

# Fallstudie E-Commerce

Thema : Webseiten mit XML

Oliver Litz – xxxxxxxx –  
Stefan Immich – xxxxxxxx –

Datum : 02.10.2000

Fach E-Commerce, Prof. Dr. Klaus Huckert  
Fachbereich Grundlagen, Informatik, Sensortechnik ( GIS )  
Studiengang Praktische Informatik

**Hochschule für Technik und Wirtschaft des Saarlandes**

## Inhaltsverzeichnis

1. Einleitung .....	1
2. XML-Interpretation im Web .....	2
2.1 Serverseitige Verarbeitung .....	2
2.2 Clientseitige Verarbeitung .....	3
3. XML-Unterstützung aktueller Webbrowser .....	5
3.1 Microsoft Internet Explorer 5.x .....	5
3.2 Netscape Navigator 6 .....	5
4. Aufbau .....	6
5. Entwicklung und Gestaltung .....	8
6. Templates .....	9
6.1 Standard-Template .....	9
6.2 Titel-Template .....	9
6.3 Versandhandel-Template .....	10
6.4 Leistungen-Template .....	11
7. Suchfunktionalität .....	12
8. Problematik .....	13
9. Bewertung .....	14
Abbildungsverzeichnis .....	15
Literaturverzeichnis .....	16

## 1. Einleitung

Die Extensible Markup Language, kurz XML, wird in naher Zukunft sicher in viele IT-Anwendungsbereiche Einzug halten. Die Einsatzgebiete dieser Beschreibungssprache sind sehr vielfältig und werden wohl auch im Laufe der Zeit ältere, unflexiblere oder auch kostenintensive Technologien verdrängen. Aber besonders im Internet wird XML die Aufgabe zugeschrieben, die Präsentation und den Austausch von Daten mit einheitlichen, nicht-proprietären Formaten für die Anwender und Entwickler zu vereinfachen. [Bünthe, Seite 200] schreibt hierzu : "Für das Web markiert XML hoffentlich das Ende der Flickschusterei, bei der jeder Browser-Hersteller HTML durch eigene Erweiterungen verschlimmbessert, die sich dann doch wieder nicht vernünftig ausdrücken lassen."

Die Aufgabe der vorliegenden Ausarbeitung soll es zum einen sein, herauszufinden auf welche Art sich XML für die Entwicklung im Web nutzen lässt. Desweiteren soll ein konkreter "virtueller" Webauftritt mit den bisher verfügbaren XML-Standards und Entwicklungswerkzeugen realisiert werden. Abschließend werden noch Realisierungs- und Anwendungsprobleme erläutert, die bei dem derzeitigen Entwicklungsstand der Extensible Markup Language zu erwarten sind.

Diese Ausarbeitung richtet sich an alle ( angehenden ) Webentwickler, die zwar die Grundzüge der Extensible Markup Language bereits verstanden haben, aber noch keine Erfahrung in der Verwendung von XML für Internetseiten besitzen.

## 2. XML-Interpretation im Web

Ein XML Dokument kann auf verschiedene Arten auf dem heimischen Browser dargestellt werden. Der Ausgangspunkt ist in jedem Falle ein XML-Dokument das sich auf einem Server befindet. Hierbei unterscheidet man zwischen statischen XML-Dokumenten und dynamischen. Letztere können aus einer Clientanfrage heraus dynamisch generiert werden.

Soll nun ein XML-Dokument über das Internet beispielsweise in einem Browser dargestellt werden, gibt es hierzu zwei grundlegende Strategien. Einerseits ist es möglich das Dokument bereits auf dem Server so zu wandeln, das der Browser lediglich ein für ihn verständliches HTML-Dokument erhält. Desweiteren besteht noch die Möglichkeit ein XML-Dokument unmittelbar an den Client zu übertragen. Dieser muss aber dann bereits XML unterstützen um die Seite korrekt anzeigen zu können. Nachfolgend wollen wir näher auf diese beiden Verarbeitungsmöglichkeiten eingehen.

### 2.1 Serverseitige Verarbeitung

Für den Benutzer ist es natürlich die einfachste Art, wenn die Transformation von XML nach HTML auf dem Webserver stattfindet. Es ergeben sich für ihn dadurch keine Änderungen zur heutigen Praxis, da er immer noch wie bisher HTML-Seiten empfängt, die jeder Webbrowser darstellen kann. Natürlich ist diese Vorgehensweise für denjenigen der die Seiten im Netz bereitstellt etwas aufwendiger, bietet aber auch gewisse Vorteile.

