



**UNIVERSITÄT PADERBORN**  
*Die Universität der Informationsgesellschaft*

# ■ **Edirom Summer School 2010**

**Workshop: XML und XML-Technologien**  
**21.-24. November 2010**

■ **M.Sc. Julian Dabbert**

**Fakultät für Kulturwissenschaften der Universität Paderborn**

## ■ Block 5: Anwendungen

### ■ Anwendungen von XML

Eine Anwendung von XML ist kein konkretes Programm, sondern ein Vokabular für einen beschränkten Aufgabenbereich.

Beispiele :

- TEI (Text Encoding Initiative)
- MEI (Music Encoding Initiative)
- DocX (Textverarbeitung Microsoft)
- SVG (Scalable Vector Graphics)
- XHTML (eXtensible HyperText Markup Language)
- CSS (Cascading StyleSheets)
- Java / .NET (De-)Serialisierung

## ■ Block 5: Anwendungen

### ■ Bedingungen beim Einsatz von XML

Eine Anwendung von XML löst eine Problemstellung mit den Methoden von XML zu den Bedingungen („Kosten“), unter denen XML operiert:

- Keine Binärdaten (werden gekapselt oder extern gehalten)
- Keine Verschlüsselung (des Datencontainers)
- Geringer Kompressionsgrad (XML Dokumente sind „geschwätzig“)
- Höherer Aufwand zum Parsing (als ASCII Textdateien)
- Änderungen brauchen Aufwand (als ASCII Textdateien)

Ob Einsatz von XML sinnvoll ist, erfordert Kosten/Nutzen Abwägung

## ■ Block 5: Anwendungen

### ■ TEI (Text Encoding Initiative)

**TEI ist ein Textformat für Kodierung und Austausch von Texten**

- **De-facto Standard in den Geisteswissenschaften**
- **Spezialgebiete: Editionswissenschaften und Linguistik**
- **Ausgerichtet auf wiss. Arbeit (im Gegensatz zur Freizeitlektüre)**
- **Das Schema für TEI ist verfasst in der Metasprache ODD, aus dem ein Schema in XSD, DTD oder RelaxNG abgeleitet werden kann**
- **Dateiendung: .tei**
- **Weitere Infos → Summer School Workshop TEI (ab 27.10.2010)**

## ■ Block 5: Anwendungen

### ■ TEI Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>Hallo Welt!</title>
      </titleStmt>
      <publicationStmt>
        <p>Demo für Wikipedia</p>
      </publicationStmt>
      <sourceDesc>
        <p>Originales Werk, keine Vorlage</p>
      </sourceDesc>
    </fileDesc>
  </teiHeader>
  <text>
    <body>
      <p>Hallo Welt!</p>
    </body>
  </text>
</TEI>
```

## ■ Block 5: Anwendungen

### ■ MEI (Music Encoding Initiative)

**Verwandt mit TEI, jedoch optimiert für wissenschaftliche Musiknotation**

- **Neben reiner Notation auch geeignet für nichtgrafische Arbeit**
- **Alleinstellungsmerkmal Nichteindeutigkeit (Varianten)**
- **Open-Source Konkurrenz zu MusicXML und proprietären Binärformaten**
- **Dateiendung: .mei**
- **Weitere Infos → Summer School Workshop MEI (ab 30.10.2010)**

## Block 5: Anwendungen

### MEI Beispiel

