

Von spitzen Federn zu spitzen Klammern

Modernes Publizieren mit XML

Lange Nacht der Wissenschaften in Halle – 06. Juli 2012

Dr. Thomas Meinike

Hochschule Merseburg | FB Informatik und Kommunikationssysteme

<http://www.iks.hs-merseburg.de/~meinike/>

thomas.meinike@hs-merseburg.de

Zur Person

- Naturwissenschaftliches Studium
(Computernutzung / Programmierung seit 1987)
- Netzerfahrung seit den frühen 90er Jahren
- Autor, Entwickler und Referent seit 1994
- Lehrender im Bereich Technische Redaktion
der Hochschule Merseburg seit 1997
- Lehr- und Arbeitsgebiete:
 - Auszeichnungs- und Skriptsprachen
 - Online-Dokumentation und Web-Entwicklung
 - XML-Technologien



Technische Redaktion in Merseburg



- Diplomstudiengang 1995 bis 2008
Kommunikation und Technische Dokumentation
- Masterstudiengang seit 2006
Technische Redaktion und Wissenskommunikation (4 Sem. / M. A.)
- Bachelorstudiengang seit 2010
Technische Redaktion und
E-Learning-Systeme
(6 Sem. / B. Eng.)



Technische Redaktion in Merseburg

→ Technische Redaktion und E-Learning-Systeme (Bachelor)

STUDIENINHALTE (Auswahl)

Studienphase 1

1. bis 2. Semester

- Angewandte Informatik I + II
- Gestaltung von Offlinemedien I + II
- Gestaltung von Onlinemedien I + II
- Angewandte Mathematik I + II
- Technische Grundlagen I + II

Studienphase 2

3. bis 6. Semester

Wahl der Vertiefungsrichtung

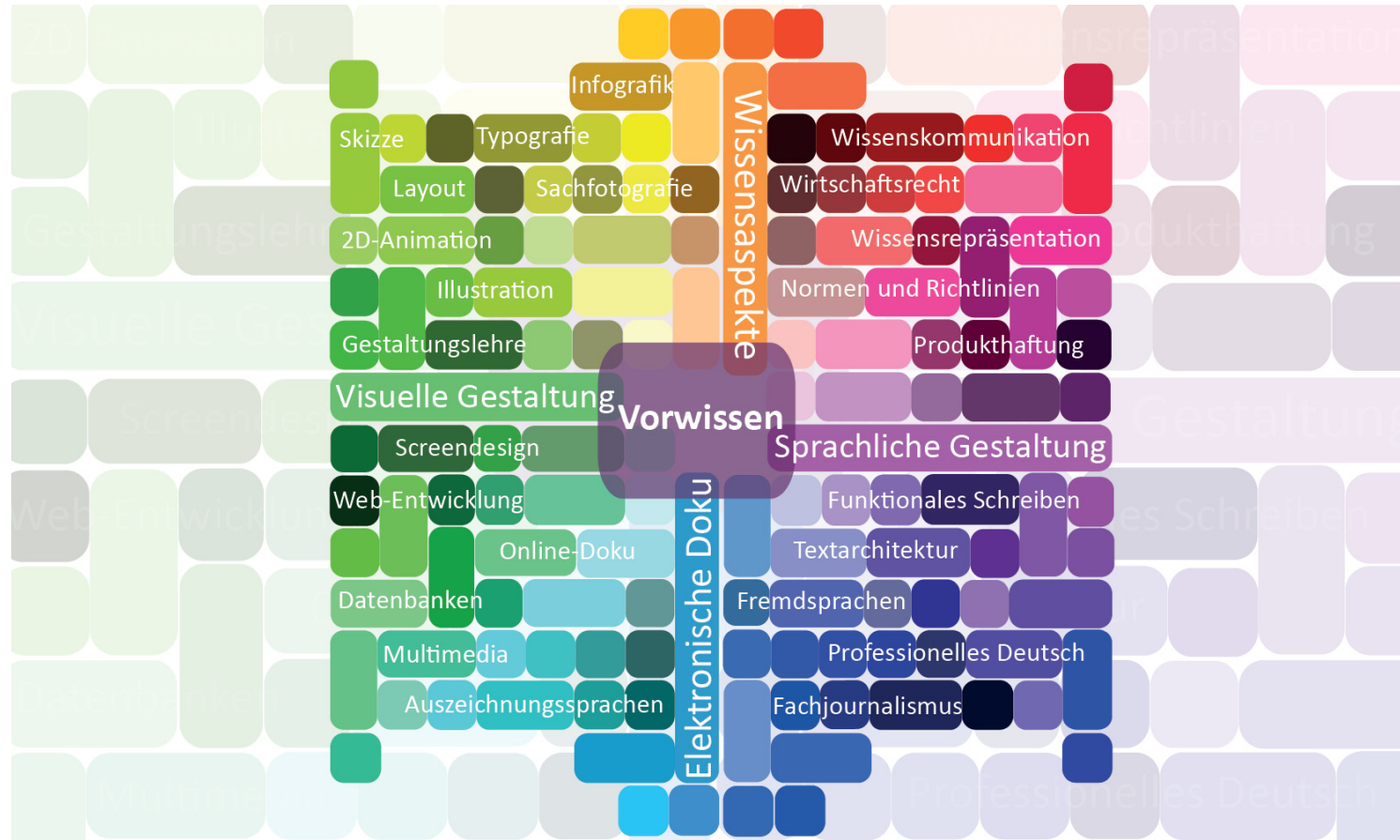
Technische Redaktion (TR) oder

E-Learning-Systeme (ELS)