Das angeforderte XML-Dokument das die eigentlichen Daten enthält, wird auf Anfrage mit einem sogenannten Stylesheet verknüpft. Solche Stylesheets sind bereits aus der HTML-Welt bekannt. Sie werden dort als Cascading Stylesheets ( CSS ) bezeichnet und haben ähnliche Funktionalität.

Die Stylesheets in der XML-Welt tragen gewöhnlich die Bezeichnung Extensible Style Sheet ( XSL ). Diese enthalten die notwendigen Formatanweisungen um die unformatierten Rohdaten der XML-Datei zu einer Ausgabe zu transformieren.

Eine solche Transformation, die häufig zusätzlicher Software bedarf, wird auch als XSLT bezeichnet, die Daten einer bestimmten DTD in eine andere überführen kann. Im Moment wird hierüber hauptsächlich die Konvertierung in HTML Seiten vorgenommen. Aber auch andere Transformationen etwa nach PDF sind möglich. Selbstverständlich erfordert eine solche XSLT zusätzliche Dokumentenverarbeitung welche bei großer, gleichzeitiger Anzahl von Zugriffen die Serverlast erhöht.

Trotz des hier angesprochenen Nachteils gibt es auch Vorteile gegenüber der klassischen HTML-Lösung. Ein nicht unwichtiger Vorteil ist die Nutzung mehrerer spezieller XSL-Stylesheets auf dem Server. Ein Dokument läßt sich nun auf jedem Browser oder Client darstellen für den ein passendes Stylesheet verfügbar ist. Somit lassen sich die gleichen XML-Daten für einen grafischen Webbrowser wie den Internet Explorer bzw. Netscape Navigator genauso aufbereiten wie für die Darstellung im textbasierten Lynx-Browser.

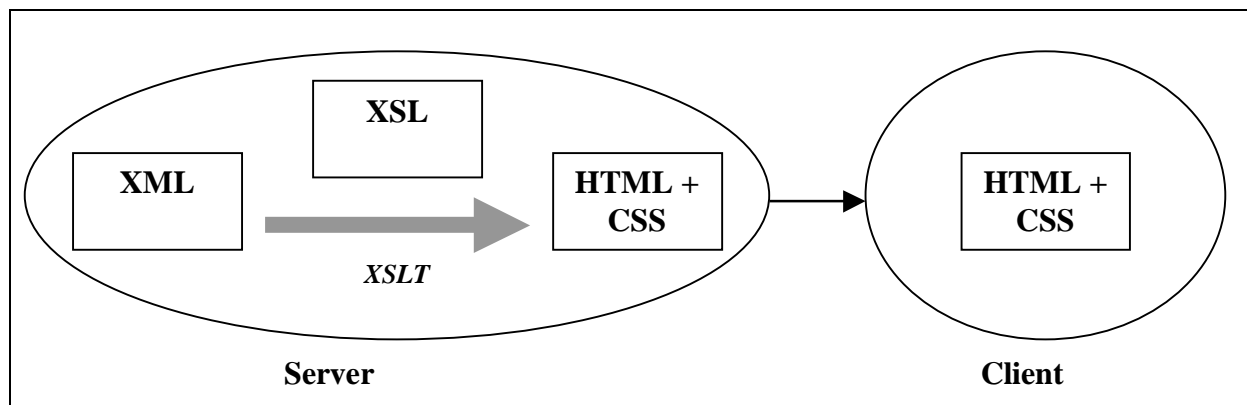


Abbildung 1: Serverseitige XML-Verarbeitung

## 2.2 Clientseitige Verarbeitung

Die zweite Möglichkeit XML-Seiten in einem Webbrowser darzustellen ist die sogenannte clientseitige Interpretation. Im Gegensatz zu der serverseitigen Transformation nach HTML wird hier die eigentliche XML-Datei normalerweise mit einem passenden XSL-Stylesheet zum Browser übertragen. Dieser ist dann für die korrekte Darstellung verantwortlich.

