

HITWK

Hochschule für Technik,
Wirtschaft und Kultur Leipzig

ABSCHLUSSPROJEKT INTERNETTECHNOLOGIEN

Transformation einer XML-Datei

Emanuel ECKSTEIN

Matrikelnummer: 85117

Sebastian STIER

Matrikelnummer: 75156

29. April 2024

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
1 Einleitung	1
1.1 Einleitung	1
1.2 Aufgabenstellung	1
2 Dateiformate und ihre Strukturen	3
2.1 Umsetzung des XML Files	3
2.2 Umsetzung des XSD Files	4
2.3 Umsetzung des XSL Files	6
3 Transformation und Ausgabe	9
3.1 Erzeugung der HTML/PDF	9
3.2 Umsetzung in HTML	9
3.3 Umsetzung in PDF	10
4 Zusammenfassung	12
4.1 Zusammenfassung	12
4.2 Ausblick	12
4.3 Verwendetes Bild- und Informationsmaterial der Saatguttütchen	13

Abkürzungsverzeichnis

FOP	Apache Formatting Objects Processor
HTML	Hypertext Markup Language
HTWK	Hochschule für Technik, Wirtschaft und Kultur Leipzig
ID	Identifikationsnummer
PDF	Portable Document Format
UTF	Unicode Transformation Format
XML	eXtensible Markup Language
XSD	XML Schema Definition
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformations

Abbildungsverzeichnis

2.1	Schematischer Aufbau	5
2.2	Aufbau der Tabelle	8
3.1	struktureller Aufbau der Transformation	9
3.2	Einblick in die generierte HTML	10

Tabellenverzeichnis

2.1	Elemente XML-Struktur	3
-----	---------------------------------	---

1 Einleitung

1.1 Einleitung

In der heutigen Zeit ist eine effiziente Verarbeitung und Darstellung von strukturierten Daten von entscheidender Bedeutung für zahlreiche Anwendungen und Prozesse. Eine standardisierte Weise Daten zu speichern und auszutauschen ist dabei über das Datenformat eXtensible Markup Language (XML). XML ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten im Format einer Textdatei, die von Mensch und Maschine lesbar ist. Eine wichtige Funktion der XML-Dateien ist die Transformation. Mit dieser lassen sich verschiedene Ausgabeformate wie Hypertext Markup Language (HTML)- und Portable Document Format (PDF)-Dateien generieren. So spielt die Transformation eine zentrale Rolle, um Informationen für verschiedene Verwendungszwecke zugänglich zu machen und sie in geeigneter Weise zu präsentieren.

1.2 Aufgabenstellung

Im Zuge des Moduls Internettechnologien ist eine Belegarbeit zur Erstellung von XML-Transformationen und einer Dokumentation dieser anzufertigen. Dazu soll der XML-Editor *editiX* verwendet werden. Dieser bietet die Möglichkeit, eine einfache Datenstruktur zu erstellen und eine Ausgabe über Transformation in HTML und PDF zu erzeugen. Zur Umsetzung dieser Aufgabe sind mehrere Anforderungen gegeben.

Es soll ein Datensatz von mindestens drei Objekten, einem Attribut und einer Datensequenz in einer XML angelegt werden. In einer XML Schema Definition (XSD) wird eine Schemadatei erstellt, welche die Datenstruktur mithilfe des Editors dokumentiert. Weiterhin ist eine Extensible Stylesheet Language Transformations (XSLT) zu schreiben,

die den HTML-Output generiert. Die HTML-Seite muss den Anforderungen entsprechen und ein einfaches Design, ohne besondere visuelle Elemente aufweisen. Dazu ist nur plain HTML zu verwenden. Im letzten Schritt wird aus der XSLT ein PDF-Output generiert. Dafür sind zwei Möglichkeiten gegeben. Zum einen kann eine direkte Apache Formatting Objects Processor (FOP)-Transformation durchgeführt werden, zum anderen kann eine DocBook-Transformation genutzt werden.