- **TR:** Gestaltung lerneffektiver Texte | Typographie / Layout | Online-Dokumentation | Textoptimierung und Usability | Contentmanagementsysteme
- **ELS:** Einführung in tutorielle Systeme | DVD-Authoring | Einführung künstliche Intelligenz | Maschinelles Lernen | E-Learning

Technische Redaktion in Merseburg

➔ Technische Redaktion und Wissenskommunikation (Master)



Technische Redaktion in Merseburg



➔ Mehr Informationen unter:

<http://www.technische-redakteure-merseburg.de>

<http://youtube.com/user/TechnischeRedakteure>

The screenshot shows the YouTube channel page for 'Technische Redakteure Merseburg'. The channel has 3 subscribers and 1,600 video views. The page displays a grid of five uploaded videos:

- Studienarbeiten im Fach 3D-Visuali...** (2:02) - 58 Aufrufe vor 4 Wochen
- Neue Postkartenserie mit Studienar...** (1:33) - 33 Aufrufe vor 4 Wochen
- E-Book-Ergebnisse im Fach Online...** (2:00) - 54 Aufrufe vor 4 Wochen
- Technische Redaktion und E-Learn...** (3:47) - 694 Aufrufe vor 2 Jahren
- Technische Redaktion und Wissen...** (3:22) - 721 Aufrufe vor 2 Jahren

{Etwas Theorie muss sein ...}



XML ...

- **Ex**tensible **M**arkup **L**anguage
- wurde 1998 vom World Wide Web Consortium (W3C) eingeführt
- ist keine eigenständige Sprache, sondern ein Konzept zur Definition von Auszeichnungssprachen
- definiert (wenige) Regeln, welche beim Schreiben und der anschließenden Verarbeitung exakt einzuhalten sind

XML ...

- basiert grundlegend auf SGML (ISO-Standard von 1986):
Notieren von Elementen, Attributen, Tags usw. – wurde jedoch kompakter und einfacher formuliert, u. a. für Web-Einsatz
- bildet mit weiteren Konzepten eine Technologiefamilie
- wird mittlerweile für vielfältige Anwendungen, speziell in der Technischen Dokumentation, eingesetzt
- häufig werden „gebrauchsfertige“ Sprachen und Eigenentwicklungen kombiniert

XML ...

- ermöglicht medienneutrale Datenhaltung: Inhalte werden ihrer Bedeutung nach strukturiert abgelegt, frei von jeglichen Formatangaben
- somit lassen sich aus einem Quelldokument unterschiedliche Zieldokumente erzeugen (Prinzip des Single-Source-Publishings)
- im Bereich der Technischen Redaktion werden z. B. Webseiten, Onlinehilfen, PDF-Dateien, E-Books produziert
- dahinter stehen natürlich einige technologische Konzepte, welche hier nur ansatzweise betrachtet werden können ...

XML ...

→ Grundgerüst:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<wurzelement>  
  <!-- weitere Inhalte -->  
</wurzelement>
```

- 1. Zeile = XML-Deklaration mit Zeichenkodierung (UTF-8 ist heute der etablierte Unicode-Standard für weltweite Kommunikation)
- jedes XML-Dokument besitzt genau ein **Wurzelement** (umschließt alle weiteren Inhalte)

XML-Kommentar: <!-- Text -->

XML ...

→ Tags <...> bilden die Auszeichnungsmarken für Elemente:

`<elementname>Inhalt</elementname>`

Starttag

Endtag

→ Leere Elemente (kein Inhalt, oft mit Attributen verknüpft):

`<elementname></elementname>` oder kurz `<elementname/>`

→ Elementverschachtelung (ermöglicht die eigentliche Strukturierung):

```
<name1>
  <name2>Inhalt</name2>
</name1>
```

`name2` ist Kindelement von `name1`
`name1` ist Elternelement von `name2`

XML ...

- Attribute erweitern Elemente um zusätzliche Eigenschaften:

```
<elementname attributname="attributwert">Inhalt</elementname>
```

- mehrere Attribute pro Element sind möglich, pro Typ jedoch nur 1x, Reihenfolge ist beliebig wählbar:

