

DocBook: From Syntax to Publication

Norman Walsh

XML 2004
15 Nov - 19 Nov 2004
Washington, DC

<http://www.sun.com/>



Tuesday, 19 Nov 2004
Version 1.0

Table of Contents

A Little Background
DocBook Markup
Publishing
Stylesheet Customization
Schema Customization
Derived Schemas
Conclusion

A Little Background

A quick look at some features of structured documentation.

Structured Documentation

Benefits

Technical Challenges

Non-Technical Challenges

Storing Structured Documentation

XML or SGML or ...

OASIS

Evolution

Structured Documentation

- **Semantic rather than presentational**
- **Components have *identifiable* structure**
- **ASCII and Word (without templates) are not structured**
- **HTML and Word are somewhat structured**
- **DocBook is strictly structured**

Benefits

- **Multiple presentations from the same source (print, online, help, etc.)**
- **Documentation reuse**
- **Authors no longer have to worry about presentation**
- **Opportunities for improved authoring interfaces**

Technical Challenges

- **Relatively sophisticated processing required for presentation**
- **Document reuse requires careful management**
- **Users benefit from special authoring tools**

Non-Technical Challenges

- **Writing reusable documentation is different**
- **Authoring within a strict structure is different**
- **Authoring tools are (sometimes expensive and) different**

Storing Structured Documentation

- **XML is a natural system for storing structured documentation**
- **XML can be used to develop different vocabularies**
- **DocBook is an XML vocabulary designed for computer documentation**

XML or SGML or ...

- **DocBook was historically an SGML DTD.**
- **DocBook V4.x is supported in XML and SGML DTDs (with unofficial RELAX NG and W3C XML Schema implementations).**
- **DocBook V5.x is expected to be a RELAX NG grammar (with W3C XML Schema and DTD implementations).**

OASIS

- **OASIS: The Organization for the Advancement of Structured Information Standards**
A non-profit, international consortium that creates interoperable industry specifications based on public standards such as XML and SGML. OASIS members include organizations and individuals who provide, use and specialize in implementing the technologies that make these standards work in practice.
- **DocBook is a work product of the OASIS DocBook Technical Committee**

Evolution

- **DocBook is *stable*.**
- **Backwards incompatible changes can only occur at full version revisions (5.0, 6.0, etc.).**
- **Minor revisions (3.1, 4.1, 4.1.2) are always backwards compatible.**
- **Backwards incompatible changes have to be announced a full version before they are implemented.**
- **The Technical Committee may break these rules for the V5.0 release.**

DocBook Markup

The elements of DocBook...

Element Classes

"Hierarchy" Elements

Case Study: A Book

Case Study: An Article

Case Study: Reference Pages

"Information Pool" Elements

Case Study: FAQs

Case Study: MediaObjects

Case Study: Function Synopsis

Linking

Element Classes

- **There are two main classes of elements in DocBook**
- **“Hierarchy” elements provide gross structure**
- **“Information Pool” elements provide prose markup**
- **The information pool could be reused in a new hierarchy**
- **Conversely, the hierarchy could be preserved with a new technical vocabulary**

"Hierarchy" Elements

- **Set and Book**
- **Part and Reference**
- **Preface, Chapter, Appendix, Bibliography, Glossary, Index**
- **Article**
- **Section, Sect1...Sect5, SimpleSect**
- **RefEntry**
- **RefSect1...RefSect3**

Case Study: A Book

This is the DocBook XML source for a book.

```
<!DOCTYPE book
  PUBLIC "-//OASIS//DTD DocBook XML V4.3//EN"
  "http://www.oasis-open.org/docbook/xml/4.3/docb
<book>
<bookinfo>
  <title>An Example Book</title>
  <author>
    <firstname>Norman</firstname>
    <surname>Walsh</surname>
  </author>
  <copyright>
    <year>2004</year>
```



Case Study: A Book (Continued)

```
<holder>Sun Microsystems, Inc.</holder>
</copyright>
<contractnum>1234</contractnum>
<contractsponsor>Our Favorite Sponsor</contract
</bookinfo>
<preface><title>Introduction</title>
<para>...</para>
</preface>
<chapter><title>The First Chapter</title>
<para>...</para>
</chapter>
<!-- ... -->
<appendix><title>An Appendix</title>
<para>...</para>
```



