

# DocBook – Eine Einführung

Tobias Heidelbergmann

14.05.2007

# DocBook – Was ist das?

- Auszeichnungsformat, das XML-konform, bzw. SGML-konform definiert ist
- Verwaltet vom DocBook Technical Committee von OASIS (Organization for the Advancement of Structured Information Standards)
- Kern von DocBook ist Sammlung von DTDs (Norm) für XML
- auch als XML-Schema, Relax-Ng, SGML-Schema
- erste Version 1991
- aktuelle Version: 4.5
- ab Version 5.0 ist Relax-Ng die Norm

Nutzung zur Auszeichnung von:

- Büchern (auch mehrbändige Werke)
- Artikeln
- Präsentationen
- ursprünglich: Unix-Dokumentation
- heute: hohe Flexibilität durch Vielzahl an Notationselementen

# DocBook – Wo ist es erhältlich?

- [www.docbook.org](http://www.docbook.org)
- In den meisten Linux-Distributionen enthalten
- Zusätzlich zu DTDs: Sammlung von Stylesheets in XSL
- Aber auch in anderen Stylesheet-Beschreibungssprachen
- Für gängige Ausgabeformate (html, xhtml, pdf, ...)

## Dokumentation:

- „DocBook – The Definitive Guide“, Norman Walsh und Leonard Muellner, O'Reilley
- Einführung und Referenz
- verfügbar online (html, pdf, chm)
- auch als Debian-Paket
- ist aber „Work in Progress“, d.h. insbesondere für Version 5.0 noch nicht vollständig

# (grobe) Kategorisierung der Elemente

## - typische Wurzelemente -

<set>

- Verwendet für z.B. mehrbändige Buchreihen oder andere „Sets“
- Spitze der DocBook-Hierarchie (d.h. es gibt kein Element außer <set>, das <set> als Kind enthalten kann), aber optional

- Notation im „Content-Model“:

set ::= (

(title, subtitle?, titleabbrev?)?,  
setinfo?,  
toc?,  
(set|book)+,  
setindex?)

- Notationselement ‘,’: Sequenz
- Notationselement ‘?’: min = 0, max = 1
- Notationselement ‘+’: min = 1, max = unbounded
- Notationselement ‘|’: OR

# Kategorisierung der Elemente

## - typische Wurzelemente -

<book>

- gebräuchlichstes Wurzelement in Dokumenten
- book ::= (  
    (title, subtitle?, titleabbrev?)?,  
    bookinfo?,  
    (dedication | toc | lot | glossary | bibliography |  
preface | chapter | reference | part | article | appendix  
| index | setindex)\*)  
)

# Kategorisierung der Elemente

## - weitere Kategorien -

- Aufteilungen (Divisions):
  - unterteilen <book> in weitere logische Einheiten
  - <part>
  - <reference> kann <refentry> enthalten (ursprünglich manpages)
- Komponenten:
  - Kapitelähnliche Unterteilungen von <book> oder <part>
  - <preface>
  - <chapter>
  - <appendix>
  - <bibliography>
  - <glossary>
  - Komponenten können ihrerseits Navigationskomponenten , Sektionen oder Blockelemente enthalten

# Kategorisierung der Elemente

## - weitere Kategorien -

- Navigationskomponenten:
  - Verzeichnisse aller Art
  - <toc> Table of Contents = Inhaltsverzeichnis
  - <lot> List of Titles = alle anderen Verzeichnisse, z.B. Tabellen-, Abbildungs-, Beispiel-, Formelverzeichnis
  - <index> Stichwortverzeichnis
  - Werden in der Regel automatisch vom Stylesheet erzeugt
- Sektionen
  - Unterteilen Komponenten (Kapitel) in Unterkapitel
  - <section> erlaubt beliebig tiefe Verschachtelung
  - <sect1>, ..., <sect5> Zahl gibt die Schachtelungstiefe an
  - <simplesect> erlaubt keine weitere Verschachtelung

# Kategorisierung der Elemente

## - weitere Kategorien -

- Blockelemente (absatzähnliche Strukturen)
  - Treten als Kinder von Komponenten (Kapitel) oder Sektionen (Unterkapitel) auf
  - `<para>` kann Text enthalten
  - `<figure>` Abbildung
  - Tabellen
  - Listen
  - Formatierungsbereiche (erhalten Zeilenumbrüche)
- Inline-Elemente
  - Auszeichnung einzelner Teile im fortlaufenden Text
  - `<quote>` Zitat
  - `<emphasis>` besondere Hervorhebung
  - `<citation>` bibliografische Referenz
  - Mathematische Elemente (`<superscript>` = Exponent)
  - ...