Eine Voraussetzung hierfür ist die "Wohlgeformtheit" des XML-Dokuments, das dann auch ohne zugehörige DTD auskommt. Geschwindigkeitsmäßig bedeutet diese Strategie eine Entlastung für den Server, da dieser lediglich die XML und XSL Dateien übertragen muss. Im direkten Vergleich zur reinen HTML-Verarbeitung nimmt die Last also nicht sonderlich zu, da auch hier eine HTML und oft auch eine CSS-Datei übertragen wird.

Problematisch ist diese Lösung im Moment dennoch, da die XML-Unterstützung in derzeitigen Webbrowsern noch sehr gering und meist nicht aktuell ist. Auf dieses Problem wird im nächsten Kapitel "XML-Unterstützung aktueller Webbrowser" sowie im Kapitel "Problematik" noch näher eingegangen.

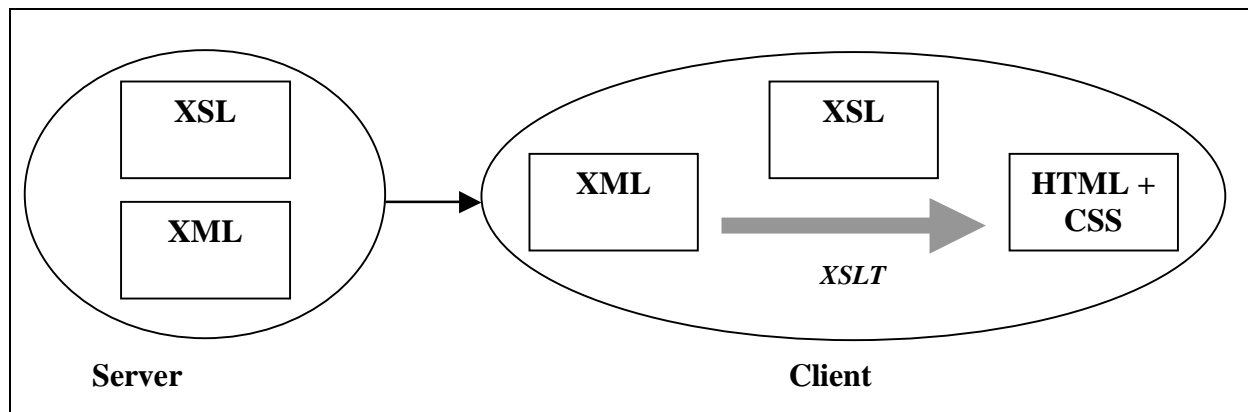


Abbildung 2: Clientseitige XML-Verarbeitung

### **3. XML-Unterstützung aktueller Webbrowser**

Der Vorteil von XML-Dokumenten ist, dass sich Inhalt und Darstellung voneinander trennen lassen. Dabei sollte sich der Inhalt im strukturierten XML-Dokument befinden und die Beschreibung der Darstellung im Stylesheet. Dieses kann ein CSS-Stylesheet sein, wie es von HTML bekannt ist, oder auch ein XSL-Stylesheet. Letzteres ist vorzuziehen, da XSL für die Darstellung von XML-Dokumenten entwickelt wurde und viel besser auf die Möglichkeiten von XML angepasst ist. Ein Webbrowser mit XML-Unterstützung muss also die XML-Dokumente verarbeiten können und wenigstens CSS-Stylesheets zur Darstellung unterstützen. Die verbreitetsten Webbrowser sind im Moment der Microsoft Internet Explorer und der Netscape Navigator. Eine Internetseite sollte auf diesen Browsern darstellbar sein.

#### **3.1 Microsoft Internet Explorer 5.x**

Der Internet Explorer unterstützt ab der Version 5.0 XML-Dokumente, XSL-Stylesheets und CSS-Stylesheets. Microsoft ist bei der Entwicklung der entsprechenden Parser im Vergleich zur Konkurrenz weit fortgeschritten und bietet damit die umfangreichste XML-Unterstützung. XSL-Dokumente werden von dem MSXML-Parser verarbeitet. Dieser ist im Internet Explorer enthalten, bietet aber bisher nur eine rudimentäre XSL-Unterstützung (siehe auch Kapitel 9). Ein XML-Dokument ohne Stylesheet wird als übersichtliche Baumstruktur dargestellt. Da der Internet Explorer 5.x im Betriebssystem seit Windows98 enthalten ist, gibt es für XML auf der Clientseite schon eine breite Basis.