Case Study: A Book (Continued)

`</appendix>`

`</book>`

Case Study: An Article

This is the DocBook XML source for an article.

```
<!DOCTYPE article
  PUBLIC "-//OASIS//DTD DocBook XML V4.3//EN"
  "http://www.oasis-open.org/docbook/xml/4.3/docb
<article>
<articleinfo>
  <title>An Example Article</title>
  <author>
    <firstname>Norman</firstname>
    <surname>Walsh</surname>
  </author>
  <copyright>
    <year>2004</year>
```

Case Study: An Article (Continued)

```
<holder>Sun Microsystems, Inc.</holder>
</copyright>
<confgroup>
  <confdates>15 Nov - 19 Nov 2004</confdates>
  <conftitle>XML 2004</conftitle>
</confgroup>
</articleinfo>
<section><title>A Section</title>
<para>...</para>
</section>
<appendix><title>An Appendix</title>
<para>...</para>
</appendix>
</article>
```

Case Study: Reference Pages

This is the DocBook XML source for a reference page.

```
<!DOCTYPE refentry
  PUBLIC "-//OASIS//DTD DocBook XML V4.3//EN"
  "http://www.oasis-open.org/docbook/xml/4.3/docb
<refentry>
<refmeta>
<refentrytitle>getpwnam</refentrytitle>
<manvolnum>3</manvolnum>
</refmeta>

<refnamediv>
<refname>getpwnam</refname>
<refname>getpwuid</refname>
```



Case Study: Reference Pages (Continued)

```
<refpurpose>get password file entry</refpurpose>  
</refnamediv>
```

```
<refsynopsisdiv><title>Synopsis</title>
```

```
<synopsis>
```

```
#include <pwd.h>
```

```
#include <sys/types.h>
```

```
struct passwd *getpwnam(const char * name);
```

```
struct passwd *getpwuid(uid_t uid);
```

```
</synopsis>
```

Case Study: Reference Pages (Continued)

```
</refsynopsisdiv>
```

```
<refsect1><title>Description</title>
```

```
<para>The <function>getpwnam</function> function  
returns a pointer to a structure containing the  
broken out fields of a line from  
<filename>/etc/passwd</filename> for  
the entry that matches the user name  
<parameter>name</parameter>.
```

```
</para>
```

```
<!--...-->
```

Case Study: Reference Pages (Continued)

```
</refsect1>
```

```
<!--...-->
```

```
</refentry>
```

"Information Pool" Elements

- **Lists (ordered, itemized, simple, ...)**
- **Admonitions (caution, warning, note, ...)**
- **“Verbatim” (program listings, screens, ...)**
- **Examples, figures, and tables**
- **Equations**
- **Graphics (Media Objects)**
- **Inlines (publishing, linking, markup, user interfaces, programming, operating systems, ...)**

Case Study: FAQs

```
<!DOCTYPE qandaset
  PUBLIC "-//OASIS//DTD DocBook XML V4.3//EN"
  "http://www.oasis-open.org/docbook/xml/4.3/docb
<qandaset>
<qandadiv><title>Sample Questions</title>
<qandaentry>
<question><label>Q1</label>
<para>Question para 1</para>
<para>Question para 2</para>
</question>
<answer><label>A1</label>
<para>Answer para 1</para>
<para>Answer para 2</para>
</answer>
```

Case Study: FAQs (Continued)

```
</qandaentry>
</qandadiv>
<qandadiv><title>Imponderables</title>
<qandaentry>
<question>
<para>Why?</para>
</question>
<!-- Some questions have no answers -->
</qandaentry>
<qandaentry>
<question>
<para>Why did the chicken cross the road?</para>
</question>
<answer>
```

Case Study: FAQs (Continued)

```
<para>To get to the other side</para>
```

```
</answer>
```

```
<answer>
```

```
<para>Some other silly reason I've forgotten.
```

```
</para>
```

```
</answer>
```

```
</qandaentry>
```

```
</qandadiv>
```

```
</qandaset>
```

Case Study: MediaObjects

```
<!DOCTYPE mediaobject
  PUBLIC "-//OASIS//DTD DocBook XML V4.3//EN"
  "http://www.oasis-open.org/docbook/xml/4.3/docb
<mediaobject>
<imageobject>
  <imagedata fileref="emc2.svg" />
</imageobject>
<imageobject>
  <imagedata fileref="emc2.eps" format="EPS" />
</imageobject>
<imageobject>
  <imagedata fileref="emc2.png" format="PNG" />
</imageobject>
<imageobject>
```

