Universität des Saarlandes Fachbereich 5.6 Informationswissenschaft
Seminar: “Elektronisches Publizieren am Beispiel von Friedrich Nietzsche"
Dozent: Prof. Dr. Harald H. Zimmermann

Thema der Hausarbeit:

Konzeption einer generellen Orte-Struktur mit exemplarischer Anwendung auf "Nietzsche-Online".

Von: Johannes Schwarz
e-Mail: schwarz.johannes@web.de

I Inhaltsverzeichnis

I Inhaltsverzeichnis: I
II Abkürzungsverzeichnis: II

1. Einleitung 1

2. XSL-Grundlagen 1
Stylesheets 2
Beispiel 1: 2
Beispiel 2: 4

3. Zusammenfassung: 9

III Literatur III

II Abkürzungsverzeichnis:

DTD:	Document Type Definition
HTML:	Hyper Text Markup Language
XML:	Extensible Markup Language
XSL:	Extensible Stylesheet Language
XSLT:	Extensible Style Language Transformation

1. Einleitung

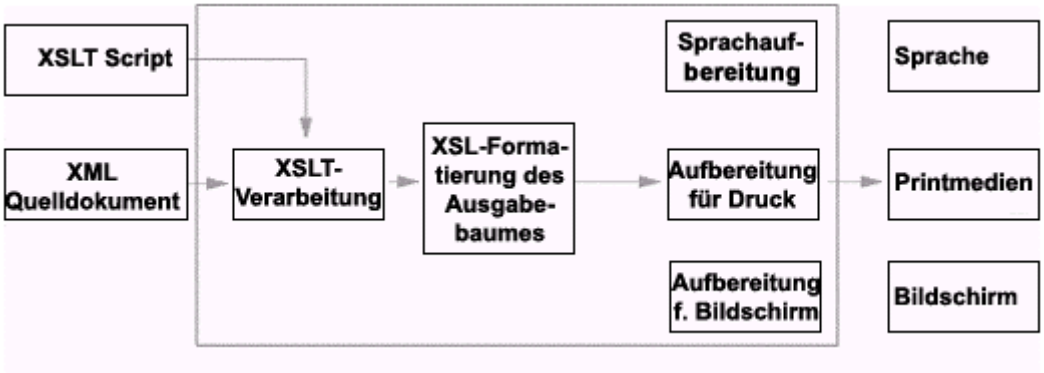
Anlass der hier zusammengestellten kurzen Einführung in XSL bildet das Seminar “Elektronisches Publizieren am Beispiel von Friedrich Nietzsche" mit dem Schwerpunkt Publikumsinformation, welches im Wintersemester 2000/01 am Fachbereich 5.6 Informationswissenschaft an der Universität des Saarlandes stattfand.

Für die Verwaltung und Präsentation von Datenmengen stellt die Kombination aus XML zur Verwaltung und XSLT zur Auswahl und Präsentation eine leistungsfähige Kombination dar. Am Beispiel einer Ortebeschreibung - welche eine Möglichkeit darstellt, sich mit dem Leben des Friedrich Nietzsche zu befassen - wird das Potential dieses Ansatzes aufgezeigt.

Anmerkung: Grundlagenwissen in XML und HTML wird vorausgesetzt!

2. XSL-Grundlagen

XSL stellt zwei Funktionen bereit: Neben dem Transformationsmechanismus verfügt XSL über einen Formatsprachschatz, wobei der bislang wichtigste Aufgabenbereich die Transformation darstellt. XSL definiert hierfür die Mechanismen, wie bestehende XML-Dokumente in neue Dokumente transformiert werden. Das transformierte Dokument kann dabei die Markierungen und die DTD des ursprünglichen Dokuments oder völlig andere Tag-Sets verwenden.



Die beiden XSL-Funktionen Transformation und Formatierung funktionieren unabhängig voneinander. So kann beispielsweise der Transformationsmechanismus

ein XML-Dokument in ein HTML-Dokument konvertieren und dabei das XSL-Formatobjekt völlig ignorieren - wie man es etwa vom Internet Explorer 5.x kennt.

Bei der Beschreibung von XSL unterscheidet man in der Regel drei Komponenten:

- XPath (XML Path Language) dient dazu, bestimmte Dokumententeile eines XML-Dokuments anzusprechen (Selektion).
- XSLT (XML Style Language Transformation) beschreibt, wie man die Baumdarstellung eines XML-Dokuments in eine andere Baumdarstellung verwandelt.
- XSL (eXtensible Style Language) XML-Sprachelemente zur Formatierung.

