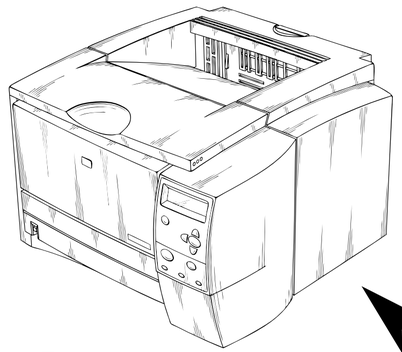
Overview

Software independent considerations

➟ General remarks

Why XML based publishing?

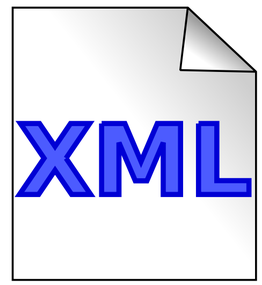


in-house printing

database

print

mobile



tablet

XML features

- Extensibility
  - Define your grammar
  - XML core extensions (linking,...)
- Interoperability
  - Cross-platform software support
- Open standard, no vendor lock-in
- Tons of (processing) frameworks / APIs

Editors, compositors, designers ...

Quote from How and Why Are Companies Using XML?.

It's Not about You! It is about publishers.

- they think it's “their” content
- they want
  - to use it, re-use it, slice it, and dice it
  - to own it and control it
  - to have access to it and be able to move it

Promises in publishing

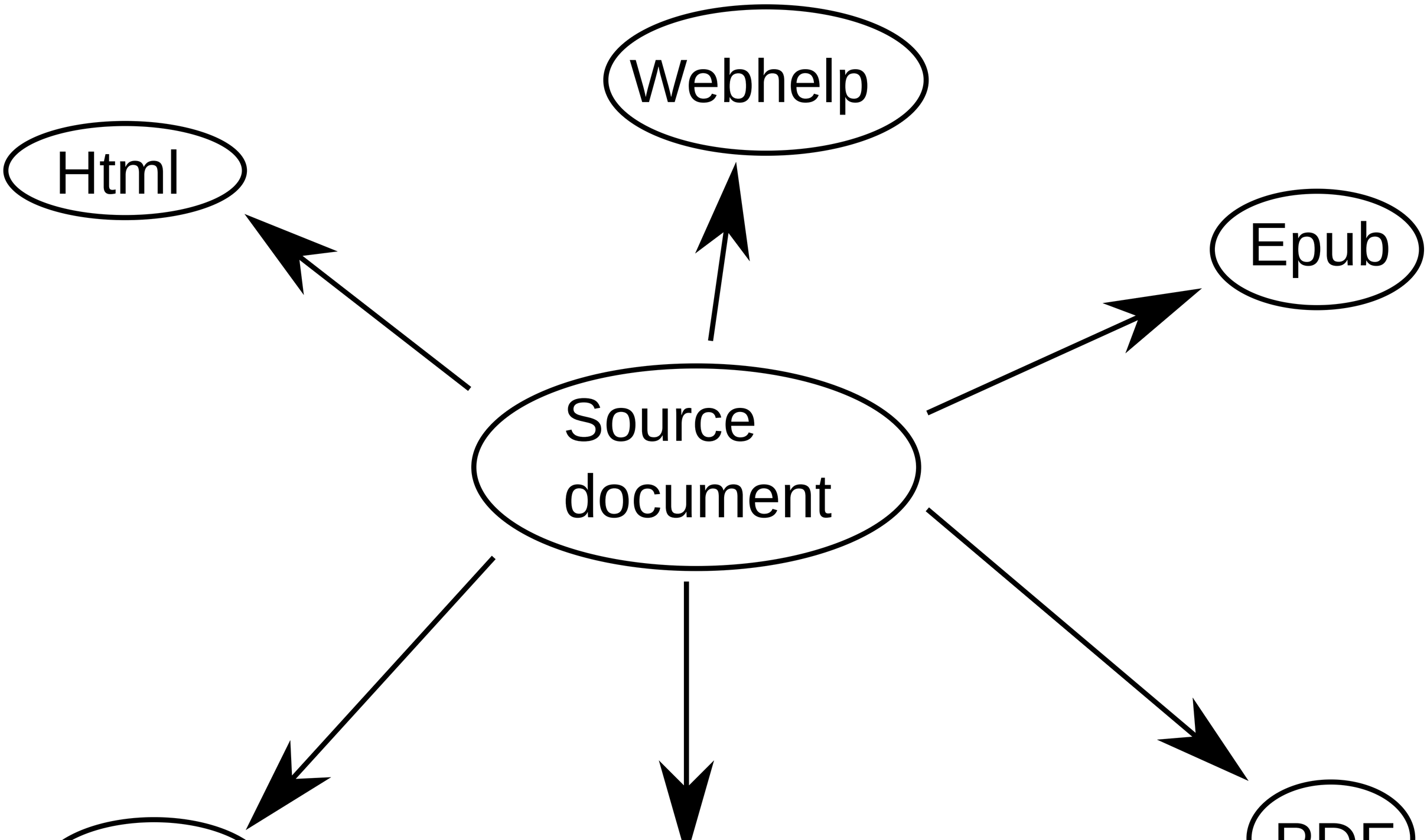
XML for publishing ...

