



Single Source Publishing mit XML

Thomas Nindel

Betreuer: Hans J. Günther

Bibliografische Beschreibung und Autorenreferat

NINDEL, Thomas, Berufsakademie Sachsen, Staatliche Studienakademie Dresden, Studienrichtung Informationstechnik, Praxisarbeit 1 Semester, 15 Seiten, 2 Literaturquellen, 2 Anlagen.

Es werden aktuelle XML-Technologien erklärt und es wird ein Weg gezeigt, wie diese bei der Realisierung von Single Source Publishing verwendet werden können.

Verwendete Abkürzungen / Begriffe	4
1. Einführung	5
1.1. Abstract	5
1.2. Aufgabenstellung	5
2. Einführung XML	5
2.1. XML	5
2.2. XSL	7
2.3. XSL-FO	8
2.4. DTD	10
2.5. Sonstiges	10
3. SSP mit XML/XSL/FO	10
3.1. Daten	10
3.2. Dateneingabe, Datenstruktur	11
3.3. Ausgabe	11
3.3.1. Processing HTML	12
3.3.2. Processing PDF	12
4. Fazit	13

Verwendete Abkürzungen / Begriffe

SSP Single Source Publishing, das automatische Erzeugen von Inhalt für mehrere Medien aus einer Datenbasis

1. Einführung

1.1. Abstract

Es werden aktuelle XML - Technologien erklärt und es wird ein Weg gezeigt, wie diese bei der Realisierung von Single Source Publishing verwendet werden können.

1.2. Aufgabenstellung

Erklären Sie aktuelle XML–Technologien und erläutern sie eine Möglichkeit wie man diese Technologien für ein SSP–System einsetzen kann.

2. Einführung XML

Die Extensible Markup Language, abgekürzt XML, beschreibt eine Klasse von Datenobjekten, genannt XML-Dokumente, und beschreibt teilweise das Verhalten von Computer-Programmen, die solche Dokumente verarbeiten. XML ist ein Anwendungsprofil oder eine eingeschränkte Form von SGML, der Standard Generalized Markup Language. Durch ihre Konstruktion sind XML-Dokumente konforme SGML-Dokumente. XML ist ein gutes Hilfsmittel für die strukturierte Speicherung von Daten in baumförmiger Art und Weise.

2.1. XML

XML-Dokumente sind aus Speicherungseinheiten aufgebaut, genannt Entities, die entweder analysierte (parsed) oder nicht analysierte (unparsed) Daten enthalten. Analysierte Daten bestehen aus Zeichen, von denen einige Zeichendaten und andere Markup darstellen. Markup ist eine Beschreibung der Aufteilung auf Speicherungseinheiten und der logischen Struktur des Dokuments. XML bietet einen Mechanismus an, um Beschränkungen der Aufteilung und logischen Struktur zu formulieren.

Ein Software-Modul, genannt XML-Prozessor, dient dazu, XML-Dokumente zu lesen und den Zugriff auf ihren Inhalt und ihre Struktur zu erlauben. Es wird angenommen, dass ein XML-Prozessor seine Arbeit als Teil eines anderen Moduls, genannt Anwendung, erledigt. Diese Spezifikation beschreibt das notwendige Verhalten eines XML-Prozessors, soweit es die Frage betrifft, wie er XML-Daten einlesen muss und welche Informationen er an die Anwendung weiterreichen muss.

XML-Dokumente bestehen aus einem oder mehreren Elementen, welche wiederum aus Elementen, Attributen oder Text bestehen können. Ein Element besteht aus einem öffnenden und einem schliessenden Tag. Das öffnende Tag kann Attribute enthalten (wie HTML auch).

```
<Preis waehrung='Euro'>24.34</Preis>
```

Das Element hat den Namen „Preis“ und den Inhalt „24.34“. Es hat ein Attribut mit dem Namen „Währung“ und dem Inhalt „Euro“.