2 Dateiformate und ihre Strukturen

2.1 Umsetzung des XML Files

Die XML Struktur von diesem Projekt stellt eine Aufzählung von verschiedensten Saatguttütchen dar. Ziel der Struktur soll der Aufbau einer Datenbank aller Saatguttütchen im eigenen Besitz sein. Dazu werden jedem Tütchen ein ID-Attribut und Elemente zugeordnet. Diese Elemente sind auf die Tütchen bereits vorgedruckt und werden zur Unterscheidung des Gemüses und für Anbaubedingungen genutzt. Eine Auflistung der Elemente folgt in der Tabelle 2.1.

Element	Datentyp	Bemerkung
Saatguttütchen	string	root element
Tüte	string	branch element mit Attribut ID
Gemüseart	string	leaf element
Familie	string	leaf element
Inhaltsmenge	string	leaf element
Marke	string	leaf element
Aussaatzeit	string	leaf element
Erntezeit	string	leaf element
Standort	string	leaf element
Beschreibung	string	leaf element
Kosten	decimal	leaf element

Tabelle 2.1: Elemente XML-Struktur

Diese Elemente werden dann in die XML-Struktur umgesetzt. Dazu ist ein kurzer Ausschnitt aus dem Code hier dargestellt, um den Aufbau zu verdeutlichen.

```
1      <tuete id="001">
2          <gemueseart>BI0-Peperoni</gemueseart>
3          <familie>Nachtschattengewächs</familie>
4          <inhaltsmenge>ca. 30 Pflanzen</inhaltsmenge>
5          <marke>Sperli</marke>
6          <aussaatzeit>Februar bis April</aussaatzeit>
7          <erntezeit>Juli bis Oktober</erntezeit>
8          <standort>sonnig</standort>
9          <beschreibung>Die langen, roten, vitaminreichen...</beschreibung>
10         <kosten>3.99</kosten>
11     </tuete>
```

2.2 Umsetzung des XSD Files

Eine XSD definiert die Struktur von XML-Dokumenten und wird dabei auch in Form solcher geschrieben. Sie gibt vor, welche Elemente in einem XML vorhanden sein müssen, welche Typen von Daten sie enthalten können und wie sie miteinander verknüpft sind. XSDs sind hilfreich, um die Konsistenz und Gültigkeit von XML-Daten sicherzustellen, indem sie eine Art Regelwerk für deren Aufbau bereitstellen.

Eine XSD-Datei ermöglicht es außerdem mithilfe von *editiX* den schematischen Aufbau der XML in einer kleinen Dokumentation darzustellen. Diese ist in Abbildung 2.1 zu sehen. Darauf lassen sich alle bereits in Tabelle 2.1 dargestellten Elemente wiederfinden. Zusätzlich erhält man aus der Abbildung noch Informationen bezüglich der möglicher Anzahl an Tütchen, auf die die Datenbank der Saatguttütchen ergänzt werden kann. So kann man entnehmen, dass die Datenbank unendlich erweiterbar ist.