#### **3.2 Netscape Navigator 6**

Der Netscape Navigator unterstützt erst ab der Version 6 XML-Dokumente. Diese befindet sich im Moment noch in der Entwicklung und ist sehr instabil. Zur Darstellung der XML-Dokumente können bisher nur CSS-Stylesheets verwendet werden. Ein XSL-Parser ist noch nicht vorhanden. Dadurch ist man bei der Präsentation der Dokumente sehr eingeschränkt. Ein XML-Dokument ohne Stylesheet wird als unübersichtlicher, fortlaufender Text dargestellt. Die Version 4.x ist für XML nicht geeignet.

## 4. Aufbau

Als Beispiel einer Internetseite dient der virtuelle Versandhandel "Virtual Computer Trading GmbH", der seine Produkte und Dienstleistungen im Web präsentiert. Die Struktur der Seiten ist mit XML realisiert. Zur Darstellung wird XSL verwendet. Daher empfehlen wir die Verwendung des Microsoft Internet Explorers ab Version 5. Die Dateien werden vom Server an den Client-Browser übertragen und von diesem aufbereitet und dargestellt. Für die Bestellaufnahme werden Daten als Email an den Server geschickt. Die Dokumente sind auf der beigelegten Diskette verfügbar.

Der Ausgangspunkt des Internetauftritts ist die Startseite. Der Benutzer kann von hier aus auf verschiedene Bereiche zugreifen. Die Bereiche lassen sich wie folgt untergliedern:

- Produktseite mit einem Überblick auf Produkte und Preise sowie einer Suchmöglichkeit
- Bestellseite mit einem Formular zur Eingabe der persönlichen Daten und der Bestellung
- Seite mit den allgemeinen Geschäftsbedingungen des Unternehmens
- Kontaktseite mit Informationen zum Unternehmen

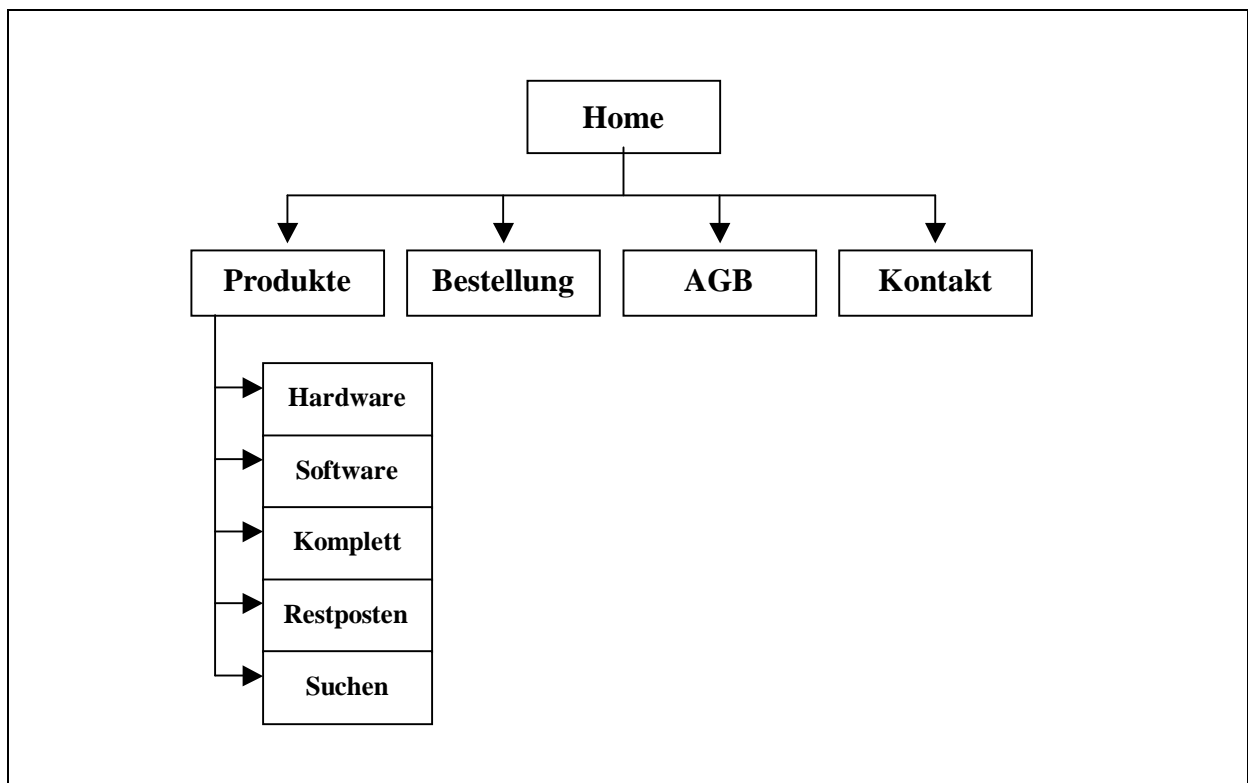
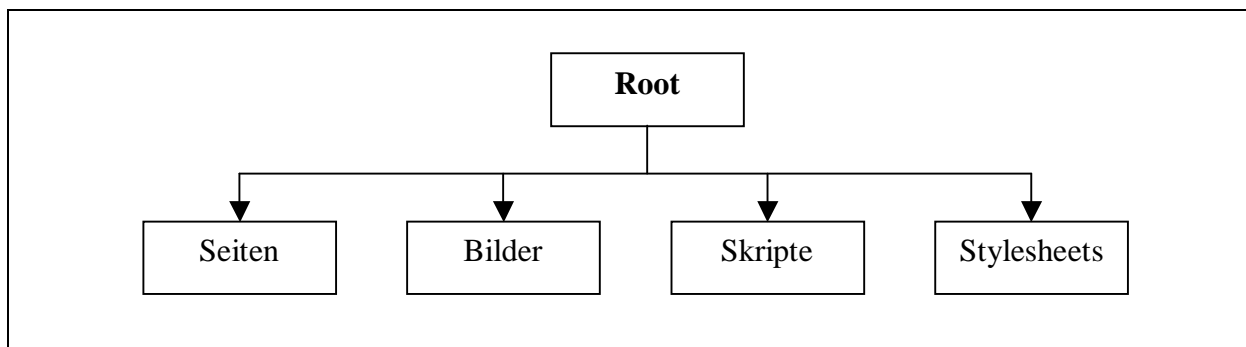