```
<?xml version="1.0" encoding="UTF-8" ?>
<mei xmlns='http://www.music-encoding.org/ns/mei' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation="http://music-encoding.org/mei/schemata/2010-05/xsd/mei-all.xsd" />
  <meihead>
    <filedesc>
      <titlestmt>
        <title>Tonleiter</title>
      </titlestmt>
      <pubstmt/>
      <sourcedesc>
        <source xml:id="Tonleiter.mei_src.2"/>
      </sourcedesc>
    </filedesc>
    <encodingdesc/>
  </meihead>
  <music>
    <body>
      <mdiv>
        <score>
          <scoredef meter.unit="4" meter.count="4" key.sig="2s" key.mode="major">
            <staffgrp>
              <staffdef n="1" midi.div="1" label.full="Sopran" key.sig="2s" key.mode="major"
clef.shape="G" clef.line="2"/>
              <staffdef n="2" midi.div="1" label.full="Alt" key.sig="2s" key.mode="major"
clef.shape="F" clef.line="2"/>
            </staffgrp>
          </scoredef>
        </score>
      </mdiv>
    </body>
  </music>
</mei>
```

## ■ Block 5: Anwendungen

### ■ DocX (Textverarbeitung Microsoft)

Das „Open Office XML Format“ (DocX) ist von Microsoft konzipiert, um ab Word 2007 (endlich) das alte Doc-Format abzulösen.

- Prinzipiell reine Auszeichnungssprache wie HTML
- Interne Struktur: ZIP-komprimierte Hülle, XML Inhalte
- Microsoft Konkurrenz zu ODF (XML Format von „*Open Office*“:)
- Mehrere Ausprägungen: Textdokument, Präsentation und Tabellen
- Dateiendung: .docx, .pptx, .xlsx



## ■ Block 5: Anwendungen

### ■ Übung: DocX entpacken

Die Containerstruktur von DocX Dateien lässt sich öffnen, um den Dokumentinhalt manuell zu lesen:

- sample.docx umbenennen in sample.zip
- Im Verzeichnis befindet sich eine Ordnerstruktur mit einer Definitionsdatei: [Content\_Types].xml
- Im Unterverzeichnis word suchen nach document.xml
- Diese Datei im Editor öffnen

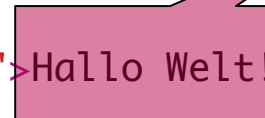
Wer mit Word arbeitet, verändert diese komplexe Dateistruktur, ohne das zu bemerken: „Transparenz“ in der Informatik

## ■ Block 5: Anwendungen

### ■ DocX Beispiel: document.xml in sample.docx

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<w:wordDocument xmlns:ve="
http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:o="urn:schemas-microsoft-com:office:office" [...] >
  <w:body>
    <w:p w:rsidR="00EA68DC" w:rsidRPr="00C703AC"
w:rsidRDefault="00EA68DC" w:rsidP="00EA68DC">
      <w:pPr>
        <w:rPr>
          <w:lang w:val="es-ES-Ts"/>
        </w:rPr>
      </w:pPr>
      <w:r>
        <w:t xml:space="preserve">Hallo Welt!</w:t>
      </w:r>
    </w:p>
    <w:sectPr w:rsidR="001A6335" w:rsidSect="001A6335">
      [...]
    </w:sectPr>
  </w:body>
</w:wordDocument>
```

Der Textinhalt



## ■ Block 5: Anwendungen

### ■ SVG (Scalable Vector Graphics)

**SVG spezifiziert statische und dynamische 2D Vektorgrafiken.**

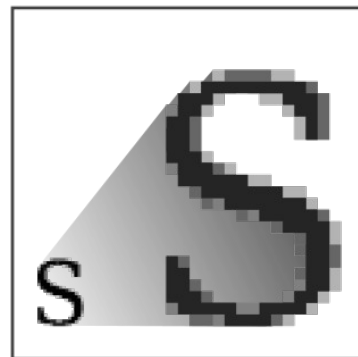
- **Vektorgrafiken sind i.d.R enorm platzsparend im Vergleich zu Bitmap Grafiken, da sie gesamte Inhaltsbereiche beschreiben**
- **Vektorgrafiken können beliebig vergrößert werden, ohne dass Artefakte sichtbar werden (Klötzchenbildung)**
- **Darstellbar durch alle aktuellen Webbrowser (bis auf IE vor v.9)**
- **Dateiendung: .svg**

## ■ Block 5: Anwendungen

### ■ SVG (Scalable Vector Graphics)

Unterschied zu sog. Rastergrafiken:

- Vektorgrafik ist aus „grafischen Primitiven“ aufgebaut:  
Pfad, Kreis, Ellipse, Rechteck, Linie, Polygon & Text
- Diese Primitiven ergeben im Zusammenspiel ein komplexeres Bild
- Auch beim Vergrössern bleiben diese Strukturen zueinander im Verhältnis, anders als bei den Pixeln einer Rastergrafik