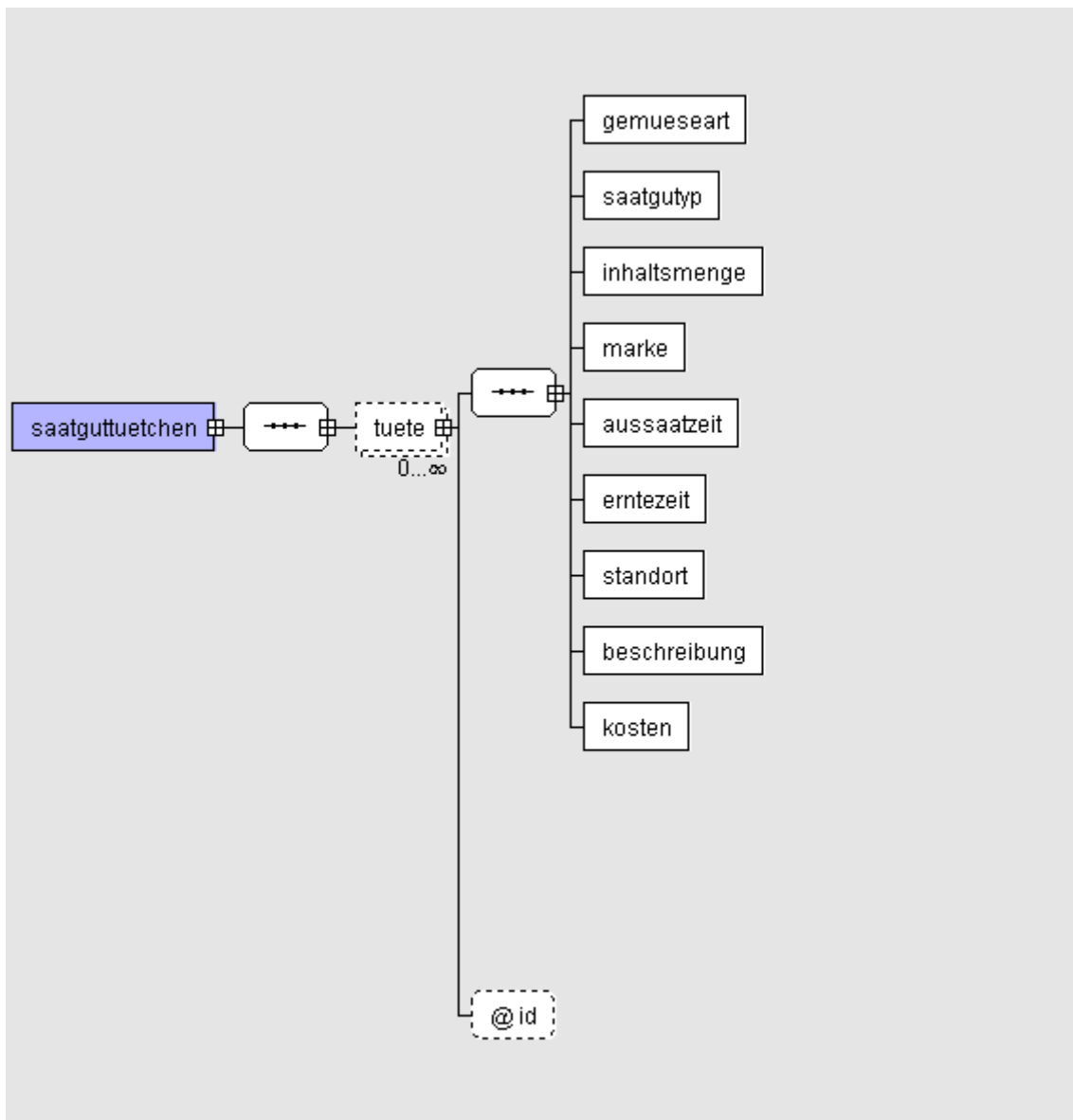


Abbildung 2.1: Schematischer Aufbau

2.3 Umsetzung des XSL Files

Die XSL-Datei beginnt mit der Version des XML-Dokuments und der Form der Zeichenkodierung in Unicode Transformation Format (UTF)-8 an. Das XSLT-Stylesheet definiert die Wurzelversion und den XML-Namespaces sowie eine Hauptvorlage, die das Wurzelelement des XML-Dokuments übereinstimmt und HTML aus der XML generiert. Zu sehen ist das im folgenden Codeabschnitt.

```
1      <?xml version="1.0" encoding="UTF-8"?>
2      <xsl:stylesheet version="1.0"
          xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3          <xsl:template match="/">
4              <html>
```