```
<elementname x="a" y="b" z="c">Inhalt</elementname>
```

```
<elementname z="c" x="a" y="b">Inhalt</elementname>
```

- Attribute stehen nur im Starttag, ihre Werte paarweise in doppelten oder einfachen Anführungszeichen

→ Namensregeln für Element- und Attributnamen:

- ✓ am Anfang ein Buchstabe oder der Unterstrich (_), danach weitere Buchstaben oder Ziffern: a1, _a1, _1a, aber nicht 1a
- ✓ Namen von Start- und Endtags müssen identisch sein
- ✓ Schreibweise wird hinsichtlich Groß- bzw. Kleinbuchstaben unterschieden, also <a>..., aber nicht <a>...
- ✓ Umlaute und Sonderzeichen je nach Zeichenkodierung erlaubt
- ✓ Doppelpunkte sind in Namen nicht erlaubt (reserviert für Namensräume)
- ✓ XML am Anfang von Namen in jeglicher Schreibweise vermeiden
- ✓ Praxistipp: lateinisches Alphabet + Ziffern, ggf. mit Binde- bzw. Unterstrichen

XML ...

- von Wohlgeformtheit spricht man, wenn die genannten und weitere Regeln erfüllt werden = **Minimalanforderung an XML-Dokumente!**
- liegt zusätzlich ein **Datenmodell** vor [Document Type Definition, (DTD) oder XML-Schema (XSD)], können Dokumente auch auf ihre Gültigkeit (Validität) überprüft werden
- ein solches Modell beschreibt die Namen der Elemente und Attribute, die Anzahl und Reihenfolge der Elemente, die Zuordnung der Attribute zu Elementen sowie mögliche Inhalte und ggf. Datentypen

{Langsam Zeit für ein Beispiel ...}

→ In der Mitteldeutschen Zeitung (MZ) vom 16. Juni stand auf Seite 1:

ANSICHTSSACHE

Volksmusiker kennen keinen Schmerz

Stefan Mross kämpft für das Publikum jenseits der 50.

Wer hätte das gedacht: Stefan Mross mag die Menschen jenseits der 50. Dabei ist der Volksmusik-Held selber doch erst 36, gefühlt allerdings schon immer um die 70. Wie das zusammenpassen soll, fragen Sie sich? Das müssen Sie Herrn Mross eigentlich selber fragen. Aber wir können es ja mal zu erklären versuchen.

Stefan Mross, 1975 in Traunstein im schönen Bayern geboren, muss die Musik schon immer im Blut gehabt haben. Und das Zeug zum Star. Als Karl Moik, der Pate aller



Stefan Mross

FOTO: ARCHIV

deutschsprachigen Schunkelmusikanten, den damals 13-jährigen Buben entdeckte, soll er geweint haben. Da konnten einem selber

die Tränen kommen. Und so ging das weiter, erinnern wir uns nur an das Traumpaar der Volksmusik, das der trompetende Stefan Mross mit der ostdeutschen Dirndlträgerin Stefanie Hertel abgegeben hat. Doch ach, die Liebenden trennten sich, und wieder gingen uns die Augen über.

Nun macht sich Mross stark für Menschen, die schon in ihrer zweiten Lebenshälfte angekommen sind: „Die gehen noch in den Laden und kaufen CDs, während der Zwölfjährige daheim sitzt und sich

die Musik zunehmend irgendwo im Internet runterlädt“, sagte der Künstler der „Neuen Osnabrücker Zeitung“. Dagegen würden im öffentlich-rechtlichen Fernsehen die Vorlieben und Sehgewohnheiten der „Zielgruppe 40 oder 50 plus“ häufig ignoriert.

So sieht das aus: traurig. Und es wird nicht besser. Wenn Mross selber einmal jenseits der Fünfzig ist, könnte es verdammt einsam um ihn werden. Aber ein richtiger Volksmusiker kennt keinen Schmerz. *Andreas Montag*

→ Kurze Artikelanalyse: Rubrikname, Haupt-/Untertitel, Fließtext mit teilweiser Hervorhebung (fett), Bild mit Beschreibung, Autorennamen

XML-Beispiel

→ XML-Strukturierung des MZ-Artikels:

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE Artikel SYSTEM "artikel.dtd">
```

Dieses Dokument ist gültig.

```
<Artikel Zeitung="MZ" Datum="2012-06-16" Seite="1">
```

```
<Autor>Andreas Montag</Autor><Rubrik>Ansichtssache</Rubrik>
```

```
<Titel>Volksmusiker kennen keinen Schmerz</Titel>
```

```
<Untertitel>Stefan Mross kämpft für das Publikum jenseits der 50.</Untertitel>
```

```
<Inhalt>
```

```
<Absatz><Hervorhebung Art="stark">Wer hätte das gedacht: ...</Hervorhebung>Stefan Mross mag ...</Absatz>
```

```
<Absatz>Stefan Mross, 1975 in Traunstein im schönen Bayern geboren, ...</Absatz>
```

```
<Bild><Name>mross.jpg</Name><Text>Stefan Mross (FOTO: ARCHIV)</Text></Bild>
```

```
<Absatz>Nun macht sich Mross stark für Menschen, ...</Absatz>
```

```
<Absatz>So sieht das aus: traurig. Und es wird nicht besser. ...</Absatz>
```

```
</Inhalt>
```

```
</Artikel>
```

XML-Beispiel

→ DTD zur Artikel-Struktur:

```

<!ELEMENT Artikel      (Autor, Rubrik?, Titel, Untertitel?, Inhalt+)>
<!ELEMENT Autor        (#PCDATA)>
<!ELEMENT Rubrik       (#PCDATA)>
<!ELEMENT Titel        (#PCDATA)>
<!ELEMENT Untertitel   (#PCDATA)>
<!ELEMENT Inhalt       (Absatz | Bild)*>
<!ELEMENT Absatz       (#PCDATA | Hervorhebung)*>
<!ELEMENT Hervorhebung (#PCDATA)>
<!ELEMENT Bild         (Name, Text?)>
<!ELEMENT Name         (#PCDATA)>
<!ELEMENT Text         (#PCDATA)>
<!ATTLIST Hervorhebung Art (einfach | stark) #REQUIRED>
<!ATTLIST Artikel      Zeitung CDATA #REQUIRED Datum CDATA #IMPLIED Seite CDATA #IMPLIED>

```

Indikatoren:

* 0...n (beliebig oft)
 + 1...n (mindestens 1x)
 ? 0/1 (optional)

