pagina
Publikations
technologien

AUSZEICHNUNG AKADEMISCHER LITERATUR IN DOCBOOK

22.04.2020 / XML User Group Stuttgart: XUGS #24

Kai Weber / @fruehlingstag

pagina
Publikations
technologien

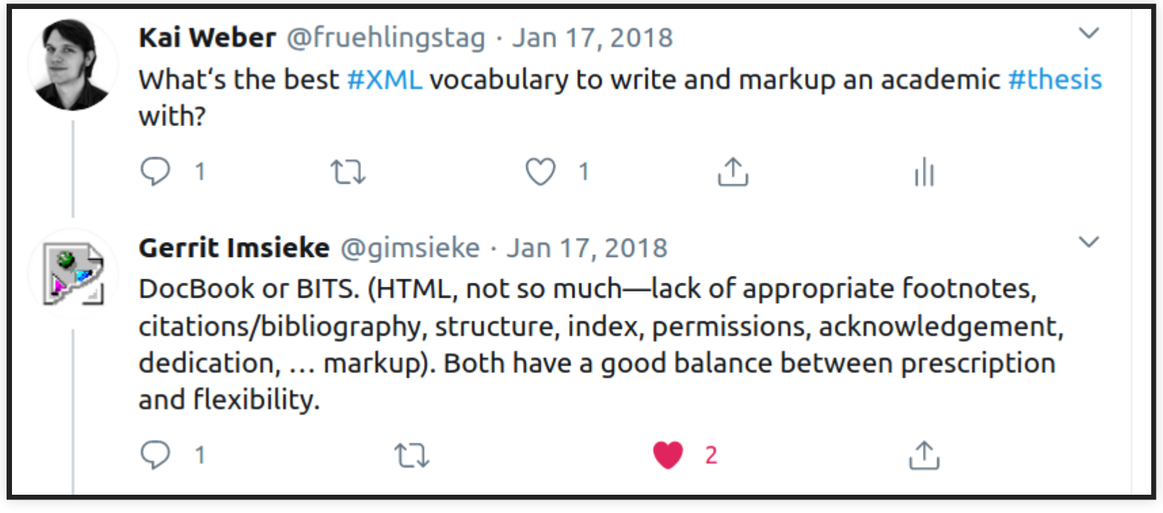
pagina Publikations technologien

Ausgangslage

- Universität stellte je ein MS-Word- und ein LaTeX-Template für die Formatierung der Abschlussarbeit bereit
- MS-Word kam für den Examinanden nicht in Frage, weil er unter Linux arbeitete
- LaTeX-Template kam für den Examinanden nicht in Frage, weil er mit dem Ökosystem aus Tools und Modulen nicht vertraut war
- Examinand fühlte sich in XML wohl

pagina
Publikations
technologien

Der Hilferuf



pagina Publikations technologien

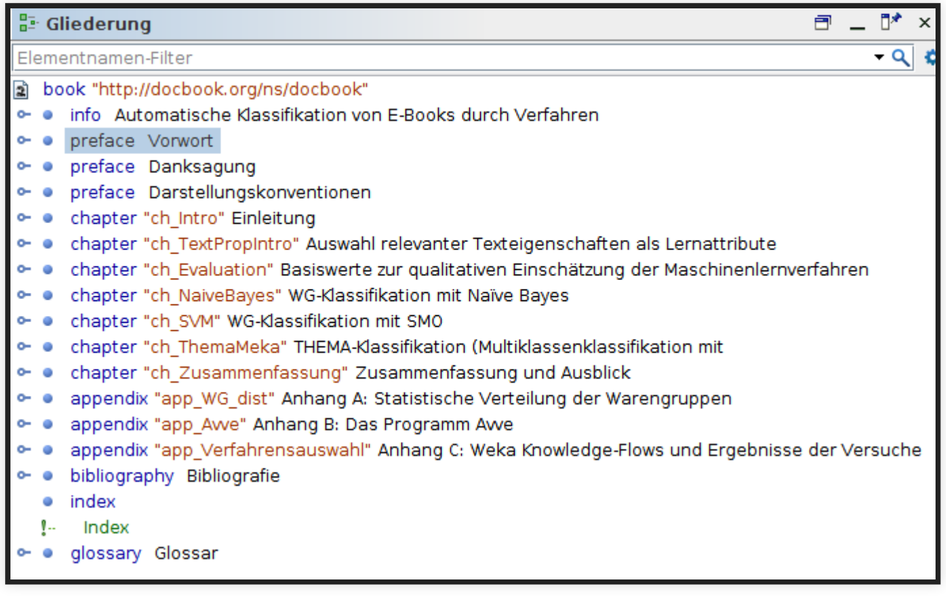
Erste Orientierung

Typische Elemente einer Buchstruktur sind schnell ermittelt und intuitiv verwendbar:

- <book> als Wurzelelement
- Metadaten / bibliogr. Daten in <info>
- allgemeine Kapitel heißen <chapter>
- es gibt Elemente für spezielle Kapitel, deren Benennung selbsterklärend ist, z. B. <preface>, <appendix>, <bibliography>, <index>, <glossary>

pagina Publikationstechnologien

Buchstruktur in <oXygen/>-Gliederungsansicht



pagina Publikations technologien

Binnenstruktur von Kapiteln

- wird mit ebenfalls intuitiv handhabbaren Elementen wie <sect1>, <sect2>, ..., <para> ausgezeichnet
- Einbindung von Abbildungen mit <figure>
- Aufbau von Tabellen mit <table>

pagina Publikations technologien

Systematische Datenpflege für Anhänge und Verlinkungen

- Viele akademische Arbeiten und Fachbücher verfügen über Anhänge mit bibliografischen Angaben, Glossaren und Registern
- Einige Verzeichnisse werden aus dem Haupttext generiert (z. B. Inhalts- und Abbildungsverzeichnis, Register)
- Andere Daten werden als Verzeichnisse gepflegt, auf die dann aus dem Haupttext verwiesen wird: Bibliografie, Glossar
- Das DocBook-Ökosystem unterstützt Verwaltung solcher Verzeichnisse inklusive einer Verlinkung

pagina Publikations technologien

Bibliografien in DocBook

- Werden aus bibliografischen Einträgen (<biblioentry> bzw. <bibliomixed>) aufgebaut
- <biblioentry>: strukturierte Erfassung als Datensatz; Renderingregeln (XSLT) bringen später den Datensatz in eine Darstellungsform
- <bibliomixed>: halbstrukturierte Erfassung als Text, der schon eine Darstellung in der gerenderten Publikation vorgibt
- Verschiedene Publikationsarten (selbständig, Zeitschriftenartikel, Kapitel in Sammelband, ...) werden über @role-Attribute unterschieden

pagina Publikationstechnologien

Beispiel Bibliografie (1)

bibliografischer Eintrag in strukturierter Form

```
<biblioentry role="monograph" xml:id="bib_ABC14">
    <abbrev>ABC14</abbrev>
    <editor><personname><firstname>Thomas</firstname><surname>Bez</surname></personname></editor>
    <title xml:lang="de">ABC des Zwischenbuchhandels</title>
    <edition>7. Auflage</edition>
    <publisher>
        <publishername>Börsenverein des Deutschen Buchhandels</publishername>
        <address><city>Frankfurt am Main</city></address>
    </publisher>
    <pubdate>2014</pubdate>
    <pagenums>50 S.</pagenums>
</biblioentry>
```

Ausgabe in PDF:

[ABC14] BEZ, Thomas (ed.). *ABC des Zwischenbuchhandels*. 7. Auflage. Frankfurt am Main: Börsenverein des Deutschen Buchhandels, 2014. 50 S.

Beispiel Bibliografie (2)

bibliografischer Eintrag in semi-strukturierter Form

```
<bibliomixed role="monograph" xml:id="bib_ABC14mixed">
    <abbrev>ABC14</abbrev> <editor><personname>BEZ, Thomas</personname></editor> (Hg.):
        <title>ABC des Zwischenbuchhandels</title>, 7. Auflage. Frankfurt am Main: Börsenverein
        des Deutschen Buchhandels, <pubdate>2014</pubdate>. <pagenums>50 S.</pagenums>
</bibliomixed>
```

Ausgabe in PDF:

Bibliografie

[ABC14] BEZ, Thomas (Hg.): ABC des Zwischenbuchhandels, 7. Auflage. Frankfurt am Main: Börsenverein des Deutschen Buchhandels, 2014. 50 S.

pagina Publikations technologien

Verlinkung auf Bibliografie aus dem Haupttext

Auf einen bibliografischen Eintrag verweist man aus dem Haupttext mit <biblioref>, z. B.:

```
<para>Die endgültige Fassung wurde 2007 zum Branchenstandard erhoben und seither von allen
Branchenteilnehmern verwendet (<biblioref linkend="bib_ABC14"/>, S. 48/49).</para>
```