Darauf folgt der Header und der Body der Datei. Im Header wurde zuerst der Titel der Seite festgelegt. Um eine einheitliche Formatierung zu gewährleisten, wurden mehrere Design Entscheidungen im Header der XSL getroffen. So wurde das Logo der Hochschule für Technik, Wirtschaft und Kultur Leipzig (HTWK) über einen Selektor in die obere, rechte Seite der Website angelegt und die Größe dieser bestimmt. Ein weiterer Selektor beschreibt die Formatierung der Tabelle mit einem Identifikationsnummer (ID)-Element. Um nicht mit dem Logo zusammenzustößen wurde die Tabelle um 50 Pixel nach unten verschoben. Eine durchgängige schwarze Linie mit einer Stärke von zwei Pixeln markiert die Umrandung dieser und die Eigenschaft Kollaps führt dazu, dass die Rahmen der Tabellenzeilen und -zellen zusammengeführt werden. Der Hintergrund der Tabelle wurde auf ein helles und fast transparentes grün gesetzt.

Weiterhin wurde ein Selektor für alle Überschriften der Ordnung *H1* erstellt. Diese wurde mit einer neuen Schriftart und einer neuen Schriftgröße versehen. Die Überschrift sollte außerdem fettgedruckt geschrieben und unterstrichen sein. Zuletzt wurde ein Selektor erstellt, welche eine schwarze Linie ausgibt, wenn die Zelle der Tabelle die Klasse *horizontal-line* enthält. Diese wurde verwendet, um alle Tütchen auf der Website voneinander zu trennen. Da diese Design Entscheidungen in der XSL jeweils nur einmal auftreten, hätten diese auch im Quellcode in den Body eingepflegt werden können. Die Entscheidung ist allerdings dagegen gefallen, da es die einzelnen Funktionen und den gesamten Quellcode besser lesbar macht und Möglichkeiten zur einfachen Erweiterung des Codes mit diesen Funktionen bietet. Diese sind im folgenden Codeabschnitt zu sehen.

```
1      <style>
2      #top-right-image {
3          position: absolute;
4          top: 0;
5          right: 0;
6          margin: 10px;
7          width: 200px;
8          height: 100px;
9      }
10     #saatgut-table {
11         margin-top: 50px;
12         border: 2px solid black;
13         border-collapse: collapse;
14         background-color: rgba(144, 238, 144, 0.05);
15     }
16     h1 {
17         font-family: Times New Roman, sans-serif;
18         font-size: 48px;
19         font-weight: bold;
20         text-decoration: underline;
21     }
22     .horizontal-line td {
23         border-top: 2px solid black;
24     }
25 </style>
```

Mit den Design-Entscheidungen schließt der Header und der Aufbau der Seite wurde fokussiert. Der Aufbau beginnt mit dem Anlegen der Überschrift, der Tabelle und dem Initialisieren des HTWK-Logos. Danach wurde die Tabelle formatiert und die entsprechenden Werte den einzelnen Zellen zugeordnet. Der Tabellen-Header wurde dazu ohne Eingaben belassen, da in der endgültigen Formatierung der Tabelle ungewollte Dopplungen im Text aufgetreten wären.

In der Abbildung 2.2 ist Aufbau der Tabellenzeilen und -zellen gezeigt. Im grünen Rahmen sieht man die standardmäßige Dateneingabe in die Tabelle. Der blaue Rahmen zeigt die Eingabe des Standorts, ergänzt durch eine Anweisung, die bestimmten Wörtern oder