Case Study: MediaObjects (Continued)

```
<imagedata fileref="emc2.gif" format="GIF" />
</imageobject>
<textobject>
  <para>Energy is equal to mass times the speed of
light squared.</para>
</textobject>
<textobject>
  <phrase>E=mc2</phrase>
</textobject>
</mediaobject>
```

Case Study: Function Synopsis

```
<!DOCTYPE funcsynopsis
  PUBLIC "-//OASIS//DTD DocBook XML V4.3//EN"
  "http://www.oasis-open.org/docbook/xml/4.3/docb
<funcsynopsis>
<funcsynopsisinfo>
#include <pwd.h>
#include <sys/types.h>
</funcsynopsisinfo>

<funcprototype>
  <funcdef>struct passwd *<function>getpwnam</fun
  <paramdef>const char * <parameter>name</paramet
</funcprototype>
```



Case Study: Function Synopsis (Continued)

```
<funcprototype>
```

```
  <funcdef>struct passwd *<function>getpwuid</fun
```

```
  <paramdef>uid_t <parameter>uid</parameter></par
```

```
</funcprototype>
```

```
</funcsynopsis>
```

Linking

- **DocBook uses ID/IDREF linking**
- `<ulink url="someURI">hot text</ulink>`
- `<link linkend="someid">hot text</link>`
- `<xref linkend="someid" />`
- **XPointer/XLink harmonization is...possible**

Publishing

Getting from here to there. DocBook goes in, what comes out?

DocBook to ...

XSLT

DocBook XSL Stylesheets

DocBook to HTML

Transforming to HTML

DocBook to PDF

Transforming to PDF

DocBook to ...

- **HTML or XHTML**
- **XSL Formatting Objects (then to PDF)**
- **HTML Help**
- **Java Help**
- **Unix “man” pages**
- **TeX**
- **PDF and PostScript**
- **RTF**

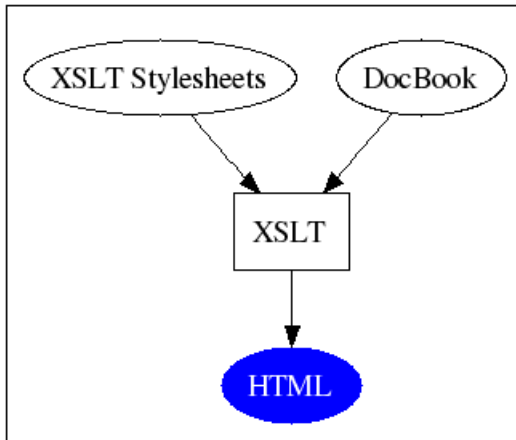
XSLT

- **XSL Transformations, part of the Extensible Style Language from the W3C**
- **Many processors available (Saxon, Xalan, ...)**
- **Only for XML; uses XML syntax and XPath as an expression language.**
- **Produces HTML, Formatting Objects, XML**
- **Formatting Objects can produce PDF (via Passive-TeX, FOP, RenderX, etc.)**

DocBook XSL Stylesheets

- **HTML or XHTML**
- **XSL Formatting Objects (then to PDF)**
- **HTML Help**
- **Java Help**
- **Unix “man” pages**

DocBook to HTML



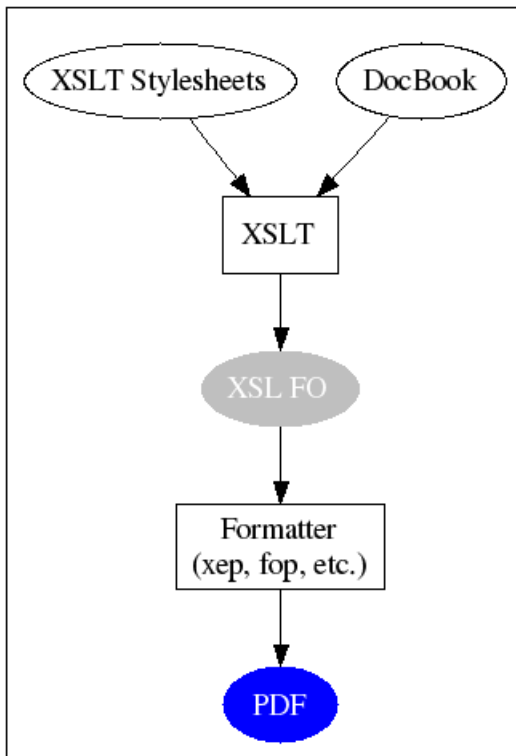
Publishing HTML

Transforming to HTML

```
java com.icl.saxon.StyleSheet -o example.html \  
    example.xml html/docbook.xsl
```