XML Dokumente müssen, im Gegensatz zu HTML, „wellformed“ sein, d.h. dass Elemente mit schließendem Tag in korrekter Schachtelungsfolge abgeschlossen werden müssen. Für leere Elemente ist eine Abkürzung möglich:

```
<leer/>
```

Des Weiteren bietet XML die Möglichkeit von Kommentaren:

```
<!-- Kommentar -->
```

und so genannten Processing Instructions, die dem Parser spezifische Einstellungen wie z.B. zu verwendenden Zeichensatz mitteilen oder mit denen Stylesheets eingebunden werden können. Außerdem können noch DTDs eingebunden werden. Eine einfache XML Datei sieht folgendermaßen aus:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="style.xsl"?> <!--
- einbinden des Stylesheet -->
<buch>
  <name>
    XML Leicht gemacht
```

```
</name>
<preis waehrung="Euro">
  54
</preis>
</buch>
```

2.2. XSL

XSL, die eXtensible Stylesheet Language, besteht aus einer Reihe von Markups, die Regeln für eine Formatierung von XML-Dokumenten darstellen. Dabei teilt sich der XSL-Standard in die Transformationssprache (XSLT) und in die Formatierungssprache (XSL-FO oder kurz FO).

Mit Hilfe der Transformationssprache können XML Strukturen in andere XML Strukturen gewandelt werden. Da HTML mit XML große Ähnlichkeit hat (im Prinzip ist HTML nur eine XML-DTD), kann also mit Hilfe einer XSL-T Transformation ein HTML-Dokument aus jeder beliebigen XML-Basis erzeugt werden. XSLT bietet zahlreiche Automatismen wie z.B. Überschrift- und Aufzählungsnummerierung. Nicht vorhandene Features können durch einbinden von Skripten (z.B. Perl) realisiert werden.

Die Transformation wird mit so genannten „Templates“ formuliert, die wiederum in XML-Dateien gespeichert werden. Also wird XML mit XML transformiert. Die Templates werden für einen bestimmten Kontext (in der Regel für jedes Element) in der XML-Datei formuliert und werden bei Übereinstimmung (Match) vom XSL-Prozessor aufgerufen. Ein einfaches Beispiel für das obenstehende XML-Dokument sieht folgendermaßen aus:

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="uri:xsl">

  <xsl:template match='/'>
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match='buch'>
    <p style="font-size=20pt; background-color=#ffaa00;
color=white">
```

```
    Buch
  </p>
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match='name'>
  Name: <xsl:value-of /><br/>
</xsl:template>

<xsl:template match='preis'>
  Preis: <xsl:value-of select="."/> <xsl:value-of
select="./@waehrung"/>
</xsl:template>

</xsl:stylesheet>
```

Dieses Stylesheet erzeugt eine HTML-Datei mit folgendem Aussehen:



Name: XML Leicht gemacht

Preis: 54 Euro

Dieses einfache Beispiel zeigt bereits die Philosophie von Publishing mit XML: Der Inhalt ist vom Layout komplett getrennt. Eine Änderung des Inhaltes (z.B. Preis) muss nur in der XML-Datei geschehen, das Stylesheet muss nicht geändert werden.

2.3. XSL-FO

Die XSL-FO (XSL-Formatting Objects) ist eine definierte XML – Struktur, die Informationen zur Formatierung in so genannten „Block-Layouts“ enthält. FO findet hauptsächlich Einsatz bei der Formatierung von XML-Daten für Printmedien. Eine Implementation von FO wurde von der Apache XML Group mit dem Programm „FOP“ bereitgestellt, welche aus FO-Dateien PDF-Dokumente erzeugen kann. Die FO-Dateien sind



ebenfalls in XML formuliert, so dass sie sich mit einer XSL-T-Transformation aus XML-Dateien erstellen lassen. Trotz einer Versionsnummer die eine Null vor dem Punkt hat, bietet FOP bereits beeindruckende Features die durchaus eine professionelle Seitengestaltung zulassen.

In der FO-Datei wird zuerst das Grundlayout der Seiten wie Format, Header und Footer, eventuelle Seitennummerierungen usw. festgelegt. Dann wird, ähnlich wie bei TeX, das Dokument mit XML-Markups beschrieben. Da die Spezifikation von FO über 1000 Seiten lang ist und selbst ein minimaler Teil den Rahmen dieser Arbeit sprengen würde, wird an dieser Stelle auf eine nähere Beschreibung von FO verzichtet.

2.4. DTD

Die DTD, oder Dokument Type Definition, ist ein Hilfsmittel zur Definition der Struktur einer XML-Datei. Es können einfache Regeln formuliert werden, die festlegen, welches Element oder Attribut wo und wie oft in der Datenstruktur vorkommen darf. Da sich jedoch herausstellt, dass DTD für die Strukturierung komplexer Datenhierarchien zu wenig Features bietet, wird auf einen näheren Einblick hier verzichtet. Um nur ein Beispiel für eine sehr bekannte DTD zu geben: Die XHTML-Spezifikation ist eine XML-DTD.