Wortgruppen Symbole zuweist. Die Eingabe der Bilder erfolgt im roten Rahmen. Hier kommt das ID-Attribut der Tüten zum tragen, womit den Gemüsesorten einfach durch Benennung die Bilder zugeordnet werden können. Im braunen Rahmen wird der Trennstrich verwendet, wie bereits in den Designanweisungen beschrieben wurden.

```
<td><strong>Erntezeit:</strong><br>
</br><xsl:value-of select="erntezeit"/></td>
<td>
  <strong>Standort:</strong><br>
</br>
  <xsl:value-of select="standort"/>
  <xsl:choose>
    <xsl:when test="standort='sonnig'"> &#9788;</xsl:when>
    <xsl:when test="standort='halbschattig'"> &#9728;</xsl:when>
    <xsl:when test="standort='schattig'"> &#9729;</xsl:when>
    <xsl:when test="standort='sonnig bis halbschattig'"> &#9788; &#9728;</xsl:when>
    <xsl:when test="standort='halbschattig bis schattig'"> &#9728; &#9729;</xsl:when>
    <xsl:otherwise></xsl:otherwise>
  </xsl:choose>
</td>
<td><strong>Kosten:</strong><br>
</br><xsl:value-of select="kosten"/> €</td>
<td></td>
</tr>
<tr class="horizontal-line">
  <td colspan="10"></td>
</tr>
<tr>
  <td colspan="9"><strong>Beschreibung:</strong></td>
</tr>
```

Abbildung 2.2: Aufbau der Tabelle

3 Transformation und Ausgabe

3.1 Erzeugung der HTML/PDF

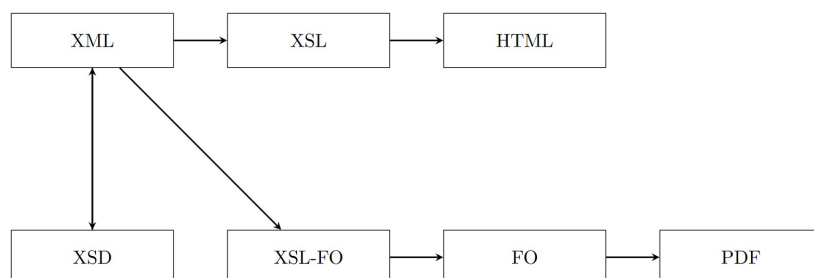


Abbildung 3.1: struktureller Aufbau der Transformation

3.2 Umsetzung in HTML

Durch die Transformation wird aus der gültigen XML-Datei und der zuletzt beschriebenen XSL-Datei mithilfe von *editiX* die HTML-Datei generiert. Die Anforderungen an das Design der HTML-Datei sind gering, es soll reines plain html benutzt werden. Die ersten Zeilen sind in Abbildung 3.2 dargestellt.

SAATGUTTÜTCHEN

HTWK

Hochschule für Technik,
Wirtschaft und Kultur Leipzig

Gemüseart: BIO-Peperoni	Familie: Nachtschattengewächs	Inhaltsmenge: ca. 30 Pflanzen	Marke: Sperli	Aussaatzeit: Februar bis April	Erntezeit: Juli bis Oktober	Standort: sonnig ☀	Kosten: 3.99 €	
Beschreibung: Die langen, roten, vitaminreichen Früchte mit dünner Fruchtwand entwickeln in rohem Zustand einen scharfen, pikanten Geschmack. Sie eignen sich ideal zum Würzen scharfer Gerichte. Paprika lieben einen stets sonnigen, windgeschützten Platz und benötigen durchlässigen, nährstoffreichen Boden. Tipp: Die erste Blüte entfernen, damit die Pflanzen später schneller neue Früchte bilden. Ab Mitte Mai (frostfrei) ins Freiland verpflanzen.								
Gemüseart: Mangold	Familie: Fuchsschwanzgewächs	Inhaltsmenge: ca. 50 Pflanzen	Marke: Kiepenkerl	Aussaatzeit: März bis Mai	Erntezeit: Juni bis Oktober	Standort: sonnig bis halbschattig ☀ *	Kosten: 2.99 €	
Beschreibung: Dekorative Blattstiele in allen Regenbogenfarben. Angenehmer Geschmack. Ertragreich. Leuchtendrot, orange, rosa, gelb und weiß sind die kräftigen Stiele dieser besonders dekorativen Mangoldzüchtung gefärbt. Abgesehen davon, dass man sie als leckeres, vitaminreiches Gemüse in der Küche und die Blätter wie Spinat verwenden kann, sind die Pflanzen mit ihren großen Blättern sehr attraktiv. In Gartenbeeten, Schalen, als Beeteinfassung oder zwischen Staudenbeeten setzen sie in Gruppen oder Reihen gepflanzt ungewöhnliche Akzente.								