**BITMAP**  
.jpeg .gif .png



**OUTLINE**  
.svg

## Block 5: Anwendungen

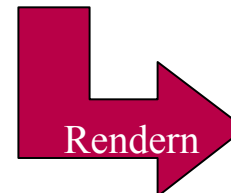
### SVG Beispiel

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

```
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
  <polygon points="220,100 300,210 170,250"
  style="fill:#cccccc;stroke:#000000;stroke-width:1"/>
</svg>
```

Vektorpunkt: X,Y

Füllfarbe: hellgrau



## ■ Block 5: Anwendungen

### ■ Übung: SVG benutzen

Nutzen Sie einen Editor sowie den Firefox Browser, um die folgenden geometrischen Formen zu zeichnen und anzusehen:

- Blaues Dreieck (Blau: #0000FF)
- Rotes Quadrat (Rot: #FF0000)
- Grünes Fünfeck (Grün: #00FF00)
- Lila Trapez (Lila: #??????)

## ■ Block 5: Anwendungen

### ■ XHTML (eXtensible HyperText Markup Language)

**Textbasierte Auszeichnungssprache, Neuformulierung von HTML in XML**

- Als Grundlage XML statt komplizierterem SGML
- XML-konforme Tags:  
Anstatt von „<br>“ (HTML) verwende „<br/>“ (XHTML)
- Namespaces (Einbettung anderer XML-Segmente möglich)
- X(HT)ML Parser sind strikt, HTML Parser dagegen fehlertolerant  
Diese Fehlertoleranz macht Probleme bei automatischer DV
- Visuelle Gestaltung von XHTML-Elementen ab XHTML Version 2.0  
nur noch über externes CSS
- Dateiendung: .xhtml

## ■ Block 5: Anwendungen

### ■ XHTML Beispiel

#### Vergleich HTML(links) mit XHTML (rechts)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Beispiel</title>
  </head>
  <body>
    <h1>Beispielseite</h1>
    <p>Ein Absatz
    <p>Noch ein<br>
      Absatz
    <ol>
      <li>Listelement
      <li>Listelement
    </ol>
    <p>
    <img src=bild.gif alt="Bildmotiv">
  </body>
</html>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de">
  <head>
    <title>Beispiel</title>
  </head>
  <body>
    <h1>Beispielseite</h1>
    <p>Ein Absatz</p>
    <p>Noch ein<br />
      Absatz</p>
    <ol>
      <li>Listelement</li>
      <li>Listelement</li>
    </ol>
    <p>
    
  </p>
</body>
</html>
```



## ■ Block 5: Anwendungen

### ■ CSS (Cascading StyleSheets) - **Kein XML!**

**Deklarative Stylesheet („Formatvorlage“) Sprache für visuelles Layout**

- Sprache ist kein Dialekt von XML, wird aber häufig in Verbindung mit XML Dateien zur Präsentation von Daten eingesetzt
- Konkurrenz zur XML Formatierungssprache XSL-FO
- Syntax: Hierarchisch geschachtelte Blöcke von Deklarationen
- Anwendung: Zu präsentierendes Datenobjekt, meist (X)HMTL oder XML Datei, enthält entweder Verweis auf externe CSS Datei oder CSS Inhalt wird in Datendatei eingefügt
- Vorteil: Trennung von Inhalt und Design
- Weitverbreiteter de-facto-Standard im Internet
- Dateiendung: .css

## Block 5: Anwendungen

### CSS Beispiel

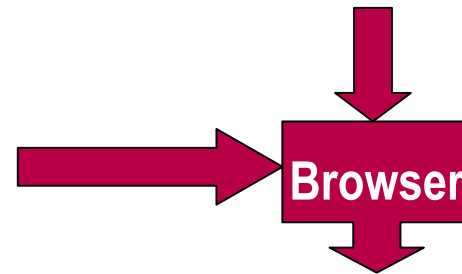
#### Deklaration im CSS

note {