XML-Beispiel

→ XML-Struktur im Browser:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Artikel SYSTEM "artikel.dtd">
- <Artikel Seite="1" Datum="2012-06-16" Zeitung="MZ">
  <Autor>Andreas Montag</Autor>
  <Rubrik>Ansichtssache</Rubrik>
  <Titel>Volksmusiker kennen keinen Schmerz</Titel>
  <Untertitel>Stefan Mross kämpft für das Publikum jenseits der 50.</Untertitel>
  - <Inhalt>
    - <Absatz>
      <Hervorhebung Art="stark">Wer hätte das gedacht: ...</Hervorhebung>
      Stefan Mross mag ...
    </Absatz>
    <Absatz>Stefan Mross, 1975 in Traunstein im schönen Bayern geboren, ...</Absatz>
    - <Bild>
      <Name>mross.jpg</Name>
      <Text>Stefan Mross (FOTO: ARCHIV)</Text>
    </Bild>
    <Absatz>Nun macht sich Mross stark für Menschen, ...</Absatz>
    <Absatz>So sieht das aus: traurig. Und es wird nicht besser. ...</Absatz>
  </Inhalt>
</Artikel>

```

→ Das sieht ja noch nicht besonders nutzerfreundlich aus. Entscheidend ist die Verarbeitung der Inhalte in „konsumierbare“ Medienprodukte!

{Wir brauchen noch etwas
Theorie für die Praxis ...}

XML-Verarbeitung

➔ XSL-Technologiefamilie (Extensible Stylesheet Language):

XSLT

Transformationssprache zur Verarbeitung von XML zu HTML, Text oder anderen XML-Strukturen

XSL-FO

Layout-orientierte Ausgabe für Druck (PDF) / FO ist selbst ein XML-Format

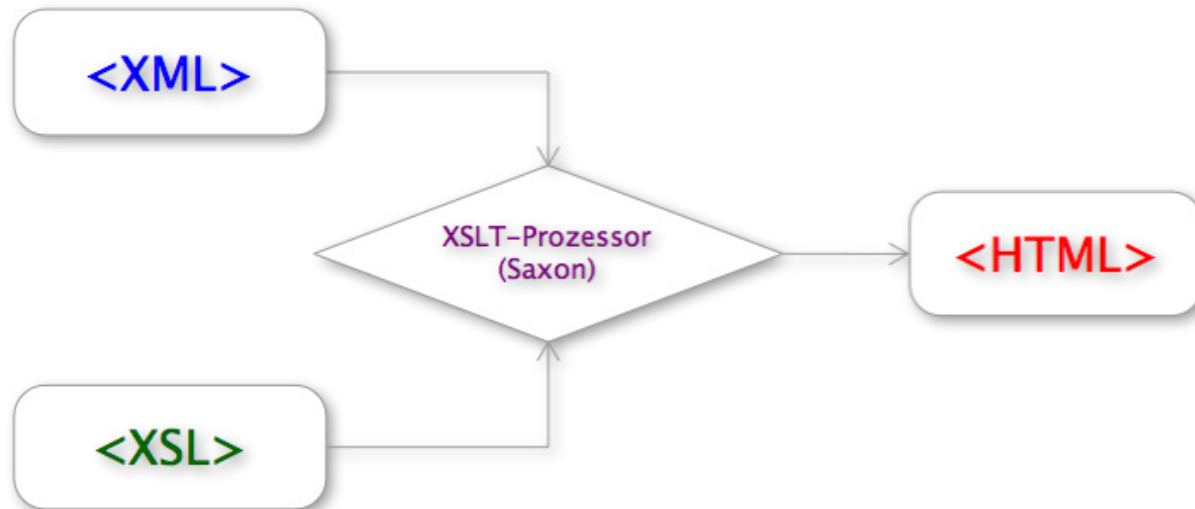
XPath

Abfragesyntax für XML-Zugriff und Auswahl von Inhalten

+ Spezielle Software-Tools: **XSLT-Prozessoren bzw. FO-Formatierer**

XML-Verarbeitung

- **XSLT** ist eine speziell für die XML-Verarbeitung ausgelegte (funktionale) Programmiersprache, selbst in XML formuliert
- Beispiel: HTML-Generierung nach diesem Ablaufschema:



XML-Verarbeitung

→ Steuerung erfolgt über sich gegenseitig aufrufender Templates

→ Zugriff auf das Quelldokument mittels XPath-Abfragen

(hier bei `match` bzw. `select`):