- saves time and money
- is platform independent
- avoids vendor lock-in
- can be validated for QA
- allows for creating different target formats

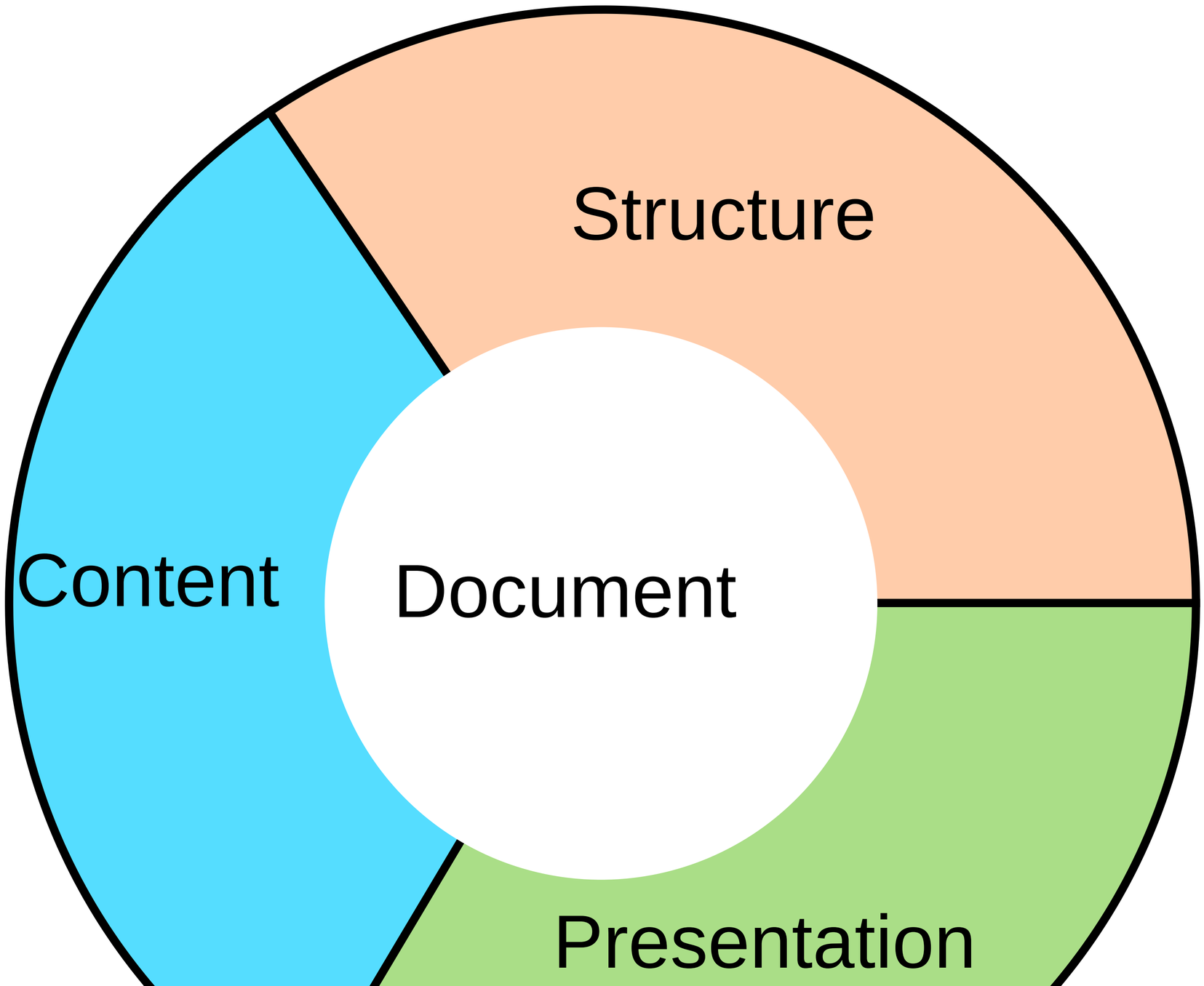
Publishing reality

- Refrain from fancy catalogs
- Stick to simple layouts
  - Technical documentation
  - Law publications

Single source publishing



Separating Structure, content and format



Separating concerns

Content

Words, images, audio / video

Structure

Chapters / sections, tables, lists

Presentation

Physical formatting (boldface, text size/color, ...)