- **Processes** `example.xml`
- **With** `html/docbook.xsl`
- **Produces** `example.html`

DocBook to PDF



Publishing HTML

Transforming to PDF

```
java com.icl.saxon.StyleSheet -o example.fo \  
    example.xml fo/docbook.xsl  
java com.renderx.xep.XSLDriver example.fo
```

- **Saxon processes example.xml**
- **With fo/docbook.xsl**
- **And produces example.fo;**
- **XEP formats example.fo**
- **And produces example.pdf**

Stylesheet Customization

Changing the presentation of your documents.

XSL Customizations

Easy: Setting Parameters

Easy Parameters

Pretty Easy: A Customization Layer

Customization Skeleton

Customization Example 1

Customization Example 2

A Little Harder

Changing Title Pages

Updating the Title Page Template

Incorporate it into a customization layer

Write a template for contractnum

Stylesheet Customization (Continued)

...

XSL Customizations

- **Easy**
- **Pretty easy**
- **A little harder**
- **SMOP (Simple Matter of Programming)**

Easy: Setting Parameters

```
java com.icl.saxon.StyleSheet -o example2.html \  
    example.xml html/docbook.xsl admon.graphics=1
```

- **Processes** `example.xml`
- **With** `html/docbook.xsl` **with** `admon.graphics=1`
- **Produces** `example2.html`

With the FO stylesheet, `example2.pdf` can be produced in the analogous way.

Easy Parameters

- **Admonition graphics**
- **Callout style**
- **Line numbering and other verbatim styles**
- **Automatic labelling of chapters, sections, etc.**
- **CSS to use in the HTML result**
- **Page size (Letter/A4/etc.) and margin properties in FO result**
- **Number of columns on a page**
- **Page Header and footer styles**
- **Font families**
- **Profiling characteristics**

Pretty Easy: A Customization Layer

- **One XSLT stylesheet can “import” another.**
- **Useful if you want to set several parameters.**
- **Necessary if you want to set parameters that aren't simple string or numeric values.**
- **A step towards more aggressive customizations.**

Customization Skeleton

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999
                version="1.0">

    <xsl:import href="/path/to/docbook/xsl/html/doc

    <!-- your stuff goes here -->
    <xsl:param name="admon.graphics" select="1"/>

</xsl:stylesheet>
```

Customization Example 1

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999
                version="1.0">

  <xsl:import href="/path/to/docbook/xsl/html/doc

  <xsl:param name="admon.graphics" select="1"/>
  <xsl:param name="html.stylesheet">my.css</xsl:p

  <xsl:attribute-set name="shade.verbatim.style">
    <xsl:attribute name="border">0</xsl:attribute
    <xsl:attribute name="bgcolor">#E0E0E0</xsl:at
  </xsl:attribute-set>
</xsl:stylesheet>
```

Customization Example 2

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999
                version="1.0">

  <xsl:import href="/path/to/docbook/xsl/html/doc

  <xsl:param name="admon.graphics" select="1"/>
  <xsl:param name="html.stylesheet">my.css</xsl:p

  <xsl:attribute-set name="shade.verbatim.style">
    <xsl:attribute name="border">0</xsl:attribute
    <xsl:attribute name="bgcolor">#E0E0E0</xsl:at
  </xsl:attribute-set>
```

Customization Example 2 (Continued)

```
<xsl:template name="user.header.content">
  <div align="center">
    <h1>Your Corporate Header</h1>
  </div>
</xsl:template>

</xsl:stylesheet>
```

A Little Harder

- **Changing title pages**
- **Replacing individual templates**

Changing Title Pages

- **Copy the title page templates file**
- **Modify the template(s) you want to change**
- **Rebuild the templates stylesheet**
- **Incorporate it into a customization layer**
- **Style your document with the new stylesheet**

Updating the Title Page Template

- **Copy .../html/titlepage.templates.xml to tpage.xsl**
- **Edit the “book” template:**