# Typische Kapitelstrukturierung

```
<chapter>
  <chapterinfo> ... </chapterinfo>
  <title> Das erste Kapitel </title>
  <toc/>
  <para> Eine Einführung in das erste Kapitel </para>
  <sect1>
    <title> Das erste Unterkapitel </title>
    <para> Inhalt des <emphasis>ersten</emphasis>
      Unterkapitels
    </para>
    <sect2> ... </sect2>
  </sect1>
  <sect1>
    <figure>...</figure>
    ...
  </sect1>
  ...
</chapter>
```

# Metainformationen

- dienen zur Beschreibung des Dokumentinhalts
- realisiert als \*info – Element innerhalb von Elterknoten, z.B.
  - <bookinfo>, <chapterinfo>, <sectioninfo>, ...
  - ab 5.0: einheitlich nur ein Info-Element <info>
- bookinfo ::= (
  - (abstract | address | author | authorgroup |  
authorinitials | copyright | edition | isbn |  
othercredit | publisher | pubdate | subtitle | title)+)

Die meisten Elemente wiederum zusammengesetzte Elemente

# Verbindung Content-Model zu XML-Schema am Beispiel <author>

```
author ::= [  
  {personname | (honorific | firstname | surname | lineage | othername |  
  affiliation | authorblurb | contrib)+ },  
  {personblurb | email | address }*  
]
```

```
author ::= [  
    {personname | (honorific | firstname | surname | lineage |  
othername |      affiliation | authorblurb | contrib)+ },  
    {personblurb | email | address }*  
    ]
```

```
<xs:element name="author" substitutionGroup="docinfo.char.class">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:choice>  
        <xs:element ref="personname"/>  
        <xs:group maxOccurs="unbounded" ref="person.ident.mix"/>  
      </xs:choice>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

```
author ::= [  
    {personname | (honorific | firstname | surname | lineage |  
othername |      affiliation | authorblurb | contrib)+ },  
    {personblurb | email | address }*  
    ]
```

```
<xs:choice minOccurs="0" maxOccurs="unbounded">
```

```
    <xs:element ref="personblurb"/>
```

```
    <xs:element ref="email"/>
```

```
    <xs:element ref="address"/>
```

```
</xs:choice>
```

```
</xs:sequence>
```

```
    <xs:attributeGroup ref="author.attlist"/>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
author ::= [
    {personname | (honorific | firstname | surname | lineage |
othername |          affiliation | authorblurb | contrib)+ },
    {personblurb | email | address }*
]
```

```
personname ::= (
    (honorific | firstname | surname | lineage | othername)+
)
```

```
<xs:element name="personname" substitutionGroup="gen.char.class">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="honorific"/>
      <xs:element ref="firstname"/>
      <xs:element ref="surname"/>
      <xs:element ref="lineage"/>
      <xs:element ref="othername"/>
    </xs:choice>
    <xs:attributeGroup ref="personname.attlist"/>
  </xs:complexType>
</xs:element>
```

```
author ::= [
    {personname | (honorific | firstname | surname | lineage |
othername |          affiliation | authorblurb | contrib)+ },
    {personblurb | email | address }*
]
```

```
personname ::= (
    (honorific | firstname | surname | lineage | othername)+
)
```

```
firstname ::= (
    #PCDATA | link | olink | ulink | emphasis | trademark | replaceable
|          remark | subscript | superscript | inlinegraphic |
inlinemediaobject |          indexterm)*
```

```
<xs:element name="firstname">
  <xs:complexType mixed="true">
    <xs:group minOccurs="0" maxOccurs="unbounded" ref="docinfo.char.mix"/>
    <xs:attributeGroup ref="firstname.attlist"/>
  </xs:complexType>
</xs:element>
```