```
<xsl:template match="Artikel">
```

```
<html><head>...</head><body>
```

```
<!-- Titel, Untertitel, Autor ausgeben -->
```

```
<h1><xsl:value-of select="Titel"/></h1>
```

```
<h2><xsl:value-of select="Untertitel"/></h2>
```

```
<p>Von <em><xsl:value-of select="Autor"/></em></p>
```

```
<!-- Inhalte verarbeiten / weitere Templates anwenden -->
```

```
<xsl:apply-templates select="Inhalt"/>
```

```
</body></html>
```

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

HTML-Ausgabe:

Volksmusiker kennen keinen Schmerz

Stefan Mross kämpft für das Publikum jenseits der 50.

Von *Andreas Montag*

Wer hätte das gedacht: ... Dabei ist der Volksmusik-Held ...

Stefan Mross, 1975 in Traunstein im schönen Bayern geboren, ...



Stefan Mross (FOTO: ARCHIV)

Nun macht sich Mross stark für Menschen, ...

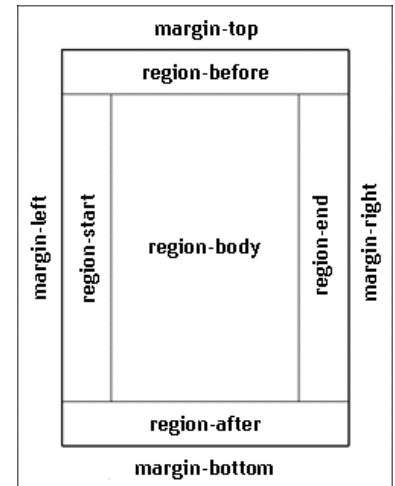
So sieht das aus: traurig. Und es wird nicht besser. ...

{Praktische Demonstration ...}

XML-Verarbeitung

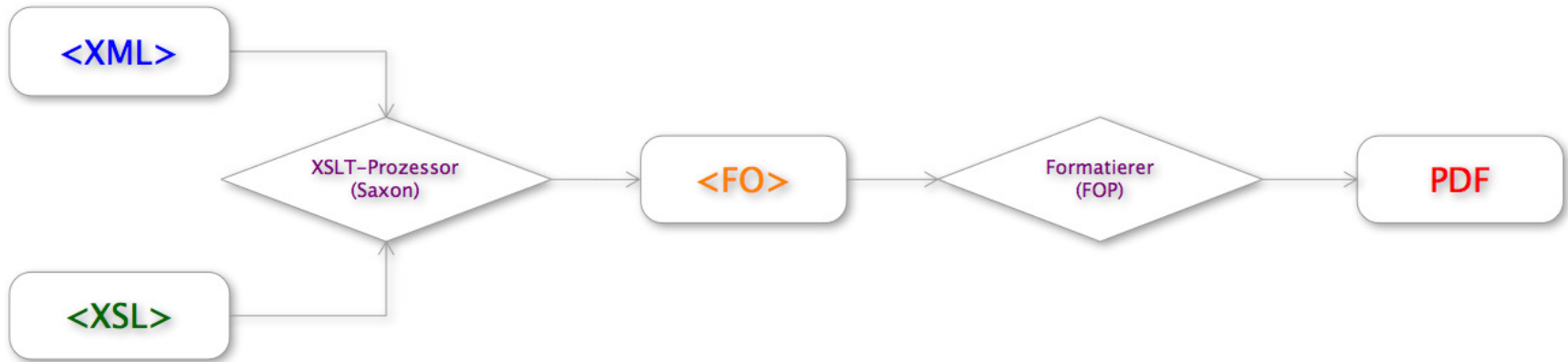
- XSL-FO ist ein Format zur Beschreibung von Seitenlayouts (z. B. A4)
- Vordefinierte Regionen werden mit Inhalten gefüllt