Eine Transformation, die mit XSLT ausgedrückt wird, beschreibt die Umwandlung eines Dokumentquellbaums (source tree) in einen Ergebnisbaum (result tree). Eine solche Transformation wird auch Stylesheet genannt. Bei einer XSL-Transformation liest der Prozessor sowohl das XML-Dokument als auch das zugehörige XSL-Stylesheet. Basierend auf den Informationen, die der Prozessor im XSL-Stylesheet findet, generiert er ein neues XML-Dokument oder Teile davon. Außerdem ist auch eine Umwandlung von XML in HTML möglich.

Stylesheets

Stylesheets sind eine Ansammlung von Vorlagenregeln (template rules), die jeweils aus zwei Teilen bestehen:

1) Muster (pattern)

2) Schablone (template)

Das Muster beschreibt, welche Quellknoten (Elemente) von der Regel verarbeitet werden sollen. Wenn Knoten gefunden werden, die mit dem Muster übereinstimmen, beschreibt die XML-Struktur die Schablone, die erzeugt werden soll.

Beispiel 1:

Das folgende Beispiel, welches aus dem Nietzscheprojekt entnommen ist, gibt einen Eindruck, wie XSL- Dokumente aufgebaut sind.

Ein XSL-Dokument beginnt immer mit der XML-Deklaration,

```
<?xml version="1.0"  encoding="iso-8859-1" standalone="yes"?>
```

gefolgt vom sogenannten Dokumentenelement mit der XSLT-Namensraumdeklaration.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/TR/REC-html40/loose.dtd">
```

Es folgen Top-Level-Elemente mit den eigentlichen Template-Regeln (template rules).

Template-Regeln werden mit dem Element xsl:template definiert. Sie legen fest, welche Knotentypen des Ausgangsdokuments den Transformationsvorgang durchlaufen. Es folgt die Definition des Musters (Pattern) über match. Match gibt an, auf welche Elemente eine Template-Regel angewendet werden soll; hier wird das Orteelement (direkt unter der Wurzel) transformiert.

```
<xsl:template match="/ort">
<xsl:call-template name="gesamt"/>
</xsl:template>
```

Im nächsten Abschnitt erfolgt über die XSLT-Instruktion xsl:apply-templates die Auswahl von Elementen aus dem Quelldokument. Diese Auswahl kann auch über xsl:for-each erfolgen. (Der Begriff "literale Ergebnisdokumente", der in diesem Zusammenhang oft gebraucht wird, beschreibt Elemente, die im Ergebnisdokument erzeugt werden.)

```
<xsl:template match="sprache">
   <i>
     <xsl:apply-templates/>
   </i>
</xsl:template>
```

Der Fett markierte Befehl xsl:value-of select, fügt an der aktuellen Stelle den 'Inhalt' der Knoten aus dem darauffolgenden Pfad ein.

```
xsl:template name="beschreibung">
     <br/>
     <xsl:apply-templates select="body/ortdaten/homepage"/>
     <br/>
     <br/>
     <b>Ortsbeschreibung
          <xsl:value-of select="/ort/body/ortdaten/ortsname"/>:
          </b>
        <xsl:apply-templates select="body/beschreibung"/>
```

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

In diesem letzten Beispiel kann man gut die bereits erwähnten Bestandteile im Zusammenspiel sehen.

Beispiel 2:

Das vorliegende XML-Orte-Template bildet die Grundlage der Ausprägungen der einzelnen XML-Ortedateien, also die Vorlage. Hier ein Ausschnitt:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE ort SYSTEM "ort.dtd">

<ort id="ort_ortname_00" typ="akt-sehenswert|akt-events|hist-politik|hist-kultur|sonstige">
 <head>
  <titel alpha="alphabeteintrag">Dokumentbenennung</titel>
  <autor id="per_geburtsname_vorname_00">
   <datum>
    <jahr>0000</jahr>
    <monat>00</monat>
    <tag>00</tag>
   </datum>
  </autor>
  <design id="per_geburtsname_vorname_00" typ="erstellung|modifikation|verschlagwortung">
   <datum>
    <jahr>0000</jahr>
    <monat>00</monat>
    <tag>00</tag>
   </datum>
  </design>
 </head>
 <body sprache="DE">
  <ortdaten>
   <ortsname>Ortsname</ortsname>
   <plz>00000</plz>
   <strasse>Strasse mit Nummer</strasse>
   <region>Region</region>
   <bundesland>Bundesland</bundesland>
   <land>Land</land>
...
```

