

# Docbook and ChunkCMS:

Creating and Maintaining Multilingual Documentation

by  
David C. Weichert

Workshop  
Skolelinux gathering in Erkelenz 2006  
at the Cornelius-Burgh Gymnasium  
2006-01-28

# What is DocBook?

- XML/ SGML Vocabulary
- Intended mainly for technical documentation (Hard-/Software)
- Also useful for other documents like scientific papers

# Docbook and Skolelinux

- Most of the documentation of individual components is written in Docbook already
  - KDE
  - GNOME
  - Linux Kernel Documentation
- The existing Skolelinux manuals are also written in Docbook

# Getting started (1)

- Open your preferred (XML-) editor
- Some editors like jEdit or (X)Emacs offer syntax checking against the Docbook Grammar (DTD) which helps writing well-formed and valid documents

# Getting started (2)

```
<?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>
  <bookinfo>
    <title>CAT - CipUX Administration Tool</title>
    <author>
      <firstname>Christian</firstname>
      <surname>Külker</surname>
    </author>
  </bookinfo>
  <chapter id="getting_started">
    <title>Getting started</title>
    <para>
      <userinput>aptitude install cipux</userinput>
    </para>
    <para>Finished!</para>
  </chapter>
</book>
```

# What is well-formed and valid?

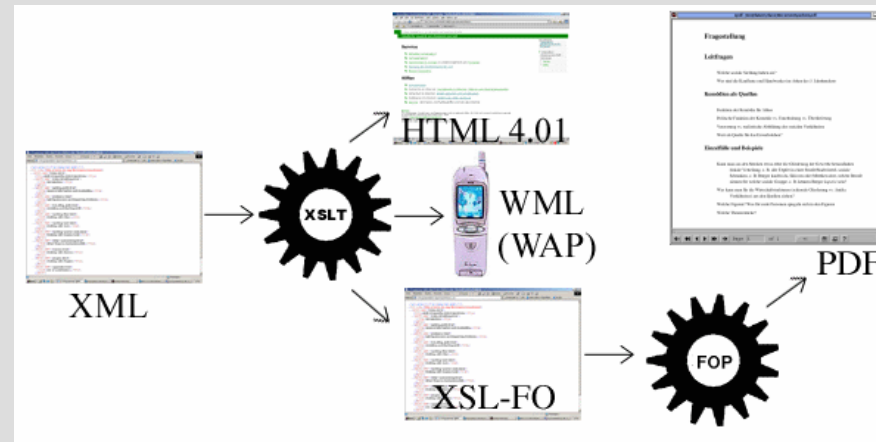
- In a well-formed XML document all elements are properly nested e.g.  
`<a><b><c></c></b></a>` and not  
\*`<a><b></a><c></b></c>`
- In a valid document all elements must conform to a grammar that specifies which element may, must or must not appear in which context. Like the data type of a variable can be enforced in a programming language a document grammar enforces conformance to the document (data) model.
- XML Grammars can be written as DTDs, Schemas or RelaxNG Schemas

# Parsing Docbook with Perl

- Fortunately checking well-formedness and validity are easy thanks to many parsers available. One easy way is using Perl:

```
#!/usr/bin/perl -w
use XML::LibXML;
# validate input file
my $file = $ARGV[0];
my $parser = new XML::LibXML;
my $document = $parser->parse_file($file);
my $validation_results = $document->validate;
if ($validation_results) {
    print "The document is valid XML! Processing...\n";
    print_document($file);
} else {
    print $validation_results."\nThe document is not valid XML! Aborting...\n";
    exit 0;
}
sub print_document {
    my $filei = shift;
    my @content = `cat $filei`;
    my $content = "";
    foreach (@content) {
        print $_;
        $content .= $_;
    }
}
```

# Processing Docbook (Single Source Publishing)



- DocBook can be processed using many existing general XML- / SGML-Tools like DSSSL, XSLT or XSL-FO or specialised tools/toolchains like db2latex, db2pdf, ...
- Most of them are available as Debian packages



# Processing our sample document

- Saxon (XSLT processor), FOP (XSL-FO processor) and DocBook XSL Stylesheets
  - `saxon sample.xml <path>/docbook_xslt/docbook-xsl-1.68.1/html/docbook.xsl > sample.html`
  - `saxon sample.xml <path>/docbook_xslt/docbook-xsl-1.68.1/html/chunk.xsl`
  - `saxon sample.xml <path>/docbook_xslt/docbook-xsl-1.68.1/fo/docbook.xsl > sample.fo;`  
`fop sample.fo sample.pdf`
- Jade(TeX) and DSSSL Stylesheets
  - `db2pdf -d <path>/docbook_dsssl/docbook-dsssl-1.79/print/docbook.dsl sample.xml`  
(Pitfall: wants iso-8859-encoding)
  - `db2html -d <path>/docbook_dsssl/docbook-dsssl-1.79/html/docbook.dsl sample.xml`

# Introducing ChunkCMS

- ChunkCMS is a content management system for writing and maintaining multilingual technical documentation
- It supports multiple authors and translators and includes features facilitating management of the document

# What's a chunk?

- The name Chunk is indicative of the way the document is treated by Chunk CMS. A chunk is a unit of text that can be as large as a complete document or a small part of a document like a paragraph. Any well-formed XML-Structure of any larger valid XML file can be a chunk.
- Chunks can be created manually (as long as you observe above rules) or automatically
- Thus your chunk can exactly match the amount of work a person can be expected to do

# Integration of the DocBook toolchain in ChunkCMS

- ChunkCMS supports cron scheduled (nightly) builds of the complete document in HTML and PDF (possibly more formats in the future)
- ChunkCMS supports builds on demand (triggered by a user)

# Workflow and Roles in ChunkCMS

- ChunkCMS has three predefined roles:
  - Language Maintainer
  - Chunk Maintainer
  - Translation Maintainer

# ChunkCMS: Language Maintainer

- Assigns chunks for authoring to chunk maintainers
- Assigns chunks for translation to translation maintainers
- Coordinates activity with other Language Maintainers
- ChunkCMS supports the Language Maintainer with
  - Up-to-date statistics of the project
  - “nightly” builds
  - Automatically generated emails to chunk maintainers, translation maintainers and himself if work is not completed as scheduled

# ChunkCMS: Chunk Maintainer

- 1) Writes chunks assigned by Language Maintainer
- 2) Several revision cycles (optional)
- 3) Marks chunk as completed
- 4) Language Maintainer approves chunk (or rejects for further revision)
- 5) Other Language Maintainers are notified by the system via email and can now assign the chunk to Translation Maintainers for translation

# ChunkCMS: Translation Maintainer

- 1) Translates chunks assigned by Language Maintainer
- 2) Several revision cycles (optional)
- 3) Marks chunk as completed
- 4) Language Maintainer approves chunk (or rejects for further revision)



# Literature and Links

- Norman Walsh: DocBook. The Definitive Guide. 1999-2005. <http://www.docbook.org/>
- DocBook Wiki.  
<http://wiki.docbook.org/topic/FrontPage>
- DocBook XSL Stylesheets.  
<http://docbook.sourceforge.net/projects/xsl/>
- DocBook DSSSL Stylesheets.  
<http://docbook.sourceforge.net/projects/dsssl/>