```
<t:titlepage t:element="book" t:wrapper="div" cla
  <t:titlepage-content t:side="recto">
    <title/>
    <subtitle/>
    <corpauthor/>
    <authorgroup/>
    <author/>
    <othercredit/>
    <releaseinfo/>
```

Updating the Title Page Template (Continued)

```
<copyright />  
<legalnotice />  
<pubdate />  
<revision />  
<revhistory />  
<abstract />  
<contractnum />  
<contractsponsor />  
</t:titlepage-content>
```

- Transform `tpage.xml` to `tpage.xsl` (with `.../templates/titlepage.xsl`)

Incorporate it into a customization layer

- Use the import/include two-step:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999
                version="1.0">

    <xsl:import href="/sourceforge/docbook/xsl/html
    <xsl:include href="tpage.xsl"/>

</xsl:stylesheet>
```

- Transform with that stylesheet.

Write a template for `contractnum`

- Improve the presentation with generated text:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999
                version="1.0">

  <xsl:import href="/sourceforge/docbook/xsl/html
  <xsl:include href="tpage.xsl"/>

  <xsl:template match="contractnum" mode="titlepage"
    <div class="contract">
      <xsl:text>This work was funded by Contract
      <xsl:apply-templates mode="titlepage.mode"/
    </div>
```

Write a template for `contractnum` (Continued)

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

```
</xsl:stylesheet>
```

Uh, but what about internationalization?

- **Putting generated text directly in templates isn't very friendly to internationalization.**
- **The DocBook stylesheets support 45 (!) languages out of the box: Afrikaans, Arabic, Bangla, Basque, Bosnian, Bulgarian, Catalan, Chinese (Traditional), Chinese Simplified, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hebrew, Hungarian, Indonesian, Italian, Japanese, Korean, Latin, Lithuanian, Norwegian, Nynorsk, Polish, Portuguese, Portuguese (Brazil), Romanian, Russian, Serbian in Cyrillic script, Serbian in Latin script, Slovak, Slovenian, Spanish, Swedish, Thai, Turkish, Ukranian, Vietnamese, Xhosa.**

Uh, but what about internationalization?

(Continued)

- It's not difficult to leverage this flexibility in your stylesheets.

Write an I18N-friendly template for `contractnum`

- Internationalize the generated text:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999
                version="1.0">

  <xsl:import href="/sourceforge/docbook/xsl/html
  <xsl:include href="tpage.xsl"/>

  <xsl:template match="contractnum" mode="titlepage"
    <div class="contract">
      <xsl:apply-templates select="." mode="object"
    </div>
  </xsl:template>
```



Write an I18N-friendly template for `contractnum` (Continued)

```
</xsl:stylesheet>
```

- **Easy, right? But where does the text come from?**

Adding I18N Generated Text

Put I18N in the stylesheet:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999
                version="1.0">

    <xsl:import href="/sourceforge/docbook/xsl/html
<xsl:include href="tpage.xsl"/>

    <xsl:param name="local.l10n.xml" select="docume

<l:i18n>
    <l:l10n language="en">
        <l:context name="title">
```

Adding I18N Generated Text (Continued)

```
<l:template name="contractnum"
  text="This work was funded by Contract %n"/
</l:context>
</l:l10n>
<l:l10n language="de">
  <l:context name="title">
<l:template name="contractnum"
  text="Diese Arbeit wurde durch eine freundl
  </l:context>
</l:l10n>
</l:i18n>

<xsl:template match="contractnum" mode="titlepa
  <div class="contract">
```



Adding I18N Generated Text (Continued)

```
        <xsl:apply-templates select="." mode="object">
    </div>
</xsl:template>

</xsl:stylesheet>
```

Adding I18N Generated Text (Continued)

Handle the label:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999
                version="1.0">

    <xsl:import href="/sourceforge/docbook/xsl/html
<xsl:include href="tpage.xsl"/>

<xsl:param name="local.l10n.xml" select="docume

<l:i18n>
    <l:l10n language="en">
        <l:context name="title">
```

Adding I18N Generated Text (Continued) (Continued)

```
<l:template name="contractnum"
  text="This work was funded by Contract %n"/
</l:context>
</l:l10n>
<l:l10n language="de">
  <l:context name="title">
<l:template name="contractnum"
  text="Diese Arbeit wurde durch eine freundl
  </l:context>
</l:l10n>
</l:i18n>

<xsl:template match="contractnum" mode="titlepa
  <div class="contract">
```



Adding I18N Generated Text (Continued)

(Continued)

```
<xsl:apply-templates select="." mode="object">
  </div>
</xsl:template>

<xsl:template match="contractnum" mode="label.m">
  <xsl:apply-templates/>
</xsl:template>

</xsl:stylesheet>
```

S.M.O.P.