```
fo:root
|
|---fo:layout-master-set
|   |
|   |---fo:simple-master-set master-name="..."
|   |   |
|   |   |---fo:region-body | -before | -after | -start | -end
|   |   |
|   |---fo:page-sequence-master master-name="..."
|   |
|   |...
|---fo:declarations
|   |
|   |...
|---fo:page-sequence master-reference="..."
|   |
|   |---fo:static-content flow-name="xsl-region-..."
|   |---fo:flow flow-name="xsl-region-..."
```



XML-Verarbeitung

➔ Typische Prozesskette der automatisierten PDF-Produktion:



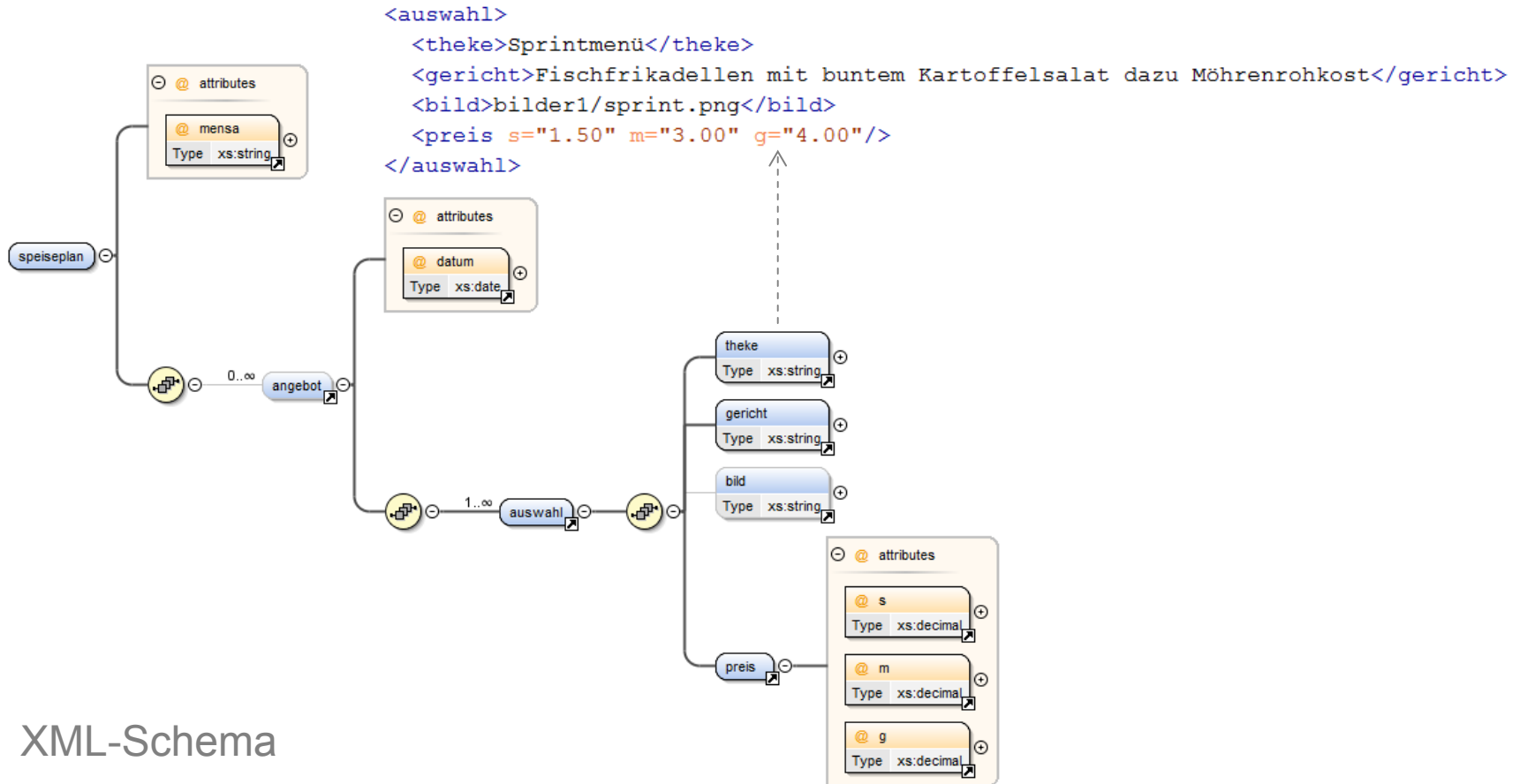
➔ Zunächst entsteht aus dem Quelldokument via XSLT-Prozessor ein Zwischen-FO-Dokument, welches schließlich in das Zielformat PDF überführt wird.

Formatierer: u. a. Antenna House (kommerziell, hochwertige Ergebnisse);

Apache FOP (Open Source, unterstützt noch nicht alles, brauchbare Ergebnisse)

Beispiel Mensa-Speiseplan


➔ Umsetzung der im Web verfügbaren Daten als XML-Struktur:



XML-Schema

Beispiel Mensa-Speiseplan

→ Ergebnis: **Speiseplan der Mensa Merseburg**
 (PDF via FO) **Angebot am 2012-07-06**

Theke	Gericht	Studenten	Mitarbeiter	Gäste	Bild
Sprintmenü	Fischfrikadellen mit buntem Kartoffelsalat dazu Möhrenrohkost	1.50 €	3.00 €	4.00 €	
Kleine Theke	Griechische Kartoffelpfanne mit buntem Gemüse, Oliven, Hirtenkäse	2.20 €	3.60 €	4.40 €	
Aktionstheke	Geschmortes Zwiebelfleisch vom Schweinenacken in Majoransauce / Wahlbeilagen: Mischgemüse, Kartoffelspalten, Bunter Salat, Röstkartoffeln	2.60 €	3.90 €	4.70 €	
Salat	leckere Salate zur Selbstentnahme Preis nach Gewicht (je 100g)	0.55 €	0.90 €	1.10 €	
Dessert	Dessertschälchen	0.50 €	0.80 €	1.00 €	

{Praktische Demonstration ...}

XML-Anwendungen in der TR

- Spezielle Dokumentationsstandards:
 - [DocBook](#) (Buch-orientiert mit Kapiteln, Abschnitten, ...)