2.5. Sonstiges

Weitere XML-Technologien sind z.B. SCHEMA [1], eine erweiterte Form von DTDs und SOAP [2] zum direkten Zugriff in ein XML-Dokument im WWW. Alle XML-Technologien werden vom World Wide Web Consortium (W3C) betreut [3].

3. SSP mit XML/XSL/FO

Derzeit präsentieren produzierende Firmen ihre Waren auf unterschiedlichen Medien: Print, Internet und CDROM. Der Inhalt für jedes Medium wird von Hand erstellt und kostet dadurch enorm viel Geld. Selbst kleine Änderungen kann die Firma selbst nicht durchführen und muss wiederum Firmen beauftragen, die diese Änderungen durchzuführen. Mit XML/XSL/FO bietet sich jetzt die Möglichkeit, aus einer Datenquelle verschiedene Medien zu bedienen und das mit einem hohen Automatisierungsgrad. Dieses Verfahren wird als Single Source Publishing bezeichnet.

3.1. Daten

Wie schon oben angesprochen, bietet sich für Single Source Publishing an, die Daten im XML-Format zu speichern, da sie dann durch XSL-Transformationen ausgegeben werden. Ein weiterer Vorteil von XML als Datenformat ist die Möglichkeit des Austauschs mit anderen Systemen. Da sich beim Im- oder Export von Daten lediglich die Struktur unterscheidet, ist eine Schnittstelle zu anderer, XML-unterstützender Software mit XSL-

Transformationen einfach realisierbar. Des Weiteren gibt es für die Speicherung und Verwaltung von XML eine Reihe spezieller Datenbanken (z.B. POET), so dass das Entwickeln spezieller Datenbanken nicht erforderlich ist.

3.2. Dateneingabe, Datenstruktur

Ein sehr wichtiger Schritt zur strukturierten Erfassung von Daten ist die Strukturierung selbst. Firmen sind in ihren Produktkatalogen oft inkonsequent, wenn es um die Zuordnung von Produkten in Produktgruppen geht. Da XML jedoch keine „Links“ zwischen den Datenebenen erlaubt, ist ggf. eine Anpassung der Struktur der Daten notwendig.

Wenn die Daten strukturiert sind und die Hierarchiestufen bekannt sind, kann ein Format für die XML-Daten definiert werden. Da eine willkürliche Benennung der XML-Struktur (betrifft Element- und Attributnamen etc.) jedoch erfordert, dass die Dateneingabesoftware jedes Mal angepasst wird, muss ein „universales“ Datenformat geschaffen werden. Dies beinhaltet Strukturierungselemente wie Aufzählungen (z.B. „Produktgruppe“) sowie Objekte („Produkt“) und deren Eigenschaften („Preis“). Die meisten Anwendungen werden so in einem Format erfasst.

Wenn also eine XML-Struktur fest steht, ist es relativ einfach, einen komfortablen Editor zu schaffen, mit dem man Daten erfassen und ändern kann. Dies geschieht in einer objektorientierten Art und Weise.

3.3. Ausgabe

Die Ausgabe der Daten, die ja im XML-Format vorliegen, geschieht über eine zweistufige XSL-Transformation. Die erste Stufe erzeugt ein XML-Dokument (angelehnt an die DocBook-DTD [4]), welches eine dem Publizieren angelehnte Struktur aufweist.. Außerdem werden nur die zur Ausgabe ausgewählten Daten verarbeitet. In einer zweiten Konvertierungsstufe erfolgt eine Trennung und das Processing für die spezifischen Ausgabemedien.

3.3.1. Processing HTML

Aus dem Zwischenformat wird über ein weiteres XSL-Stylesheet (speziell für HTML) ein HTML-Dokument mit entsprechender Formatierung erzeugt. Dieses Dokument wird unter zu Hilfenahme eines Perlskript aufgeteilt, um es auf einem Webserver anbieten zu können. Ein weiteres Perlscript wandelt absolute Hyperlinks im HTML in relative um.

3.3.2. Processing PDF

Aus dem Zwischenformat wird mit einem für die FO-Ausgabe spezialisiertem Stylesheet der FO-Baum konstruiert und mit entsprechenden Formatierungen belegt. Danach stellt ein Perlscript die Verfügbarkeit extern referenzierter Daten (z.B. Bilder) sicher.