Eine mögliche Ausprägung wird am Beispiel des Geburtsortes von Nietzsche - Röcken - aufgezeigt:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE ort SYSTEM "ort.dtd">

<?xml-stylesheet href="ort-html.xsl" type="text/xsl"?>
<?cocoon-process type="xslt"?>

<ort id="ort_roecken_00" typ="historisch">
 <head>
  <titel alpha="roecken"</titel>
  <autor id="per_schwarz_johannes_00">
   <datum>
    <jahr>2001</jahr>
    <monat>09</monat>
    <tag>01</tag>
   </datum>
  </autor>
  <design id="per_schwarz_johannes_00" typ="erstellung">
   <datum>
    <jahr>2001</jahr>
    <monat>08</monat>
    <tag>25</tag>
   </datum>
  </design>
 </head>
 <body sprache="DE">
  <ortdaten>
   <ortsname>Röcken</ortsname>
   <plz>06686</plz>
   <strasse>Strasse mit Nummer</strasse>
   <region>Kreis Weißenfels</region>
   <bundesland>Sachsen-Anhalt</bundesland>
   <land>Deutschland</land>
...
```

Die Fett markierten Bestandteile sind Röcken-spezifischen Daten.

Exemplarisch folgt die darstellende XSL-Datei (Auszug) für "Nietzsche-online":

```
<xsl:template name="html-body">
            <!-- Layout -->
<body>
<a name="top"></a>
<div class="nav">
 <div class="ignore">
  <table width="100%" border="0" cellspacing="0" cellpadding="0" bgcolor="#EEEDEA"
       bordercolor="#EEEDEA"> <tr>
    <td width="15%" bgcolor="#EEEDEX"> </td>
    <td width="0%" bgcolor="#EEEDEX"> </td>
    <td width="85%" bgcolor="#EEEDEX"> </td>
   </tr>
   <tr>
    <td width="15%">
     <div align="center"><a href="http://is.uni-sb.de"><img src="../../img/infowiss_logo.gif"
      width="200" height="80" border="0" alt="Informationswissenschaft"/></a></div>
    </td>
    <td width="0%" bgcolor="#EEEDEX"> </td>
    <td width="85%">
     <table width="100%" border="0" cellspacing="0" cellpadding="0">
      <tr>
       <td width="8%">
        <div align="center"><a href="../../index.html"><img src="../../img/nietzsche_index_left.gif"
         width="67" height="81" border="0" alt="Home"/></a></div>
       </td>
       <td width="84%">
        <div align="center"><img src="../../img/nietzsche_online_schatten.gif" width="273" height="69"/>
         </div></td>
       <td width="8%">
        <div align="center"><a href="../../index.html"><img src="../../img/nietzsche_index_right.gif"
         width="67" height="81" border="0" alt="Home"/></a></div>
       </td> </tr></table></td></tr>
   <tr>
    <td width="15%" bgcolor="#EEEDEX" valign="top"> </td>
    <td width="0%" bgcolor="#EEEDEX"> </td>
    <td width="85%" bgcolor="#EEEDEX">  </td>
   </tr>
   <tr>
    <td width="15%" valign="top" bgcolor="#DBDEDF">
     <table width="100%" border="0" cellspacing="0" cellpadding="0">
      <tr>
       <td bgcolor="#EEEDEA" valign="top">
        <table width="100%" border="0" cellspacing="2" cellpadding="2" bgcolor="#EEEDEA">
         <tr>
          <td bgcolor="#DBDEDF"><a href="../index.html"><font face="Verdana, Arial, Helvetica,
           sans-serif" size="2">Home</font></a></td></tr>
…
```

Vergleichend kann die XSL-Datei des "Elsa" -Projekts angeführt werden, dem die gleiche XML-Datei zugrunde liegt:

```
<xsl:template name="html-body">
  <body>
        <!-- Layout -->
        <center>
   <table width="800pt" bgcolor="#F7F4E5" cellpadding="2" border="0">
   <tr><td colspan="11">
     <table width="100%" bgcolor="#E7C385" bordercolor="#EEEDEA" cellpadding="5">
     <tr><td width="10%"><img src="w_pferd_klein03.jpg" width="73" height="63" border="0"
         alt="weisses Pferd"/> </td>
       <td width="90%" align="center" valign="middle" bgcolor="#DFD6AD"><br/>
         <i><font size="+4">ELSA</font>
           <font size="+2"> Elektronisches Literaturarchiv</font>
           <font size="+1"> Saar-Lor-Lux-Elsass</font>
                                    <br/> 
         </i></td><td> </td></tr></table></td></tr>
   <tr><td width="10pt" align="center" bgcolor="#C79268"> </td>
     <td align="center" bgcolor="#C79268"><a  href='../../../inhalt.php'>home</a> </td>
     <td align="center" bgcolor="#C79268"><a  href='../../../projekt.php'>projekt</a> </td>
          <td align="center" bgcolor="#C79268"><a  href='../../../suche.php'>suche</a> </td>
          <td align="center" bgcolor="#C79268"><a  href='../../../vorlass.php'>vorlass</a> </td>
          <td align="center" bgcolor="#C79268"><a  href='../../../technik.php'>technik</a> </td>
     <td width="10pt" align="center" bgcolor="#C79268"> </td>
    </tr>
  <tr><td colspan="11" bgcolor="#F3E9DF">
     <table border="0" width="99%" bordercolor="#EEEDEA" bgcolor="#F7F1F3" cellpadding="10">
     <tr><td width="100%">
   <!-- Sichtbarer Inhalt der Briefe -->
   <xsl:call-template name="body"/></td> </tr></table>
      
          </td>
         </tr>
  <tr><td colspan="11" align="right"><small><a href="mailto: d.luckhardt@is.uni-sb.de">Dr. Heinz-Dirk
    Luckhardt</a>, <a href="http://is.uni-sb.de">Informationswissenschaft</a><br/>
<script language="JavaScript">
                        <!--document.write(document.lastModified);// -->
                       </script> </small>
                       </td></tr>
 <tr><td width="10pt" align="center" bgcolor="#C79268"> </td>
   <td align="center" bgcolor="#C79268"><a  href='../../../inhalt.php'>home</a> </td>
          <td align="center" bgcolor="#C79268"><a  href='../../../projekt.php'>projekt</a> </td>
          <td align="center" bgcolor="#C79268"><a  href='../../../suche.php'>suche</a> </td>
          <td align="center" bgcolor="#C79268"><a  href='../../../vorlass.php'>vorlass</a> </td>
          <td align="center" bgcolor="#C79268"><a  href='../../../technik.php'>technik</a> </td>
   <td width="10pt" align="center" bgcolor="#C79268"> </td>
 </tr>
 </table>
 <table width="90%" bordercolor="#ffffff">
 <tr><td align="right"> </td></tr>
 <tr><td> </td></tr></table>
 </center>
  </body>
 </xsl:template>
```

(Kenntnisse über den hier zur Formatierung eingesetzten HTML-Code wird vorausgesetzt und im Rahmen dieser Arbeit nicht näher betrachtet.)

Schon beim bloßen überfliegen wird deutlich, dass sich die grafischen Ausgaben der beiden Dateien stark unterscheiden. Die nachfolgenden Grafiken verdeutlichen die unterschiedlichen Präsentationsformen:

Abbildung 1: Projekt "Nietzsche-Online"- Beispiel einer möglichen Ausgabe der XML-Ortedatei
(Internetquelle (10.09.01): http://bigmac.phil.uni-sb.de:82/ort/xsl/ort_leipzig_00.xml)

Abbildung 2: Projekt Elsa - Weiteres Beispiel einer möglichen Ausgabe der XML-Ortedatei
(Internetquelle(10.09.01): http://bigmac.phil.uni-sb.de:81/gulden/briefe/xml/ort_saarlouis_00.xml)

3. Zusammenfassung:

Der größte Vorteil von XSL besteht darin, dass XML-Dokumente durch verschiedene Darstellungsdateien (XSL-Dateien) ausgedrückt werden können, in welchen unterschiedliche Möglichkeiten der Präsentation beschrieben werden (z.B. verschiedene Internetseiten für Geschäfts- und Privatkunden). Dies wurde anhand der beiden Projekte, Elsa und Nietzsche, veranschaulicht. Der Nachteil von XSL - bzw. XML im allgemeinen - ist, dass sich durch die Trennung von Inhalt, Struktur und Präsentation der Umfang im Vergleich zu Binärdateien aufbläht. Diesem Nachteil kann die Plattformunabhängigkeit entgegengestellt werden, die in einer Zeit, in der stetig größere Datenmengen zwischen Universitäten, Unternehmen oder Privatpersonen transferiert werden, eine immer größere Rolle spielt.

.

III Literatur

Bach, Mike: XSL und XPath - verständlich und praxisnah. Transformation und Ausgabe von XML- Dokumenten mit XSL, Addison-Wesley, September 2000, ISBN: 3827316618