Abbildung 3: Strukturübersicht des Versandhandels



Die in der Abbildung 3 dargestellte Strukturübersicht beinhaltet die rein logische Anordnung der einzelnen Bereiche ausgehend von der Startseite. Ein weiterer Strukturierungsaspekt ist die physikalische Aufteilung der Internetseiten und deren Elemente wie Skripte, Bilder und Stylesheets auf der Festplatte. Eine derartige Aufteilung sollte auf jedes Projekt angewandt werden. Dabei ist es unwichtig ob nun HTML-Seiten oder XML-Seiten strukturiert werden sollen. Die folgende Abbildung verdeutlicht das prinzipielle Vorgehen.



**Abbildung 4: Physikalische Struktur des Versandhandels**

Bis auf das Verzeichnis "Seiten", das für die XML- bzw. HTML-Dokumente vorgesehen ist, realisierten wir unseren virtuellen Webauftritt wie in Abbildung 4 dargestellt. Die XML-Seiten selbst lagerten wir direkt in das Rootverzeichnis aus. Nachdem wir die logische/physikalische Struktur erläutert haben, folgt nun die konkrete Entwicklung der XML/XSL-Dateien.

## 5. Entwicklung und Gestaltung

Bei der Entwicklung haben wir nach dem Grundsatz von XML den Inhalt und die Darstellung getrennt. Die XML-Dokumente enthalten die strukturierten Daten für die Internetseite und sind in den Dateien mit der Endung ".xml" gespeichert. Dabei gibt es für jeden Bereich eine eigene XML-Datei, wie z.B. "produkte.xml". So bleiben die Dateien übersichtlich und sind einfach zu warten.

Jede XML-Datei hat folgenden Aufbau:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<?xml-stylesheet href="versandhandel.xsl" type="text/xsl" ?>
<XmlSeite>
  <Titel>
    <Text>Überschrift der Seite</Text>
    <Bild name="bild.gif"/>
  </Titel>