Content

WHEN on board H.M.S. 'Beagle,' as naturalist, I was much struck with certain facts in the distribution of the inhabitants of South America, and in the geological relations of the present to the past inhabitants of that continent. These facts seemed to me to throw some light on the origin of species—that mystery of mysteries, as it has been called by one of our greatest philosophers. On my return home, it occurred to me, in 1837, that something might perhaps be made out on this question by patiently accumulating and reflecting on all sorts of facts which could possibly have any bearing on it. After five years' work I allowed myself to speculate on the

Hierarchical structure

<?xml	version="1.0" encoding="UTF-8"				
<?x...	href="http://docbook.org/xml/5.0/rng/docbook.rng" schematypens="http://relaxng.org/ns/structure/1.0"				
<?x...	href="http://docbook.org/xml/5.0/rng/docbook.rng" type="application/xml" schematypens="http://purl.oclc.org/dsdl/schematron"				
book	@xmlns	http://docbook.org/ns/docbook			
	@xmlns:xlink	http://www.w3.org/1999/xlink			
	@version	5.0			
	part	title			
		chapter	title	A Chapter	
			sect1	title	A section
				para	some content

Hierarchical structure, XML source

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="http://docbook.org/xml/5.0/rng/docbook.rng"
    schematypens="http://relaxng.org/ns/structure/1.0"?>
<?xml-model href="http://docbook.org/xml/5.0/rng/docbook.rng"
    type="application/xml" schematypens="http://purl.oclc.org/
<book xmlns="http://docbook.org/ns/docbook"
    xmlns:xlink="http://www.w3.org/1999/xlink" version="5.0">
    <part>
        <title/>
        <chapter>
            <title>A Chapter</title>
            <sect1>
                <title>A section</title>
                <para>some content</para>
            </sect1>
        </chapter>
    </part>
</book>
```

Presentation

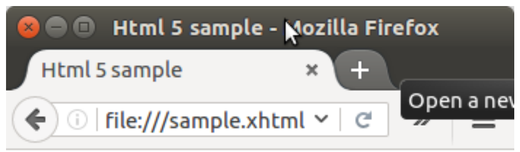
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>CSS sample</title>
    </head>
    <body>
        <p>Something <span
            style="color:red;font-weight:bold;"
            >big</span></p>
    </body>
</html>
```

Example 1: HTML 5, pure structure

Structure

```
<html xmlns="http://www.w3.org/1999/xhtml
    <head>
        <title>Test</title>
    </head>
    <body>
      <section>
        <h1>Intro</h1>
        <p>Some content</p>
      </section>
    </body>
</html>
```

Presentation



Introduction

Some content

Example 2: TeX / LaTeX

Structure / content	Presentation (PDF)

`\documentclass[12pt]{article}`
 	
`\begin{document}`
`  A nice LaTeX formula:`
 	
`  \begin{displaymath}`
`  e^x = \sum_{i=0}^{\infty}{x^i \over i`
`  \end{displaymath}`
 	
`\end{document}`
	A nice LaTeX formula:
	
	$$e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

Separating structure and presentation(s)

Pros	Cons
• Separation of editing / formatting concerns • Focus on content rather than formatting • Oblivious to format evolution (e.g. Epub) • Well suited for SCM, “diff-ing”	• No “true” WYSIWYG • Fixed formatting rules, no flexibility • Less layout control, especially in print

To set up your Raspberry Pi you will need:

	Item	Minimum recommended specification & notes
1	SD card	• Minimum size 4Gb; class 4 (the *class* indicates how fast the card is). • We recommend using branded SD cards as they are more reliable.
2a	HDMI to HDMI / DVI lead	• HDMI to HDMI lead (for HD TVs and monitors with HDMI input). **OR** HDMI to DVI lead (for monitors with DVI input). • Leads and adapters are available for few pounds -- there is no need to buy expensive ones!
2b	RCA video lead	• A standard RCA composite video lead to connect to your analogue display if you are not using the HDMI output.
3	Keyboard and mouse	• Any standard USB keyboard and mouse should work. • Keyboards or mice that take a lot of power from the USB ports, however, may need a powered USB hub. This may include some wireless devices.