Der letzte Schritt erzeugt mit FOP aus dem FO-Baum das fertige PDF-Dokument

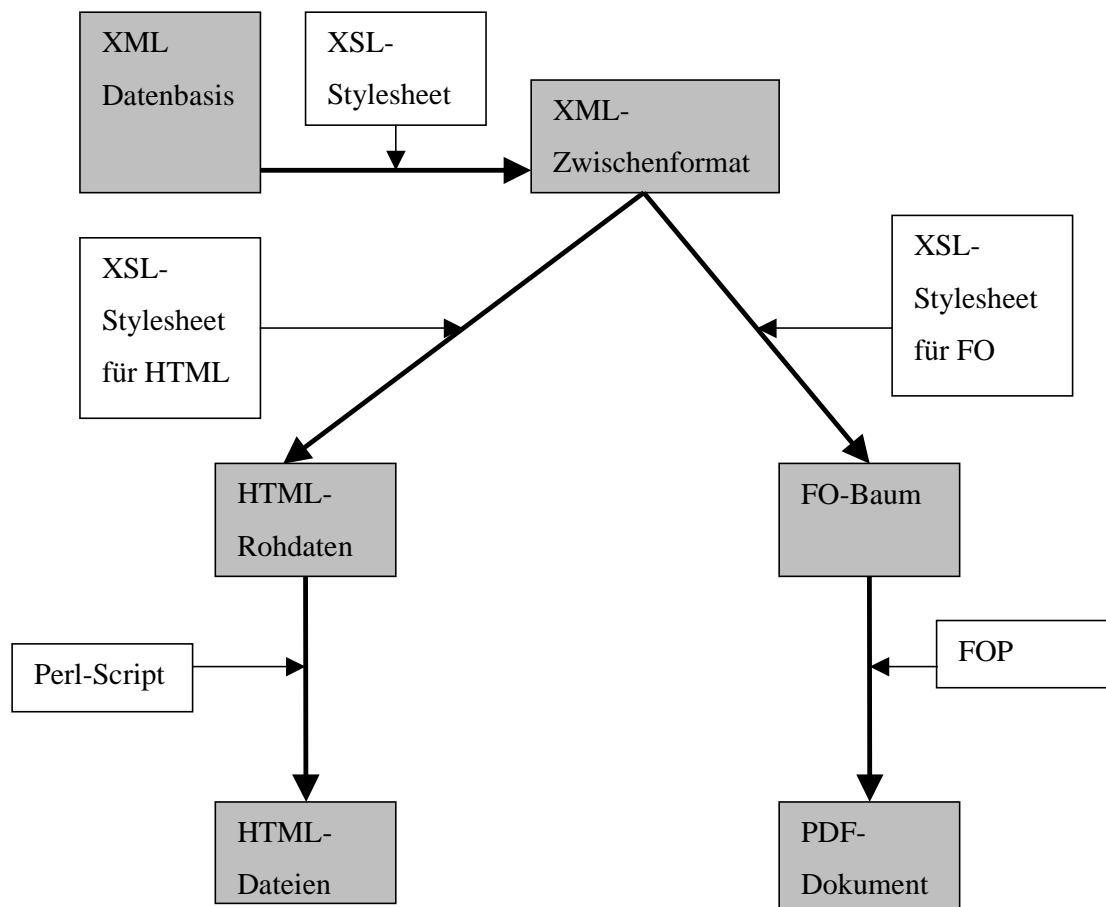


Abbildung 1: Schema des SSP mit XML

4. Fazit

Die aktuellen XML-Technologien und ihre Implementaitionen erweisen sich als exzellentes Mittel zur Realisierung eines SSP-Systems, was hier, natürlich nur beispielhaft, umrissen wurde. Wegen der noch andauernden Weiterentwicklung sowohl des Standards als auch der XML-kompatiblen Software darf man also gespannt in die Zukunft blicken.

Anlage 1: Quellen

- [1] WWW-Consortium, XML Schema, <http://www.w3.org/XML/Schema>
- [2] WWW-Consortium, XML-SOAP, <http://www.w3.org/2002/ws/>
- [3] WWW-Consortium, <http://www.w3.org>
- [4] Norman Walsh, DocBook, <http://www.docbook.org/>