  Beschreibende Tags wie z.B.: <Leistungen>...</Leistungen>
                             oder <Produkte>...</Produkte>
</XmlSeite>
```

**Abbildung 5: Aufbau der XML-Dokumente**

Die Beschreibung der Darstellung ist in der XSL-Datei "versandhandel.xsl" enthalten. Hierin erfolgt eine Transformation nach HTML. Die Datei bietet ein Grundgerüst einer HTML-Datei. In dieser ist auch die Kopfzeile realisiert, die dadurch auf jeder Seite dargestellt wird. Alle XML-Dateien benutzen dieses eine XSL-Stylesheet zur Darstellung, das sich im Verzeichnis "Stylesheets" befindet.

Die HTML-Elemente werden anhand des Cascading-Stylesheets "versandhandel.css" formatiert. Hierin werden Schriftgröße, Schriftart, Abstand, Rand, usw. festgelegt. Folgender Eintrag formatiert z.B. das Element Titelüberschrift:

```
.Titelueberschrift
{
  font-size:26pt;
  font-weight:bold;
  text-align:center;
  text-decoration:underline;
}
```

**Abbildung 6: CSS-Style-Angabe**

## 6. Templates

Das Stylesheet enthält mehrere Templates, die eine Darstellung der jeweiligen XML-Tags beschreiben. Ein XSL-Stylesheet kann für jedes - im XML-Dokument vorkommende - Tag eine Formatvorlage enthalten, die zur Ausgabe der Tag-Struktur dient.

### 6.1 Standard-Template

Im Standard-Template wird ein Grundgerüst für die HTML-Ausgabe erstellt. Im Bereich "<BODY>" werden dann die Templates angewendet.

```
<xsl:template match="/">
<link rel="stylesheet" type="text/css" href="versandhandel.css"/>
<HTML>
<HEAD>
  <TITLE>
    <xsl:value-of select="XmlSeite/Titel/Text"/>
  </TITLE>
</HEAD>
<BODY>
  <TABLE>...Kopfzeile...</TABLE>
  <xsl:for-each select="XmlSeite">
    <SPAN>
      <xsl:apply-templates/>
    </SPAN>
  </xsl:for-each>
</BODY>
</HTML>
</xsl:template>
```

Abbildung 7: XSL-Template zur HTML-Darstellung

### 6.2 Titel-Template

Durch das Titel-Template wird die Darstellung einer Überschrift für jede Seite erreicht. Bei Vorhandensein des Bild-Tags ist es auch möglich ein angegebenes Bild unter der Überschrift darzustellen. Verwendet wird das Titel-Template durch den Gebrauch des Tags "<Titel>" im XML-Dokument.

```

<xsl:template match="Titel">
<DIV CLASS="Titelueberschrift">
  <xsl:value-of select="Text"/>
</DIV>
<xsl:choose>
  <xsl:when match=".[Bild]">
    <DIV CLASS="Titelbild">
      <IMG>
        <xsl:attribute name="SRC">
          pics/<xsl:value-of select="Bild/@name"/>
        </xsl:attribute>
      </IMG>
    </DIV>
  </xsl:when>
</xsl:choose>
</xsl:template>

```

Abbildung 8: XSL-Titel-Template

### 6.3 Versandhandel-Template

Durch dieses Template ist es möglich einen ganzen Datenbestand in einer Tabelle grafisch zu präsentieren. Dabei werden die Inhalte sortiert und eventuelle Sonderpreise farbig hervorgehoben. Der Datenbestand kann aus einer Datenbank durch einen einfach zu erstellenden Filter nach XML exportiert werden. Die XML-Datei hat dabei den Aufbau, wie z.B. aus "hardware.xml" ersichtlich:

```

<Versandhandel>
  <Kategorie>Hardware</Kategorie>
  <Produkt>
    <Typ>Grafikkarte</Typ>
    <Bezeichnung>Velocity 4400</Bezeichnung>
    <Hersteller>STB</Hersteller>
    <Preis typ="Sonderpreis">199</Preis>
  </Produkt>
  ...
</Versandhandel>

```

Abbildung 9: Datei "hardware.xml"

Das entsprechende Versandhandel-Template bereitet die Daten durch folgende Schleife in einer Tabelle auf: `<xsl:for-each select="Produkt">...</xsl:for-each>`

Die rote Darstellung des Sonderpreises wird durch folgende Abfrage erreicht:

```
<xsl:when match=".[Preis/@typ='Sonderpreis']">
  <TD STYLE="color:red; text-align:right">
    <xsl:value-of select="Preis"/>,- DM
  </TD>
</xsl:when>
```

**Abbildung 10: Darstellung des Sonderpreises**

## 6.4 Leistungen-Template

Hyperlinks werden mit einfachen Xlinks erzeugt. Dazu ist es notwendig dem XML-Tag einen Verweis im Attribut "href" mitzugeben, wie z.B.