Beim Rendern nach PDF wird anstelle des Verweises die Abkürzung des Eintrags aus dem <abbrev>-Element eingesetzt (und falls keine Abk. vorhanden, dann stattdessen die @xml:id):

gruppensystematik für den Buchhandel entwickelt. Die endgültige Fassung wurde 2007 zum Branchenstandard erhoben und seither von allen Branchenteilnehmern verwendet ([ABC14], S. 48/49). Das Verzeichnis Lieferbarer Bücher (VLB) machte die Angabe von genau einer Waren-

pagina Publikations technologien

Glossare in DocBook

- Werden aus dem Blockelement <glossentry> aufgebaut und bei Bedarf durch <glossdiv> zu größeren Einheiten gruppiert.
- Binnenaufteilung eines Glossareintrags in Term und Definition ist intuitiv verständlich (s. Bsp.)

```
<glossentry xml:id="gt_imprint">
  <glossterm>Imprint</glossterm>
  <glossdef>
    <para>Eine Verlagsmarke, die entweder keine rechtlich selbständige Firma ist oder als Firma zu einer Verlagsgruppe (Konzern) gehört, d
       im Markt als Verlag auftritt, z. B. Penguin Deutschland als Teil der Verlagsgruppe Random House.</para>
  </glossdef>
</glossentry>
```

pagina Publikationstechnologien

Glossarverweise

- `<glossterm linkend="IDREF">Glossartext</glossterm>`
- Standardstylesheet erzeugt bei Ausgabe von PDF hier einen Link aus dem Text in das Glossar
- Viele PDF-Reader haben leider keinen Zurückbutton wie Browser, so dass es mitunter schwierig, vom Glossar (oder vom bibliogr. Eintrag) zurück in den Text zu springen

pagina Publikations technologien

Register in DocBook

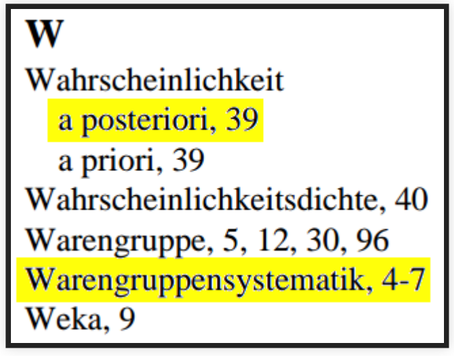
- Register können vom Renderingsystem automatisch erstellt werden, wenn Registerbegriffe im Haupttext entsprechend ausgezeichnet werden
- Registereinträge können sich auf einzelne, zusammenhängende Textstellen oder auch auf ganze Kapitel oder beziehen
- Registereinträge können bis zu drei Stufen tief gestaffelt werden

pagina Publikations technologien

Registerbegriffe, zwei Varianten

- ```
  Gesucht wird die <indexterm><primary>Wahrscheinlichkeit</primary><secondary>a posteriori</secondary>
  </indexterm>A-posteriori-Wahrscheinlichkeit, dass [...]
  ```

- ```
  Die Warengruppensystematik <indexterm class="startofrange" xml:id="idt_002"><primary>Warengruppensystematik</primary></indexterm>
  des Deutschen Buchhandels [...] Mit uneindeutigen Grenzfällen muss somit sowohl bei der automatischen Klassifikation als auch
  bei der manuellen Eingruppierung stets gerechnet werden.<indexterm class="endofrange" startref="idt_002"/>
  ```



W

Wahrscheinlichkeit
  a posteriori, 39
  a priori, 39
Wahrscheinlichkeitsdichte, 40
Warengruppe, 5, 12, 30, 96
Warengruppensystematik, 4-7
Weka, 9

pagina Publikationstechnologien

Mathematische Formeln in DocBook

- Je nach DocBook-Version oder -Dialekt ist Verwendung von MathML innerhalb von <equation> und <inlineequation> schon via Relax-NG-Schema erlaubt.
- Bei Bedarf lässt sich leicht eine eigene Schema-Anpassung erstellen, die MathML auch in anderen Elementen erlaubt (vgl. z. B. Anleitung auf www.data2type.de
- oXygen unterstützt sowohl die Anzeige integrierter Formeln im Autormodus als auch das Rendern nach PDF

Hamming Loss $\frac{1}{\mathrm{tn}}\sum_{i=1}^{t}\mathrm{hd}\left(y_i, \hat{y}_i\right)$

Formel in oXygen Author

$$\frac{1}{\mathrm{tn}}\sum_{i=1}^{t}\mathrm{hd}\left(y_i, \hat{y}_i\right) \qquad (6.1)$$

Formel in PDF-Ausgabe

Schema-Anpassung (1)

Dem DocBook Technical Committee als Standardisierungsgremium ist es wichtig, dass DocBook durch Anwender*innen an eigene Bedürfnisse angepasst werden kann. Um dies zu erleichtern, hat man sich bewusst für RELAX NG + Schematron entschieden und beim Modellieren auf leichte Erweiterbarkeit geachtet.