```

author ::= [
    {personname | (honorific | firstname | surname | lineage |
othername |          affiliation | authorblurb | contrib)+ },
    {personblurb | email | address }*
]

personname ::= (
    (honorific | firstname | surname | lineage | othername)+
)

firstname ::= (
    #PCDATA | link | olink | ulink | emphasis | trademark | replaceable
|          remark | subscript | superscript | inlinegraphic |
inlinemediaobject |          indexterm)*

```

- weiteres Dereferenzieren liefert die anderen Notationselemente
- insgesamt sehr komplexes Schema ( ~ 10.000 Zeilen XML Schema)
- liefert aber hohe Flexibilität zur individuellen Gestaltung von Dokumenten
- Aber: gültiges DocBook-Dokument muss nicht sinnvolle Daten enthalten:
- Bsp: <personname> kann aus (mehreren) <lineage> bestehen



# Ausgewählte Blockelemente

## - Listen -

- 7 Listentypen, hier Auswahl:
- Einfache Liste <simplelist>
  - Attribut type gibt genaue Art der Liste an
  - <member> kennzeichnet Listenelemente
  - type = 'inline' bedeutet Aufzählung im Textfluss (A, B, C, D, ...)
  - type = 'horiz'  
bedeutet tabellenartige Struktur  
columns gibt Anzahl der Spalten an  
in der die Listenelemente v.l.n.r. und v.o.n.u. eingetragen werden.  
type = 'vert' analog.

```
<simplelist type='horiz' columns='1'>  
  <member>Eintrag 1</member>  
  <member>Eintrag 2</member>  
</simplelist>
```

# Ausgewählte Blockelemente

## - Listen -

- Geordnete Liste <orderedlist>
  - Kann <title> enthalten
  - Elemente werden als <listitem> gekennzeichnet
  - Inhalt von <listitem> kann jedes Blockelement sein
  - Attribut „spacing“ bestimmt Abstand zwischen Einträgen
  - Attribut „numeration“ bestimmt Art der Nummerierung (mögliche Werte z.B. ´arabic´, ´loweralpha´, ´upperroman´)
  - Schachtelung möglich (Attribute „continues“, „inheritnum“)

```
<orderedlist numeration=´upperroman´>  
  <title>Nummerierte Liste</title>  
  <listitem><para>Eintrag 1</para></listitem>  
  <listitem><para>Eintrag 2</para></listitem>  
</orderedlist>
```

# Ausgewählte Blockelemente

## - Abbildungen / Medienelemente -

- `<figure>` benötigt Titel und erscheint als Eintrag im Abbildungsverzeichnis
- `<informalfigure>` benötigt keinen Titel, erscheint nicht im Abbildungsverzeichnis
- Inhalt von `<figure>` kann u.a. `<mediaobject>` sein
  - Kann enthalten: Metainformationen `<objectinfo>`
  - Medienobjekt `<imageobject>`, `<videoobject>`, `<audioobject>`, ...
    - Diese enthält `<***data fileref='xyy.jpg' format='JPG'>`
    - Formatierungsanweisungen über Attribute spezifiziert (z.B. `align`, `valign`, ...)

```
<figure>
```

```
  <title> Abbildungstitel </title>
```

```
  <mediaobject><imageobject>
```

```
    <imagedata fileref='image/xyz.jpg' format='JPG' />
```

```
  </imageobject></mediaobject>
```

```
</figure>
```

# Ausgewählte Blockelemente

## - Tabellen -

- CALS – Tabellenmodell (ein Standard)
- `<table>` oder `<informaltable>`
- `<table>` benötigt Titel und erscheint im Tabellenverzeichnis
- Darstellung über Attribute gesteuert:
  - `colsep` steuert die Abgrenzung von Spalten durch Linien (`colsep=´1´`)
  - `rowsep` steuert die Abgrenzung von Zeilen durch Linien
  - `frame` steuert die Umrandung der Tabelle (Werte: `top`, `all`, `bottom`, ..)
- Enthält mindestens ein Element `<tgroup>` (Container für Tabelleninhalt)
- `<tgroup>` enthält obligatorisch `<tbody>`, optional `<thead>`, `<tfoot>`
- Zeilen werden innerhalb von `<tbody>` über `<row>` angezeigt
- Spalten über eine Folge von `<entry>` Elementen innerhalb von `<row>`
- `<entry>` enthält normale Blockelemente oder Text

# Ausgewählte Blockelemente

## - Tabellen -

