PB138 — Docbook Markup

(C) 2018 Masaryk University -- Tomáš Pitner, Luděk Bártek, Adam Rambousek

DocBook as an example of a more complex markup

- big project, one complex markup for all programmmer's documentation
- now many other purposes — writing papers (`article`), books (`book`), chapters (`chapter`), sections (`section`, `sectX`)
- authored by Norman Walsh (formerly Sun Microsystems Inc.)
- details, DTD, help, software, styles, see docbook.org
- probably the biggest markup for technical documentation ever
- there is the TDG (DocBook: The Definitive Guide) — also as Windows Help

What is Docbook?

- Docbook is a XML (and SGML) markup for writing *documents*,
- namely of technical nature, eg. *computer/software manuals*, technical documentation.
- Originally as a tool to cope with large UNIX-systems documentation.
- In principle, DB is a *logical (semantic) markup* (i.e. visual representation is not of importance when writing the source).

Key structural elements

Text is created using semantic elements for:

big text blocks

`book`, `paper`, `chapter`, `section`, `paragraph`, `screen`...

in-line parts

`emphasized`, `link`, `productname`, `command`,...

multimedia elements

images, videos, sounds...

helper elements and *metadata*

`title`, `author`, date of creation, copyright, index items, ToC...

Advantages of Docbook

Easy processing:

- *visualization* (using CSS, using XSLT for transformation to HTML, via LaTeX or XSL:FO to PDF, but also PostScript, PDF, RTF, DVI and plain-ASCII...), or documentation/help formats (HTML Help, Microsoft CHM, man-pages)

- selected parts or elements can be extracted separately (take the intro chapter, generate the book ToC…) or connect more texts into one

Origin

- Docbook since beginning of 90s (1991), as a SGML markup that time.
- After introduction of XML as de-facto standard for semistructured data (W3C spec. XML in 1998) is Docbook predominantly encoded in XML — mainly because of plethora of tools available.
- Further development under OASIS (The Organization for the Advancement of Structured Information Standards).
- Jirka Kosek is involved in the development, the editor of specifications is Norm Walsh.

Basic structures of Docbook

- Storing Docbook into files
- Elements in Docbook markup

Storing files

- Usual extension for files containing Docbook documents is `.dbk`, or simply `.xml`
- MIME type for Docbook is 'application/docbook`xml'

Document categories

The nature (purpose, size) of the document is mainly determines by using certain structural elements. The categories include:

set

collection of (`book`) or other collections — may be nested.

book

book containing chapters(`chapter`), papers (`article`) or parts (`part`), may contain indices (`index`), appendices (`appendix`) etc.

part

part containing one or more chapters, may be nested, may contain intro texts.

article

paper, may contain a sequence of block element (like chapters, paragraphs).

chapter

named and usually numbered section of a bigger document (`book`, `paper`).

appendix

příloha

dedication

dedication of a certain element

Block elements

- paragraphs (`para`)
- tables (`table`)
- lists (`itemizedlist`, `orderedlist`, `variablelist`)
- examples (`example`)
- figures (`figure`), etc.

These block elements are visualized in the order they will be read, ie. — top-down in Western languages, but left-right in Chinesse.

Inline elements

Inline elements are contained in block elements:

- emphasized text (`emphasis`…)
- links (eg. `link`, `ulink`, `olink`…) — we usually use `ulink` which is useful for internet addresses
- meaning (`keyword`, `command`, `filename`…)

Example of Docbook 5 document

Docbook 5 is the latest but still developed standard. It usesXML Namespacesand no DOCTYPE declaration.

```
<?xml version="1.0" encoding="UTF-8"?>
<book id="simple_book" xmlns="http://docbook.org/ns/docbook"
     version="5.0">
  <title>Very simple book</title>
  <chapter id="chapter_1">
    <title>Chapter 1</title>
    <para>Hello world!</para>
    <para>I hope that your day is proceeding <emphasis>splendidly</emphasis>!</para>
  </chapter>
  <chapter id="chapter_2">
    <title>Chapter 2</title>
    <para>Hello again, world!</para>
  </chapter>
</book>
```