Schema-Anpassungen behalten den DocBook-Namensraum bei, spezifizieren aber einen eigenen Versionswert. Das Technical Committee empfiehlt dazu folgendes Benennungsmuster: *(subset|extension|variant) (name[-version])+*. Beispiel:

```
<book xmlns="http://docbook.org/ns/docbook"
xmlns:mml="http://www.w3.org/1998/Math/MathML"
xmlns:xlink="http://www.w3.org/1999/xlink" xml:lang="de"
version="5.0-extension Avve-1.0">
```

pagina Publikations technologien

Schema-Anpassung (2)

Ein RELAX-NG-Schema passt man an, indem man zunächst das zu erweiternde (oder einzuschränkende) Schema inkludiert:

```
namespace db = "http://docbook.org/ns/docbook"
# evtl. weitere Namespace-Deklarationen

include "docbook.rnc" {
   # Neudefinition von DocBook-Mustern, v. a. für Einschränkungen
}

# neue und erweiterte Muster (zusätzliche Attribute und Elemente)
```

pagina Publikations technologien

Schema-Anpassung, Anwendungsbeispiel: Ziel

- In meiner Arbeit wollte ich einige Variablen verwenden für Zahlwerte, die an mehreren Stellen im Text genannt werden, die ich aber in einer separaten Tabelle pflegen wollte, um im Falle einer Aktualisierung der Zahl nicht alle Textstellen suchen zu müssen, an der die Zahl genannt oder behandelt wird.
- Pro Variable definierte ich ein leeres XML-Element, das ich als Inline-Element in beliebigen Text einsetzen konnte
- Das Rendering (in meinem Fall: XSL-FO) sollte dann beim Erzeugen der Ausgabe die Variablen-Elemente durch die in der separaten Tabelle hinterlegten Werte ersetzen (ggf. auch aus den Werten per Formel abgeleitete Zahlen berechnen)

pagina Publikations technologien

Schema-Anpassung, Anwendungsbeispiel: RELAX NG

```
# custom avve elements
db.domain.inlines |= db.lucene.numberOfDocuments
db.domain.inlines |= db.lucene.numberOfTerms
db.domain.inlines |= db.lucene.numberOfUniqueTerms
db.domain.inlines |= db.modifiedInstancesRatio
db.domain.inlines |= db.numberOfClasses
db.domain.inlines |= db.numberOfTrainingInstances

db.lucene.numberOfDocuments = element numberOfLuceneDocuments {
    empty
}
db.lucene.numberOfTerms = element numberOfLuceneTerms {
    empty
}
[...]
```

pagina Publikations technologien

Schema-Anpassung, Anwendungsbeispiel: Verwendung

DocBook-XML:

```
<para>[...] In einem Bestand von <numberOfLuceneDocuments/> Dokumenten mit insgesamt <numberOfLuceneTerms/>
unterschiedlichen Lemmata betrug die Zahl der Lemmata, welche nur in einem Dokument
vorkommen, <numberOfUniqueLuceneTerms/>. [...] <para>
```

PDF-Ausgabe:

dann, wenn es keine Klassen gibt, in deren Trainingsmenge sich nur ein einzelnes Textdokument befindet. In einem Bestand von 6927 Dokumenten mit insgesamt 2.971.631 unterschiedlichen Lemmata betrug die Zahl der Lemmata, welche nur in einem Dokument vorkommen, 1.852.691. Ein Ausschluss dieser Lexeme aus der Wortliste kann also die Menge der für den endgültigen

pagina Publikations technologien

DocBook-XSL anpassen

- oXygen hat bereits optimierte DocBook-Transformationsszenarien eingebaut
- Diese kann man wiederum als Basis für eigene Erweiterungen nutzen
- Wie auch bei der Schema-Erweiterung importiert man zunächst das Basis-XSLT, das man erweitern möchte:

```
<!-- in oXygen this path is translated to a local framework file via XML catalog -->
<xsl:import href="http://cdn.docbook.org/release/xsl/current/fo/docbook_custom.xsl"/>
```