- **Rewrite entire sections of the stylesheets**
- **Support new markup constructs**

Schema Customization

Subsetting and extending DocBook.

The Role Attribute

Subsets

Extensions

General Form of a Customization Layer

Case Study: Restricting Role on Emphasis

Case Study: Removing Procedures from the DTD

Case Study: Removing Procedures from the RELAX

NG Schema

The Role Attribute

- All elements have a `role` attribute
- Stylesheets can key off of role values
- `<literal>` vs. `<literal role="widgetSpec">`
- DocBook *never* specifies role values

Subsets

- **Subsets constrain DocBook**
- **All documents that conform to the subset also conform to the full schema**
- **Enumeration of attribute values**
- **Removing elements**
- **Constraining content models**
- **Doesn't usually require stylesheet/tool customization**

Extensions

- **Extensions add new elements or attributes to DocBook**
- **Documents that conform to the extension may not conform to DocBook**
- **Adding new attributes or elements**
- **Extending content models**
- **Extensions can also remove elements**
- **Always requires stylesheet/tool customization**

General Form of a Customization Layer

In general, DTD customization layers look like this:

```
<!-- Turn off markup -->
```

```
<!-- Redefine parameter entities -->
```

```
<!-- include the base DTD -->
```

```
<!ENTITY % docbook PUBLIC "-//OASIS//DTD DocBook  
    "http://www.oasis-open.org/docbook/xml/  
>  
%docbook;
```

```
<!-- Redefine elements/attributes -->
```

Case Study: Restricting Role on Emphasis

```
<!ENTITY % emphasis.role.attrib
           role      (normal|emphasis)      "normal"
>

<!ENTITY % docbook PUBLIC "-//OASIS//DTD DocBook
           "http://www.oasis-open.org/docbook/xml/
>

%docbook;
```

Case Study: Removing Procedures from the DTD

```
<!DOCTYPE DocBook XML V4.3 No Procedures Subset -->
```

```
<!ENTITY % ebnf.block.hook "">
```

```
<!ENTITY % local.compound.class "">
```

```
<!ENTITY % compound.class
```

```
  "msgset | sidebar | qandaset
```

```
    %ebnf.block.hook;
```

```
    %local.compound.class;">
```

```
<!ENTITY % procedure.content.module "IGNORE">
```

```
<!ENTITY % task.content.module "IGNORE">
```

```
<!ENTITY % sidebar.element "IGNORE">
```

```
<!ENTITY % qandaset.element "IGNORE">
```

```
<!ENTITY % qandadiv.element "IGNORE">
```

Case Study: Removing Procedures from the DTD (Continued)

```
<!ENTITY % question.element "IGNORE">
<!ENTITY % answer.element "IGNORE">
<!ENTITY % revdescription.element "IGNORE">
<!ENTITY % caution.element "IGNORE">
<!ENTITY % important.element "IGNORE">
<!ENTITY % note.element "IGNORE">
<!ENTITY % tip.element "IGNORE">
<!ENTITY % warning.element "IGNORE">

<!ENTITY % docbook.dtd PUBLIC "-//OASIS//DTD DocB
                                "http://docbook.org/xml/4.
%docbook.dtd;

<!ENTITY % my.sidebar.mix
```

Case Study: Removing Procedures from the DTD (Continued)

```
"%list.class; |%admon.class;
|%linespecific.class; |%synop.class;
|%para.class; |%informal.class;
|%formal.class;
|%genobj.class;
|%ndxterm.class;          |beginpage
%local.sidebar.mix;">
```

```
<!ELEMENT sidebar (sidebarinfo?,
                   (%formalobject.title.content; )
                   (%my.sidebar.mix;)+)>
```

```
<!ENTITY % my.qandaset.mix
"%list.class;          |%admon.class;
```