```
position: relative;
left: 15%;
width: 80%;
padding: 30px;
padding-bottom: 45px;
border: 0px solid black;
line-height: 1.5em;
color: black;
font-weight: bold;
text-align: justify;
background-color: #eeeeee;
}
```

#### Einbindung im XHTML

<p class="note">  
Dies ist ein kleiner Testabsatz.  
</p>



#### Visuelle Präsentation

Dies ist ein kleiner Testabsatz.

## ■ Block 5: Anwendungen

### ■ Übung: CSS benutzen

**Benutzen Sie CSS für visuelle Präsentation**

- Finden Sie die beiden Dateien  
`css_xhtml_testdatei.xhtml` und `css_testdatei.css`
- Öffnen Sie die XHTML Datei im Firefox, die CSS Datei im Editor
- Verändern Sie nur die CSS Datei, und erwirken Sie die folgenden Änderungen:
  - Überschrift weiss auf blauem Hintergrund
  - Note schwarz auf rotem Hintergrund
  - Text blau aufweissem Hintergrund

**Sobald fertig: Freies Experimentieren ist erwünscht**



## ■ Block 5: Anwendungen

### ■ Möglichkeiten von CSS

**Was mit CSS alles möglich ist, wird durch den CSS Zen Garten veranschaulicht: Beim Wechseln des Designs bleibt die HTML-Seite gleich, nur das Stylesheet ändert sich – trotzdem wirkt die Seite total verändert**

**Link: <http://www.csszengarden.com/>**

## ■ Block 5: Anwendungen

### ■ Java / .NET (De-)Serialisierung

Moderne Programmiersprachen bieten Konzepte, die das Programmieren vereinfachen, auch unter Zuhilfenahme von XML. Dazu zählen:

- Modellierung (Planung von Programm-Architektur)
- Nachrichtentausch (Austausch zwischen Programmteilen)
- Persistenz (Längerfristige Speicherung von Programmdaten)

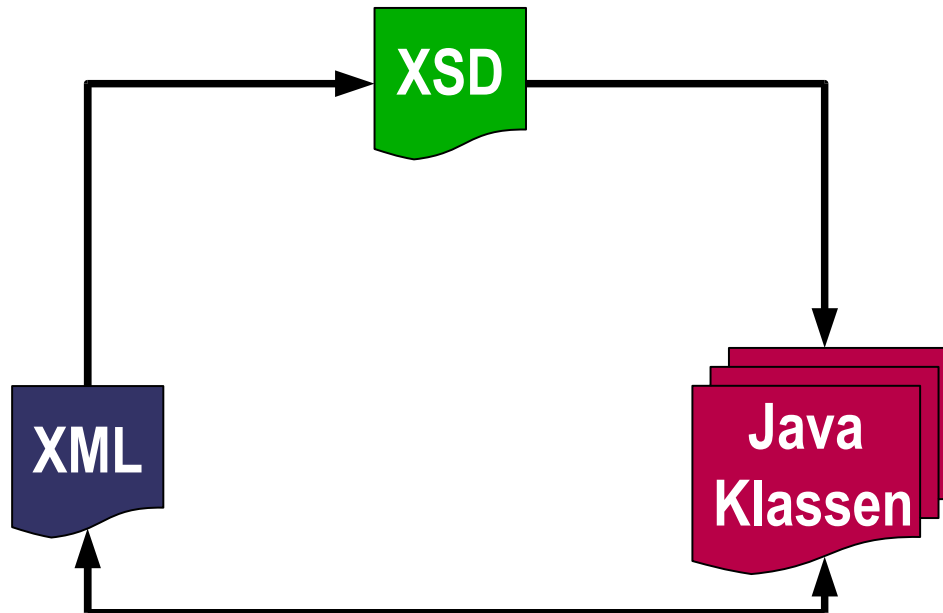
Hier wird kurz die Persistenz durch Serialisierung beleuchtet.

Die Idee ist es, XML Strukturen direkt mit Programmcode zu verknüpfen, um damit schnell und zuverlässig korrekten Programmcode zu produzieren

## ■ Block 5: Anwendungen

### ■ Java / .NET (De-)Serialisierung

Beispiel: Serialisierung mit Java JAXB





**UNIVERSITÄT PADERBORN**  
*Die Universität der Informationsgesellschaft*



**Ende des Blocks 5**