pagina Publikations technologien

Anwendungsbeispiel (1)

```
<xsl:param name="lucene.numberOfDocuments" select="6927"/>
<xsl:param name="lucene.numberOfTerms" select="2971631"/>
<xsl:param name="lucene.numberOfUniqueTerms" select="1852691"/>

<xsl:template match="db:numberOfLuceneDocuments">
  <xsl:value-of select="$lucene.numberOfDocuments"/></xsl:template>
<xsl:template match="db:numberOfLuceneTerms">
  <xsl:value-of select="format-number($lucene.numberOfTerms, '###.###', 'european')"/></xsl:template>
<xsl:template match="db:numberOfUniqueLuceneTerms">
  <xsl:value-of select="format-number($lucene.numberOfUniqueTerms, '###.###', 'european')"/></xsl:template>
```

pagina Publikations technologien

Anwendungsbeispiel (2)

Das XSLT kann natürlich auch Werte zum Renderingzeitpunkt berechnen:

```
<xsl:template match="db:randomClassifierSuccessRate">
    <xsl:value-of select="
      format-number(
        sum(
            for $i in $avve.warengruppenDistribution/warengruppen/warengruppe
            return (($i/@traininstances + $i/@testinstances) div ($avve.numberOfTrainingInstances + $avve.numberOfTestInstances))
                * (($i/@traininstances + $i/@testinstances) div ($avve.numberOfTrainingInstances + $avve.numberOfTestInstances))
        ),
      '#0,0000%',
      'european')" />
</xsl:template>
```

pagina Publikations technologien

Anwendungsbeispiel (3)

Duch ein entsprechendes Template habe ich ganze Tabellen auf Basis von Variablen im XSLT dynamisch berechnet:

```
<xsl:template match="db:warengruppenDistribution">
    <xsl:variable name="temp.warengruppenTable">
        <db:table xml:id="temp.warengruppenTable" pgwide="0">
            [Definition DocBook-Tabelle gekürzt]
            <xsl:for-each select="$avve.warengruppenDistribution/warengruppen/warengruppe">
                <xsl:sort select="@traininstances + @testinstances" order="descending" />
                <db:row>
                    [...]
                    <db:entry><xsl:value-of select="format-number((@traininstances + @testinstances) div
                        ($avve.numberOfTrainingInstances + $avve.numberOfTestInstances), '0,00%', 'european')"/></db:entry>
                </db:row>
            </xsl:for-each>
    </xsl:variable>
    <xsl:apply-templates select="$temp.warengruppenTable" mode="warengruppe"/>
</xsl:template>
```

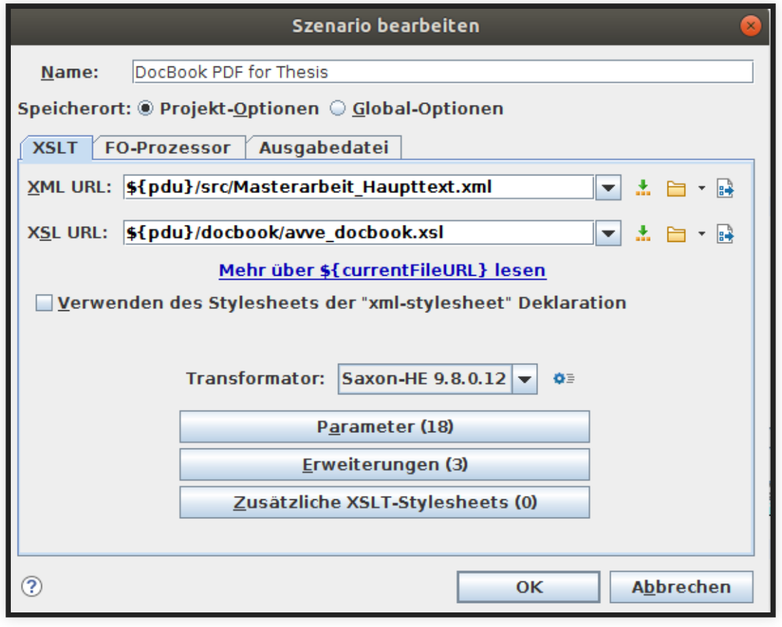
Anwendungsbeispiel (4): Dynamisch ausgegebene Tabelle