```
<table frame='none'> <title>Tabellentitel</title>
  <tgroup>
    <tbody>
      <row>
        <entry>Eintrag 1</entry>
        <entry>Eintrag 2</entry>
      </row>
      <row>
        <entry>Eintrag 3</entry>
        <entry>Eintrag 4</entry>
      </row>
    </tbody>
  </tgroup>
</table>
```

# Ausgewählte Blockelemente

## - Tabellen -

- Ebenfalls möglich:
  - Textausrichtung einzelner Zellen definieren (Attribut „align“)
  - Zeilenübergreifende Zellen: Attribut „morerows“  
<entry morerows='2'>
  - Spaltenübergreifende Zellen:  
Identifizierung der Spalten durch <colspec colname='A'>, ...  
Spaltenübergreifenden Eintrag mit Attributen versehen (namest, nameend)

```
<table frame='none'> <title>Tabellentitel</title>
  <tgroup>
    <colspec colname='A' /><colspec colname='B' />
    <tbody>
      <row>
        <entry namest='A' nameend='B'>Eintrag 1</entry>
      </row>
    </tbody>
  </tgroup>
</table>
```

# Ausgewählte Komponenten

## - Bibliografie -

- Daten werden „roh“ ausgezeichnet, d.h. Autor präsentiert Datensatz  
Stylesheet legt Interpunktion fest und stellt möglicherweise nur einen Teil des Datensatzes dar.
- Daten werden „vorgekocht“ ausgezeichnet, d.h. Interpunktion wird von Dokumentautor vorgegeben und gesamter Datensatz wird angezeigt.
- Hier: roh
- `<bibliography>` enthält :
  - `<title>` optional
  - `<bibliographyinfo>`
  - `<biblioentry>` enthält Publikationsdaten, hier Auswahl möglicher Elemente:

```
biblioentry ::= (  
    (author | authorgroup | edition | isbn | issuenum | pagenums |  
    pubdate | publisher | subtitle | title | volumenum)+  
)
```

# Ausgewählte Komponenten

## - Bibliografie -

- Für Referenzierung innerhalb des Dokuments:
- `<biblioentry id='eindeutig' xreflabel='Pub01'>...</biblioentry>`
- Innerhalb des Dokuments referenziert durch:  
`<citation><xref linkend='eindeutig' /></citation>`
- In formatiertem Text erscheint Referenz z.B. als [Pub01]



# Ausgewählte Komponenten

## - Index -

- `<index>`
  - Kann `<title>`, `<para>`, ... Enthalten
  - Einträge: Stichpunkt, Seite(nbereich)
- a) Markierung einer einzelnen Seite als Sprungziel für einen Stichpunkt
- Sprungziels durch Inline-Element `<indexterm>` kennzeichnen.
  - Spezifikation des Stichpunkts durch `<primary>`, .., `<tertiary>` (Einrückungen)

`<para>`

```
DocBook<indexterm><primary>XML</primary><secondary>DocBook</secondary>  
></indexterm> ist angewandtes XML.
```

`</para>`

Eintrag:

XML

DocBook, S. xyz

# Ausgewählte Komponenten

## - Index -

b) Markierung eines ganzen Bereichs als Sprungziel

- Start des Bereichs: `<indexterm id='eindeutig' class='startofrange'>`  
eindeutige Identifikation mittels id-Attribut  
class attribut definiert Bereichsstart
- Ein weiteres `<indexterm startref='eindeutig' class='endofrange' />` Element  
markiert Ende des Bereichs  
Referenz auf Start des Bereichs durch startref  
class Attribut definiert Bereichsende

```
<para>
```

```
DocBook <indexterm id='dbook' class='startofrange'>  
<primary>XML</primary><secondary>DocBook</secondary></indexterm> ist  
angewandtes XML.
```

```
</para>
```

```
<para>...</para>...
```

```
<indexterm startref='dbook' class='endofrange' />
```

# Literatur

- „DocBook – The Definitive Guide“, Norman Walsh und Leonard Muellner, O'Reilly
- „Dokumentationen mit DocBook-XML“, Lars Trieloff,  
<http://trieloff.net/docbook/tutorial/docbook.pdf>