Abbildung 3.2: Einblick in die generierte HTML

3.3 Umsetzung in PDF

Das generieren der PDF-Datei wird wie in Abbildung 3.1 dargestellt in diesem Beleg mit einer eigenen XSL-Datei (Saatguttuetchen-FO.xsl) realisiert. Ziel dabei ist es, sowohl die Übersichtlichkeit als auch den Informationsgehalt der PDF zu maximieren. So wird in der eigens angelegten Saatguttuetchen-FO.xsl durch Formatting Objects definiert, wie Elemente des PDF's angeordnet werden sollen. Im dargestellten Quellcodeausschnitt wird in Zeile 1 dargestellt, wie eine horizontale Ausrichtung der PDF erreicht wird und in den Zeilen 5-9 die Wahl von zwei Eigenschaften der Saatguttütchen als Beispiel angeführt.

```

1      <fo:simple-page-master master-name="Saatgut-Overview"
2          page-height="210mm" page-width="297mm" margin="15mm">
3          ...
4          <fo:table-row background-color="#CCCCCC">
5              <fo:table-cell>
6                  <fo:block
7                      font-weight="bold">Gemüseart</fo:block>
8              </fo:table-cell>
9              <fo:table-cell>
10                 <fo:block
11                     font-weight="bold">Familie</fo:block>
12             </fo:table-cell>

```

editiX generiert mithilfe dieser XSL-Datei und der XML-Datei durch die direkte FOP-Transformation eine Saatguttuetchen.fo Datei sowie die gewünschte PDF.

4 Zusammenfassung

4.1 Zusammenfassung

In diesem Beleg wurden verschiedene Aspekte des Themas Internettechnologie betrachtet. Vor allem auf den Gebieten der XSD, XML und XSL sowie mit *editiX* konnte viel praktische Erfahrung gesammelt werden. Darüber hinaus wurde außerdem deutlich, wie flexibel die Transformation von XML nach PDF und HTML ist.

4.2 Ausblick

Mit dem Abschluss dieser Belegarbeit steht man jetzt einem Erweiterungspotential gegenüber, auf deren Einzelheiten nun eingegangen werden soll. Aufbauend auf den bisherigen Ergebnissen könnte, abweichend vom Beleg, beispielsweise das Aussehen der HTML und der PDF komplexer gestaltet werden. Außerdem könnten weitere Elemente hinzugefügt werden, wie zum Beispiel ein Bestelllink oder die Herstellernummer. Elemente wie die Familie der Pflanzen könnte noch feiner untergliedert werden und somit die Struktur der XML erweitert. Weitere Vorschläge zugunsten der Funktionalität des Webinterfaces und der Datenbank könnten dann beispielsweise ein Programm zur Erweiterung der XML sein, damit man einfach neue Saatguttütchen der Liste hinzufügen kann. Ein weiterer Vorschlag zur Erweiterung wäre eine Filterfunktion für die einzelnen Elemente, zum Beispiel nach der Aussaatzeit, den Kosten oder auch nach der Inhaltsmenge. Abschließend lässt sich sagen, dass dies alles Gedankenspiele sind. Diese könnten für eine fortführende Aufgabe im Zusammenhang mit diesem Beleg oder als eigenständiger Beleg dienen, würden aber dennoch den Rahmen der Aufgabenstellung deutlich überspannen.

4.3 Verwendetes Bild- und Informationsmaterial der Saatguttütchen

Das verwendete Bild- und Informationsmaterial stammt von den Webseiten der Hersteller und außerhalb des Rahmens dieses Projekts erworbenen Saatguttütchen. Die Verwendung dient lediglich zur Veranschaulichung und Katalogisierung. Die Rechte bleiben bei den Herstellern der Saatguttütchen.