```
<Link xml:link="simple" href="agb.xml">AGB</Link>
```

Im XSL-Template wird das Attribut "HREF" des HTML-Tags `<A>` erzeugt. Dazu wird aus dem XML-Tag "`<Link>`" der Wert des Attributes "href" entnommen.

```
<xsl:template match="Leistungen">
...
  <xsl:for-each select="Leistung">
    <LI>
      <A>
        <xsl:attribute name="HREF">
          <xsl:value-of select="Link/@href"/>
        </xsl:attribute>
        <DIV CLASS="Liste"><xsl:value-of select="Link"/></DIV>
      </A>
    </LI>
  </xsl:for-each>
...
</xsl:template>
```

**Abbildung 11: Leistungen-Template**

## 7. Suchfunktionalität

Durch die Einbindung eines JavaScript-Programmes kann eine Suche auf die Daten der Produktpalette erfolgen. Das JavaScript-Programm benutzt XML-Dateninseln innerhalb einer HTML-Seite um auf die XML-Dokumente zugreifen zu können. XML-Dateninseln werden beim Parsen der HTML-Dateien vom Browser nicht beachtet. Eine Dateninsel ist laut [Seeboerger-Weichselbaum, Seite 296] : "Alles, was sich zwischen <XML> und </XML> befindet, ist eine so genannte Dateninsel ('Data Island')".