 - [DITA](#) (Topic-orientiert, separate Informationseinheiten unter dem Aspekt der besseren Wiederverwendbarkeit)

- Webseiten: [\(X\)HTML](#)

- CHM-Onlinehilfen (CAB-Container mit [XHTML](#))

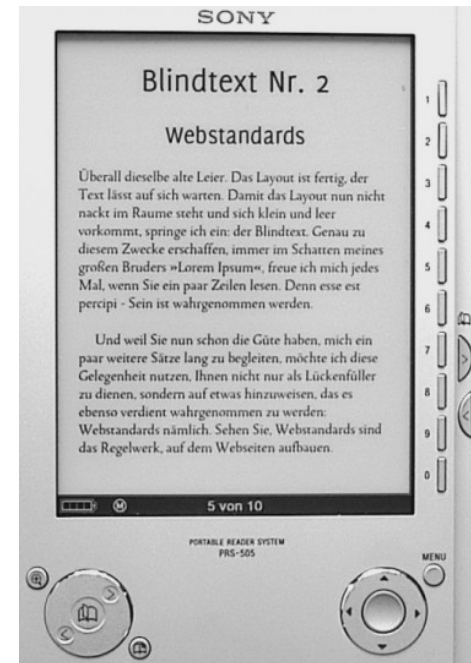
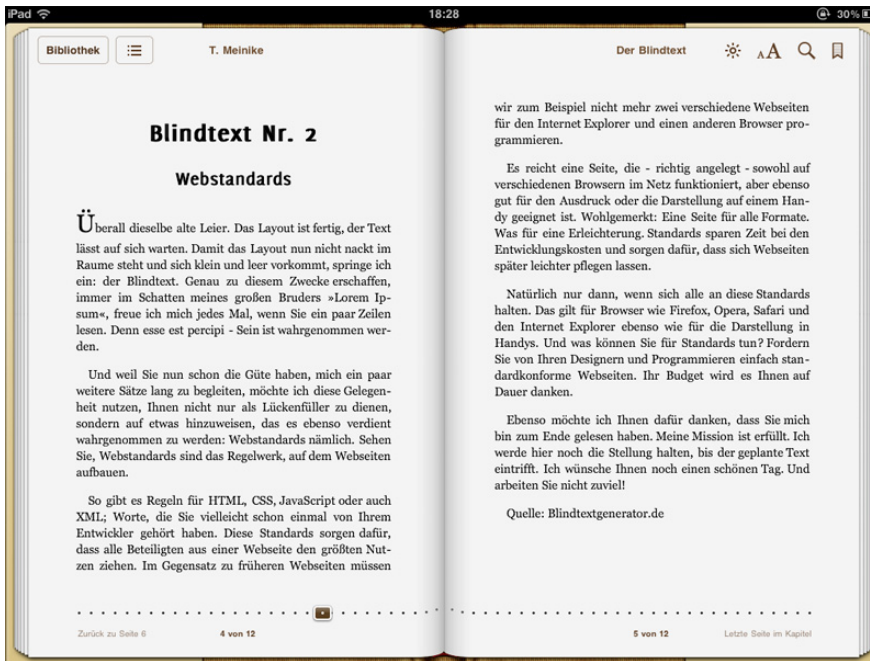
- Newsfeeds (u. a. für Webartikel und Podcasts): [RSS](#), [Atom](#)

XML-Anwendungen in der TR

- Mathematische Formeln: [MathML](#)
- 2D- und 3D-Vektorformate: [SVG](#) und [X3D](#)
- Print-Layout (PDF): [XSL-FO](#)
- InDesign: [IDML](#)
- E-Books: [EPUB](#) (ZIP-Container mit XML-Inhalt)
- Office-Formate (ZIP-Container mit XML-Inhalt):
Microsoft Office ab 2007 (Dateiformate [DOCX](#), [PPTX](#), [XLSX](#))
OpenOffice / LibreOffice (Dateiformate [ODT](#), [ODP](#), [ODS](#))

Einblick in ein E-Book

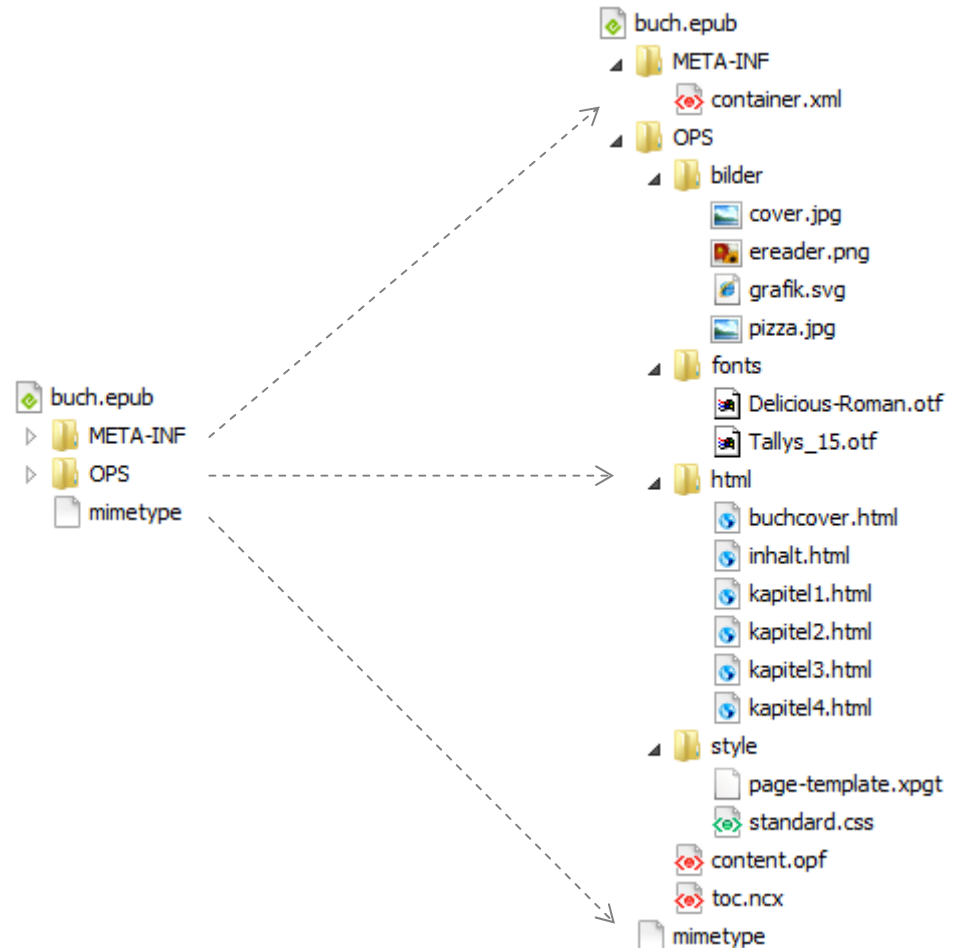
- ➔ EPUB-Format (ein Standard des IDPF, wird von vielen E-Ink-Lesegeräten und auf dem iPad über iBooks-App unterstützt)
- ➔ ZIP-gepacktes Container-Format mit Steuer- und Inhaltsdateien (XHTML), CSS zur Formatierung sowie Bildern und Schriftarten



Einblick in ein E-Book

→ EPUB-Struktur:

Zur automatisierten Produktion müssen alle nötigen Inhalts-Dateien erzeugt und mit den statischen Dateien (Bilder usw.) ZIP-gepackt und mit der Endung **.epub** abgelegt werden.



Diskografie-Projekt

- Als XSLT-Lehrprojekt zum Single-Source-Publishing (SSP) entwickelt und an anderer Stelle beschrieben
- XML-Struktur zur Erfassung aller Werke **einer** Band (mit DTD und Schema)
- Titel, Beschreibungen, Kommentare und Coverbilder werden erfasst
- 15 Ausgabeformate bzw. Nutzungsvarianten stehen zur Verfügung
- Insbesondere bei den genannten Containerformaten (CHM, DOCX, EPUB) und bei der PDF-Generierung lässt sich das SSP-Konzept anschaulich demonstrieren.

Diskografie-Projekt