Table 1. Warengruppenverteilung im Korpus

Warengruppencode	Text	Anteil im Korpus
112	erzählende Gegenwartsliteratur (ab 1945)	17,90%
121	Krimis, Thriller, Spionage	8,30%
260	Jugendbücher ab 12 Jahre	8,00%
250	Kinderbücher bis 11 Jahre	4,06%
132	Fantasy	3,89%
481	Ratgeber Lebensführung, persönliche Entwicklung	3,72%

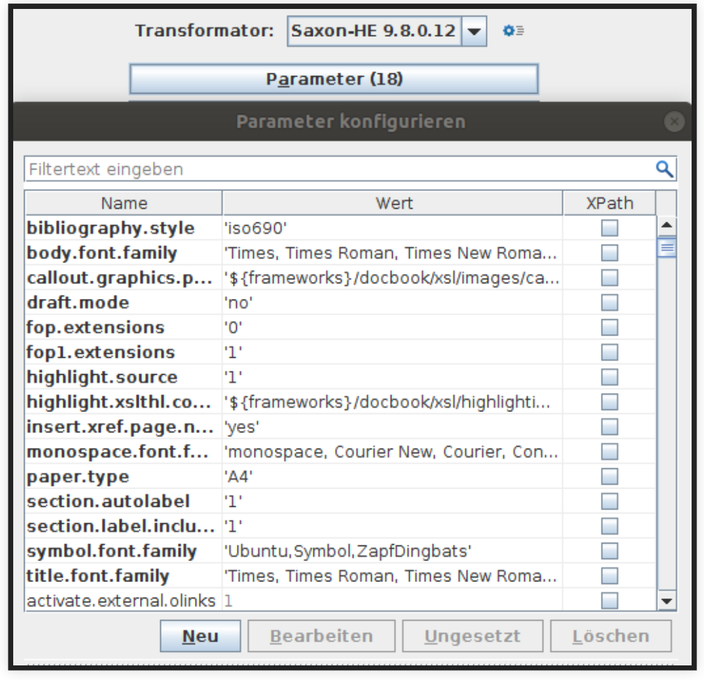
Oxygen-Transformationsszenario anlegen

- Für die Anpassungen am DocBook-XSL empfiehlt es sich, ein oXygen-Transformationsszenario anzulegen
- Dazu dupliziert man zunächst das eingebaute Szenario „DocBook PDF“ und trägt sein eigenes XSLT-Skript ein



DocBook-XSL-Parameter definieren (1)

- Die DocBook-XSL-Skripte bieten auch ohne Programmierung eigener Templates eine Vielzahl von Parametern, durch deren Belegung man das Verhalten der Ausgabe steuern kann.
- Am einfachsten lassen sich diese Parameter über den „Parameter“-Button im oXygen-Transformationsszenario setzen



pagina Publikations technologien

DocBook-XSL-Parameter definieren (2)

Verwendet man ein eigenes/angepasstes XSL-Stylesheet, kann man dort auch Defaultwerte des importierten DocBook-XSL-Skriptes überschreiben, z. B.:

```
<!-- DocBook parameters -->
<xsl:param name="paper.type" select="'A4'"/>
<xsl:param name="page.margin.inner" select="'2.5cm'"/>
<xsl:param name="page.margin.outer" select="'2.5cm'"/>
<xsl:param name="page.margin.top" select="'1.25cm'"/> <!-- margin from top edge of page to top of header -->
<xsl:param name="region.begore.extend" select="'0.75cm'"/> <!-- height of the header -->
<xsl:param name="body.margin.top" select="'1.5cm'"/> <!-- top of the text body, as calculated from margin.top -->
<xsl:param name="page.margin.bottom" select="'2cm'"/> <!--margin from bottom edge of page to bottom of footer -->
<xsl:param name="body.start.indent" select="'0cm'"/> <!-- switch off DocBook's default indenting -->
<xsl:param name="body.font.family" select="'Times Roman'"/>
<xsl:param name="body.font.master" select="12"/> <!-- master font size -->
<xsl:param name="header.column.widths">1 4 1</xsl:param> <!-- give more space to the running header title -->
```

pagina Publikations technologien

Schluss

- Fragen?
- Zum Vertiefen und eigenen Nachvollziehen der hier vorgestellten Anpassungen kann man mein oXygen-Projekt mit den gezeigten Anpassungen klonen: https://github.com/sermo-de-arboribus/db-thesis
- Im genannten Repo liegen auch die Quellen dieser Präsentationsfolien. Eine PDF der Folien wird in den nächsten Tagen auf www.xugs.de bereitgestellt
- Vielen Dank!