The same in Docbook 4.4

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.4//EN"
               "http://www.oasis-open.org/docbook/xml/4.4/docbookx.dtd">
<book id="simple_book">
  <title>Very simple book</title>
  <chapter id="chapter_1">
    <title>Chapter 1</title>
      <para>Hello world!</para>
      <para>I hope that your day is proceeding <emphasis>splendidly</emphasis>!</para>
  </chapter>
  <chapter id="chapter_2">
    <title>Chapter 2</title>
    <para>Hello again, world!</para>
  </chapter>
</book>
```

Docbook versions and variants

Version 5.x or 4.y?

- Either, or… You won’t do a big mistake still using 4.y, since there is plethora of tools and docs.
- Conversion to DB 5 any time later

DocBook: layers and customization

- DocBook can be used as basic (*Full*)

- or simplified (*Simplified*) or to make a
- *customization*.

Which means: - modify *schema* - evt. modify (XSL) *styles* - XSL styles by importing the original style and *overriding* selected templates

Docbook Layers - Simplified

- derived languages/markups can be created by reduction or extension of allowed elements: *Simplified Docbook*
- from a family of elements just one is preserved/left, eg. `programlisting`, but not `screen`
- no "big things" like `book`, just `article`
- any document in Simplified Docbook is also a (full) Docbook doc, Docs for Simplified Docbook online

Docbook Slides

- Extension of Simplified Docbook
- For writing (PowerPoint-like) presentations — "foils".
- XSLT styles allow to make static- or JavaScript-enabled web/HTML pages.
- Modern browsers can even navigate through the structure (go to next slide, toc, etc.).

Docbook Processing Workflow

edit

write the source text with either a specialized (usually WYSIWYG) editor or just as plain-XML or even plain-text editors

validation

validate the source using Docbook schemata (DTD) and tools (validators)

further processing

such as filtering undesired parts, extracting of lists, ToC

visual production

styles for transformations into visual formats, eg. HTML(5), PDF usually includes creation of indexes, ToC

Docbook Tooling

- editors
- validation schemata and tools (usually generic validators ` Docbook Schema)

- styles for transformations to visual formats

Editors

- In the worst case, any plain-text editor can be used if supporting the required charset and encoding (eg. Unicode/UTF-8).
- Better to use any editor with auto-closing (or even auto-completion) of elements.
- If an on-the-fly validation is supported — the best!
- Ideally an WYSIWYG producing a valid Docbook text — eg. *XMLMind* (XXE) or *oXygen*.

Markdown-like Docbook creation

- Recently, there are "markup"/"markdown" tools for easy manual creation of (not only) Docbook sources.
- Syntax resembles wiki.

Markdown

widely used

AsciiDoc

in Python or Ruby implemented tool for conversion of AsciiDoc syntax to Docbook, see http://asciidoc.org/

pandoc

cross-conversion tool for many document formats

Example of Pandoc

Features of `pandoc` tool, see `man pandoc`:

- Pandoc is a Haskell library for converting from one markup format to another, and a command-line tool that uses this library.
- It can read *markdown* and (subsets of) Textile, reStructuredText, HTML, LaTeX, MediaWiki markup, Haddock markup, OPML, and DocBook;
- it can write *plain text, markdown, reStructuredText, XHTML, HTML 5, LaTeX (including beamer slide shows), ConTeXt, RTF, OPML, DocBook, OpenDocument, ODT, Word docx, GNU Texinfo, MediaWiki markup, EPUB (v2 or v3), Fiction- Book2, Textile, groff man pages, Emacs Org-Mode, AsciiDoc, and Slidy, Slideous, DZSlides, reveal.js or S5 HTML slide shows*.
- It can also produce *PDF* output on systems where LaTeX is installed.

Pandoc syntax

Pandoc’s enhanced version of *markdown* includes syntax for:

- footnotes, tables, flexible ordered lists, definition lists,
- fenced code blocks, superscript, subscript, strikeout, title blocks,
- automatic tables of contents, embedded LaTeX math, citations,
- and markdown inside HTML block elements.

Available editors

xmlmind

xmlmind.com of Pixware powerfull WYSIWYG editor for Docbook, DITA, XHTML and other formats including ebooks, can be further customized, suitable for enterprise environment and integration. Professional- and Evaluation- license.

oXygen

Synchro Soft SRL’s oXygen Editor/Developer/Author.

GNU Emacs

with nxml-mode

Validation Tools

- Docbook 4.x was DTD-constraint/defined
- Docbook 5.x uses namespaces and is RelaxNG/Schematron-constraint
- for transition, see http://docbook.org/docs/howto/
- and complete reference to use Docbook XSL

Transformation Tools

Mainly for:

- conversion into other document formats ("Office-like" as Office Open XML, Open Document Format, RTF, Wordprocessing XML) or
- visualization via PDF, PS, XSL:FO, or web formats (XHTML 1.x, XHTML 5)

Fundamental tools are Docbook XSL styles

- well parametrized, rich, modifiable
- a book on Docbook XSL by Sagehill publishers
- complete reference to use Docbook XSL