```
<HTML>
<HEAD>
<TITLE>Suchergebnisse Versandhandel VCT</TITLE>
<SCRIPT LANGUAGE="JavaScript" SRC="script/code.js">
</SCRIPT>
</HEAD>
<BODY onload="abfrage()">
<XML ID="Insel1" SRC="hardware.xml">
</XML>
<XML ID="Insel2" SRC="software.xml">
</XML>
<XML ID="Insel3" SRC="komplettsysteme.xml">
</XML>
<XML ID="Insel4" SRC="restposten.xml">
</XML>
</BODY>
</HTML>
```

**Abbildung 12: XML-Dateninseln**

Das JavaScript durchläuft die einzelnen Knoten der XML-Dokumente und überprüft deren Inhalt auf Übereinstimmung mit dem Suchbegriff. Wird der Suchbegriff gefunden öffnet sich ein Fenster mit dem Ergebnis der Suche. Das JavaScript "suche.js" haben wir aus [Seeboerger-Weichselbaum, Seite 306ff.] entnommen und den Erfordernissen unseres Versandhandels angepasst.

## 8. Problematik

In aktuellen Webbrowsern ist eine XML-Unterstützung bisher nur im Microsoft Internet Explorer ab Version 5 vorhanden. Dadurch können sich Benutzer anderer Webbrowser die erstellten Seiten nicht ansehen. Für die Darstellung im Web ist daher vorerst noch eine Wandlung der Dokumente nach HTML auf der Serverseite vorzuziehen. Erst wenn auch die noch in der Entwicklung befindliche Version 6 des Netscape Navigators XML voll unterstützt, kann die Wandlung der Dokumente nach HTML direkt auf dem Client durch den Webbrowser erfolgen.

Der Microsoft Internet Explorer beachtet einige Vorgaben der DTD nicht. Benennt man z.B. die DTD mit "Adresse" (z.B. `<!DOCTYPE Adresse SYSTEM "daten.dtd">`), muss auch der Elementtyp des Wurzel-Elements "Adresse" heißen. Der Internet Explorer lässt aber auch ein Wurzel-Element "Anders" zu, obwohl dies nicht dem XML-Standard entspricht.

Im Internet Explorer werden XSL-Befehle vom MSXML-Parser verarbeitet. Dieser basiert aber auf einem alten XSL-Draft, wodurch viele Befehle nicht zur Verfügung stehen, wie z.B. `<xsl:import...>` oder `<xsl:sort...>`.

Dadurch entsteht ein weiteres Problem. Der MSXML-Parser stellt stattdessen proprietäre Erweiterungen zur Verfügung. Zum Sortieren von Datenbeständen ist unter XSL der Befehl `<xsl:sort ...>` verfügbar. Der Internet Explorer kennt diesen Befehl nicht, der dem XML-Standard entspricht. Stattdessen ist "order-by" zu verwenden, wie z.B. `<xsl:for-each ... order-by="+ Name">`. Sollen also Daten sortiert werden, kann man beim Internet Explorer nur das proprietäre "order-by" verwenden. Für andere Parser ist allerdings "sort" zwingend notwendig.

## 9. Bewertung

Anhand des simulierten Webauftritts der "Virtual Computer Trading GmbH" wurde anschaulich gezeigt, dass es unter Einsatz von XML nicht unbedingt schwerer ist eine gute und funktionsfähige Internetseite zu erstellen. Sicherlich muss die Auszeichnungssprache XML und die Formatsprache XSL erst erlernt werden, doch ist es für gute Entwickler im Bereich Internet immer Voraussetzung sich neue Technologien frühzeitig anzueignen. Falls die Sprachen dann einmal beherrscht werden, findet man schnell heraus, welche Vorteile gegenüber HTML dafür sprechen.

Die strikte Trennung von Daten und ihrer Präsentation erlaubt eine einfache und sehr flexible Verarbeitung. Die Vorteile dieses Prinzips beziehen sich nicht allein auf Webseiten, sie betreffen vielmehr den gesamten Datenworkflow des Unternehmens. So lässt sich mit Hilfe von XML eine Datenbank für alle erdenklichen Zwecke nutzen. [Holzer/Knöfel/Siegling, Seite 221] sind nach Abschluss eines Projektes mit XML zum gleichen Schluss gekommen : "Die Entwickler [...] haben nach Abschluss der Entwicklungsarbeiten [...] ihre Erfahrung zum Ausdruck gebracht, dass XML ihren Entwicklungsaufwand gegenüber vergleichbaren Projekten gesenkt und die Flexibilität erhöht hat.". Vorstellbar wären beispielsweise einfache Filter die aus Datenabfragen XML-Dokumente erstellen. Mit Hilfe von XSL-Transformationen könnten diese entweder für eine Webseite, eine Präsentation oder für die Auftragsbearbeitung verwendet werden.

Die weitere Entwicklung von XML im Internet wird stark von den erhältlichen Entwicklungswerkzeugen und der Unterstützung gängiger Browser abhängen. Ebenso befindet sich XML/XSL als zukünftiger Standard immer noch in der Weiterentwicklung. Die vorgestellten Browser Netscape 6.0 und Internet Explorer 5.5, die beide den Anwendermarkt dominieren, werden wahrscheinlich in naher Zukunft XML, XSL sowie DTDs vollständig unterstützen. Dies ist wichtig für einen professionellen Einsatz der neuen Technologien.



## Abbildungsverzeichnis

Abbildung 1: Serverseitige XML-Verarbeitung.....	3
Abbildung 2: Clientseitige XML-Verarbeitung .....	4
Abbildung 3: Strukturübersicht des Versandhandels.....	6
Abbildung 4: Physikalische Struktur des Versandhandels.....	7
Abbildung 5: Aufbau der XML-Dokumente.....	8
Abbildung 6: CSS-Style-Angabe.....	8
Abbildung 7: XSL-Template zur HTML-Darstellung .....	9
Abbildung 8: XSL-Titel-Template .....	10
Abbildung 9: Datei "hardware.xml" .....	10
Abbildung 10: Darstellung des Sonderpreises .....	11
Abbildung 11: Leistungen-Template .....	11
Abbildung 12: XML-Dateninseln.....	12

## Literaturverzeichnis

- [Seeboerger-Wechselbaum] Michael Seeboerger-Wechselbaum : XML – Das Einsteigerseminar, 2. überarbeitete Auflage, bhv-Verlag 2000
- [Floyd] Michael Floyd : Building Web Sites with XML, Prentice Hall 2000
- [Bünthe] Oliver Bünthe : XML auf dem Vormarsch, in c't – Magazin für Computertechnik, Heft 10/2000, Seite 200 - 213
- [Holzer/Knöfel/Siegling] Thomas Holzer, Sabine Knöfel, Ilka Siegling : Mit XML um die Welt, in c't – Magazin für Computertechnik, Heft 10/2000, Seite 214 – 221
- [Oberdörster] Alexander Oberdörster : XML mit Serviervorschlag, in c't – Magazin für Computertechnik, Heft 6/2000, Seite 244 – 249
- [Behme/Mintert] Henning Behme, Stefan Mintert : XML in der Praxis, 2. Auflage, Addison Wesley 2000