrisieren. Füge die folgenden Anweisungen in `db2fo.xsl` ein und rufe erneut `xsltproc` und `xep` auf:

```
<xsl:param name="paper.type" select="'A4'"/>  
<xsl:param name="double.sided" select="1"/>
```

- Weitere Informationen zur Anpassung der Stylesheet finden sich in „DocBook XSL: The Complete Guide“:

<http://www.sagehill.net/docbookxsl/>

Daten jenseits von XML

- Aufgabe: Perl-Module installieren
- Problem: Abhängigkeiten zwischen verschiedenen Modulen
- beschreibe Daten in XML:

```
<module name="XML-Parser">  
  <instance version="2.31" installer="joern" date="2002-08-19"/>  
  <depends>expat 1.95.4</depends>  
</module>
```

```
<module name="XML-XPath">  
  <instance version="1.12" installer="joern" date="2002-09-11"/>  
  <instance version="1.13" installer="joern" date="2004-07-09"/>  
  <uses module="XML-Parser"/>  
</module>
```

Daten jenseits von XML, cont.

- erzeuge Web-Seite mit tabellarischer Übersicht
 - alphabetisch sortiert
 - aktuelle Version hervorheben, ältere Versionen dokumentieren
 - Abhängigkeiten als Hyperlinks
 - „A verwendet B“ → „B wird von A verwendet“
- visualisiere Abhängigkeitsgraph
 - Softwarepaket „GraphViz“
 - erzeuge Graphbeschreibung im ASCII-Format „dot“

Daten jenseits von XML, cont.

- Text erzeugen:

```
<xsl:output method="text" />
```

```
<xsl:template match="/modules">
```

```
  digraph PERL
```

```
    size="10,7.5";
```

```
    rotate=90;
```

```
    rankdir=LR;
```

```
    ratio=fill;
```

```
    label="<xsl:value-of select="@version" />";
```

```
    node [fontname=Helvetica];
```

```
  <xsl:apply-templates />
```

```
</xsl:template>
```

Daten jenseits von XML, cont.

Module Name	Version	Author	Dependencies / Notes	Installation Date
Tk	800.24	joern.		2002-08-19
URI	1.21	joern.	uses: Business-ISBN, MIME-Base64 is used by: SOAP-Lite, libwww-perl	2002-08-19
Unicode-String	2.06	joern.	convert XML-XPath results to ISO-8859-1 uses: MIME-Base64	2002-09-11
XML-DOM	1.39	joern.	uses: XML-Parser, XML-RegExp, libwww-perl is used by: bioperl	2002-08-22
XML-NamespacesSupport	1.07	joern.	is used by: XML-SAX	2002-08-21
XML-Node	0.11	joern.	is used by: bioperl	2002-08-21
XML-Parser	2.31	joern.	is used by: SOAP-Lite, XML-DOM, XML-Twig, XML-XPath, bioperl, libxml-perl expat 1.95.4	2002-08-19
XML-RegExp	0.03	joern.	is used by: XML-DOM	2002-08-22
XML-SAX	0.10	joern.	uses: XML-NamespacesSupport	2002-08-21
XML-Twig	3.05	joern.	uses: XML-Parser is used by: bioperl	2002-08-21
XML-Writer	0.4	joern.	is used by: bioperl	2002-08-21
XML-XPath	1.12	joern.	uses: XML-Parser	2002-09-11