```
<?xml version="1.0" encoding="UTF-8"?>
<diskografie xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="diskografie.xsd" bandname="Einstürzende Neubauten">
  <bandinfo>Einstürzende Neubauten ist eine deutsche experimentelle Band. ...</bandinfo>
  <referenzen><referenz>...</referenz><referenz url="...">...</referenz></referenzen>
  <werk jahr="1987" typ="LP" bewertung="9">
    <werkname>Fünf auf der nach oben offenen Richterskala</werkname>
    <kommentar>Der Song Morning Dew ist eine Coverversion ...</kommentar>
    <coverbild>en_richterskala.jpg</coverbild>
    <titelliste>
      <titel spielzeit="8:01">Zerstörte Zelle</titel><!-- weitere titel-Elemente -->
    </titelliste>
  </werk><!-- weitere werk-Elemente -->
</diskografie>
```

Diskografie-Projekt

➔ Einige Ausgaben:

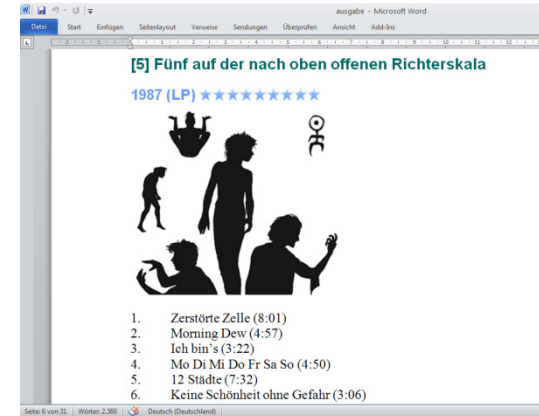
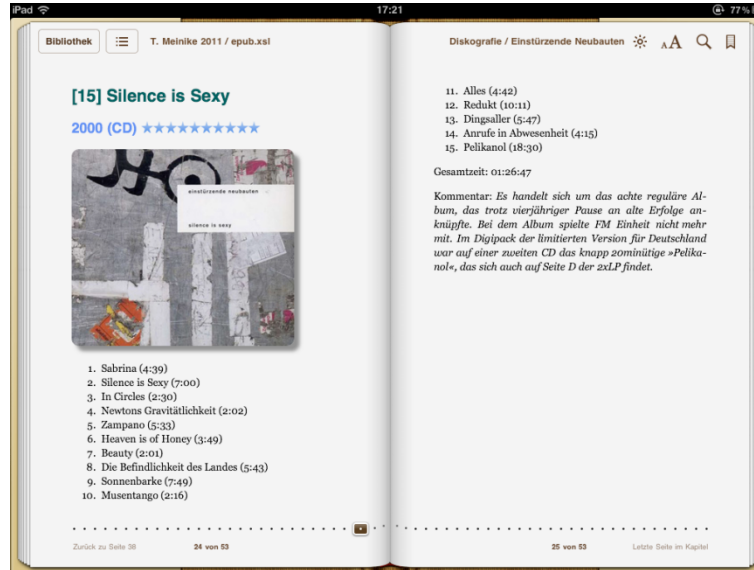
Diskografie | Einstürzende Neubauten

Strategies Against Architecture III (1991 – 2001)

2001 (DCD) ★★★★★★



1. Zentrifuge
2. 12305(te) Nacht
3. Für wen sind die Blumen?
4. Ende Neu (Live)
5. Redukt (Live)
6. Blume (French Version)
7. Three Thoughts (Devils Sect)
8. Implusion
9. Scampi alla Carina
10. Snake
11. Alles was irgendwie nützt (Live)
12. The Garden
13. Anrufe in Abwesenheit
14. Querschlanten
15. Architektur ist Geiselnahme
16. Helium
17. Wüste (Ballet Version)
18. Der leere Raum
19. Was ist ist (Extended Version)
20. I wish this would be your Colour (Live)
21. Bill Rubin
22. Die Interimliebenden
23. Installation Nr. 1 (John Is Mixing)
24. Montblanc / Open Fire
25. Salamandrina
26. Letztes Bild
27. Silence Is Sexy
28. Drachen



Seite 20 | 32

	A	B	C	D	E
1	Nr.	Jahr	Werktitel	Typ	Songs
2	1	1981	Kollaps	LP	14
3	2	1983	Zeichnungen des Patienten O.T.	LP	13
4	3	1984	Strategies Against Architecture (1980 – 1983)	LP	14

{Praktische Demonstration ...}

Zusammenfassung und Ausblick

- ➔ XML und zugehörige Konzepte haben sich in der Informations- und Wissensgesellschaft fest etabliert.
- ➔ Das Anwendungsspektrum reicht von der strukturierten Ablage von Informationen, über im Alltag genutzte Dateiformate bis zu komplexen, automatisierten Publikationsszenarien.
- ➔ XML-Technologien lassen sich bereits mit einfachen Mitteln studieren und ausprobieren.
- ➔ Um letztlich attraktive Medienprodukte zu produzieren, ist neben den theoretischen Grundlagen viel Entwicklerpraxis nötig.

Quellen

- Mitteldeutsche Zeitung vom 16. Juni 2012
<http://mz-web.de/a?id=1338485304372>
- Meine Mensa: <http://meine-mensa.de/speiseplan>
- Meinike, T.:
Diskografie-Projekt (XSLT-Tutorial, 2011) und Material zu diesem Vortrag:
<http://www.iks.hs-merseburg.de/~meinike/vortraege.php>
- XML-Editoren (Auswahl, **frei** / **kommerziell**):
 - XML Spy**: <http://www.altova.com/de/>
 - Editix Lite**: <http://www.freexmleditorsite.com/>
 - <oXygen/> XML Editor**: <http://www.oxygenxml.com/>
 - XML Copy Editor**: <http://xml-copy-editor.sourceforge.net/>