Case Study: Removing Procedures from the DTD (Continued)

```
|%linespecific.class; |%synop.class;  
|%para.class; |%informal.class;  
|%formal.class;  
|%genobj.class;  
|%ndxterm.class;  
%local.qandaset.mix;">
```

```
<!ELEMENT qandaset (blockinfo?, (%formalobject.ti  
    (%my.qandaset.mix;)*,  
                                (qandadiv+|qandaentry+))>
```

```
<!ELEMENT qandadiv (blockinfo?, (%formalobject.ti  
    (%my.qandaset.mix;)*,  
    (qandadiv+|qandaentry+))>
```



Case Study: Removing Procedures from the DTD (Continued)

```
<!ELEMENT question (label?, (%my.qandaset.mix;)+)
```

```
<!ELEMENT answer (label?, (%my.qandaset.mix;)*, q
```

```
<!ENTITY % my.revdescription.mix  
  "%list.class; |%admon.class;  
  |%linespecific.class; |%synop.class;  
  |%para.class; |%informal.class;  
  |%formal.class;  
  |%genobj.class;  
  |%ndxterm.class;  
  %local.revdescription.mix;">
```



Case Study: Removing Procedures from the DTD (Continued)

```
<!ELEMENT revdescription ((%my.revdescription.mix
```

```
<!ENTITY % my.admon.mix
```

```
  "%list.class;
```

```
  |%linespecific.class; |%synop.class;
```

```
  |%para.class; |%informal.class;
```

```
  |%formal.class; |sidebar
```

```
  |anchor|bridgehead|remark
```

```
  |%ndxterm.class; |beginpage
```

```
  %local.admon.mix;">
```

```
<!ELEMENT caution (title?, (%my.admon.mix;)+)  
                  %admon.exclusion;>
```



Case Study: Removing Procedures from the DTD (Continued)

```
<!ELEMENT important (title?, (%my.admon.mix;)+)  
                    %admon.exclusion;>
```

```
<!ELEMENT note (title?, (%my.admon.mix;)+)  
               %admon.exclusion;>
```

```
<!ELEMENT tip (title?, (%my.admon.mix;)+)  
              %admon.exclusion;>
```

```
<!ELEMENT warning (title?, (%my.admon.mix;)+)  
                  %admon.exclusion;>
```

Case Study: Removing Procedures from the RELAX-NG Schema

```
# DocBook NG "Gin" No Procedures Subset
```

```
namespace db = "http://docbook.org/docbook-ng"
```

```
default namespace = "http://docbook.org/docbook-ng"
```

```
include "docbook.rnc" {  
    db.procedure = notAllowed  
}
```

Derived Schemas

A few examples of successful and interesting subsets and extensions.

SolBook: The Sun Documentation DTD

Simplified DocBook

Websites

Slides



SolBook: The Sun Documentation DTD

- **The source for docs.sun.com**
- **Restrictions to aid authoring and enforce style**

Simplified DocBook

- **Only supports articles**
- **Far fewer block elements**
- **Far fewer inlines**
- **About 100 tags vs about 400**

Websites

- **Uses DocBook information pool**
- **Replaces most of the hierarchy**
- **A website is a tree of nested web pages**
- **Stylesheets support both flat and tabular, two-column navigation**
- **See nwalsh.com for an example.**

Slides

- **Based on simplified DocBook**
- **Replaces article with a set of slides**
- **Slides can be divided into sections**
- **Stylesheets support HTML and PDF**
- **This presentation is generated from Slides source**

Conclusion

Q&A

Resources

Resources

- **DocBook Home Page (<http://www.oasis-open.org/docbook/>)**
- ***DocBook: The Definitive Guide* (<http://docbook.org/tdg/>)
Norman Walsh and Leonard Mueller
O'Reilly & Associates, Inc.
1st Edition October 1999
ISBN: 1-56592-580-7**
- ***DocBook XSL: The Complete Guide* (<http://www.sagehill.net>)
Bob Stayton
Sagehill Enterprises
2nd Edition September 2003**

Resources (Continued)

- **DocBook resources at [docbook.sf.net](http://docbook.sourceforge.net) (<http://docbook.sourceforge.net>). (XSL Stylesheets, Customization Layers, etc.)**