Observations

- Well structured documents
- Focus on content rather than style
- Clearly defined semantics
- Automated generation supporting multiple output channels

Pros and cons of TeX / LaTeX

Pros	Cons

• Excellent typography
• Large community
• Mature engine
• Excellent platform support
• Multiple problem domain support
• Extensible vocabulary
	• Focus on print
	• Bad “office” authoring tool support
	  ◦ Steep learning curve
	  ◦ Inverse editing
	  ◦ Cryptic error messages
	• Bloated vocabulary

Tools of the trade

XMLMind Editor	**OxygenXML Editor**
• Strictly validating, near WYSIWYG, DocBook / DITA / MathML / XHTML editor. • Plugin architecture • Cross-platform Java™ based.	• Full-fledged XML IDE. • Strictly validating, near WYSIWYG, DocBook / DITA / MathML / XHTML ... editor. • Eclipse based

Overview

Software independent considerations

➟ Common building blocks

Inline formatting

HTML	`<p><b>Very</b> tiny</p>`
Docbook	`<para><emphasis>Very</emphasis> tiny.</para>`
LaTeX	`\textbf{Very} tiny.`
Rendering	**Very** tiny

Overview

- Software independent considerations
  - ➟ Common building blocks
    - ➟ Block level elements

Paragraphs

HTML	`<p>A paragraph</p>`	Docbook	`<para>A paragraph</para>`
LaTeX	`A paragraph\par`	**Rendering**	A paragraph

Lists

HTML		Docbook	
	    - One - Two 		<itemizedlist> <listitem> <para>One</para> </listitem> <listitem> <para>Two</para> </listitem> </itemizedlist>

LaTeX	\begin{itemize}
	  \item One
	  \item Two
	\end{itemize}
		Rendering	• One
			• Two

Tables

HTML

```
<table>
  <tr>
    <td>a1</td>
    <td>a2</td>
  </tr>
  <tr>
    <td>b1</td>
    <td>b2</td>
  </tr>
</table>
```

Docbook

```
<informaltable>
  <tr>
    <td>a1</td>
    <td>a2</td>
  </tr>
  <tr>
    <td>b1</td>
    <td>b2</td>
  </tr>
</informaltable>
```

LaTeX

```
\begin{tabular}{ll}
   a1 & a2 \\
   b1 & b2 \\
\end{tabular}
```

Rendering

a1	a1
b1	b2

Images

HTML	`<img src= 'smoke.png'/>`	Docbook	`<mediaobject> <imageobject> <imagedata fileref ="smoke.png"/> </imageobject> </mediaobject>`
LaTeX	`\includegraphics {smoke.png}`	Rendering	

Mathematical formulas

HTML / Docbook

```
<m:math>
  <m:mrow>
    <m:munderover>
      <m:mo>∫</m:mo>
            ...
    <m:msqrt>
      <m:mi>π</m:mi>
    </m:msqrt>
  </m:mrow>
</m:math>
```

LaTeX

```
\begin{equation}
  \int\limits_{-\infty}^{+\infty}
    e^{-x^2} dx = \sqrt{\pi}
\end{equation}
```

Rendering

$$\int\limits_{-\infty}^{+\infty} e^{-x^2}\ \mathrm{d}\,x = \sqrt{\pi}$$

Cross references

HTML

```
<h1 id="start"
>First section</h1>
<p>A remark.</p>

<h2>A subsection</h2>
<p>See <a href="#start"
>remark</a>.</p>
```

Docbook

```
<section xml:id="start">
 <title>First
   section</title>
 <para>A remark.</para>
 <section>
   <title>A subsection
     </title>
   <para>See
 <link linkend="start"
    >remark</link>.</para>
 </section>
</section>
```

LaTeX

```
\section{\label{start
}First section}
A remark.

\subsection{A subsection}
See remark at page
\pageref{start}.
```

Rendering

First section

A remark

See remark at page 1.

Document sectioning

<table>
<tr><th colspan="2">HTML</th><th>LaTeX</th><th colspan="2">Docbook</th></tr>
<tr><td><h1></td><td rowspan="6"><code><section></code> recursive</td><td>\chapter</td><td colspan="2"><part></td></tr>
<tr><td><h2></td><td>\section</td><td colspan="2"><book></td></tr>
<tr><td><h3></td><td>\subsection</td><td colspan="2"><chapter></td></tr>
<tr><td><h4></td><td>\subsubsection</td><td><sect1></td><td rowspan="3"><code><section></code> recursive</td></tr>
<tr><td><h5></td><td>\paragraph</td><td><sect2></td></tr>
<tr><td><h6></td><td>\subparagraph</td><td><sect3></td></tr>
</table>

Modular document components

HTML

```
<body>
 ...
  <object name="foo" type="text/html" data="table.html"/>
 ...
</body>
```

Docbook

```
<part xml:id="sd1">
  <title>Software development 1</title>
    <xi:include href="Sd1/gettingStarted.xml" xpointer="element(/1)"/>
    <xi:include href="Sd1/languageFundamentals.xml" xpointer="element(/1)"/>
...
```

LaTeX

```
\documentclass{article}
\input{mydefs.tex}
\begin{document}
...
\include{math.tex}
...
\end{document}
```

Overview

Software independent considerations

➠ Docbook

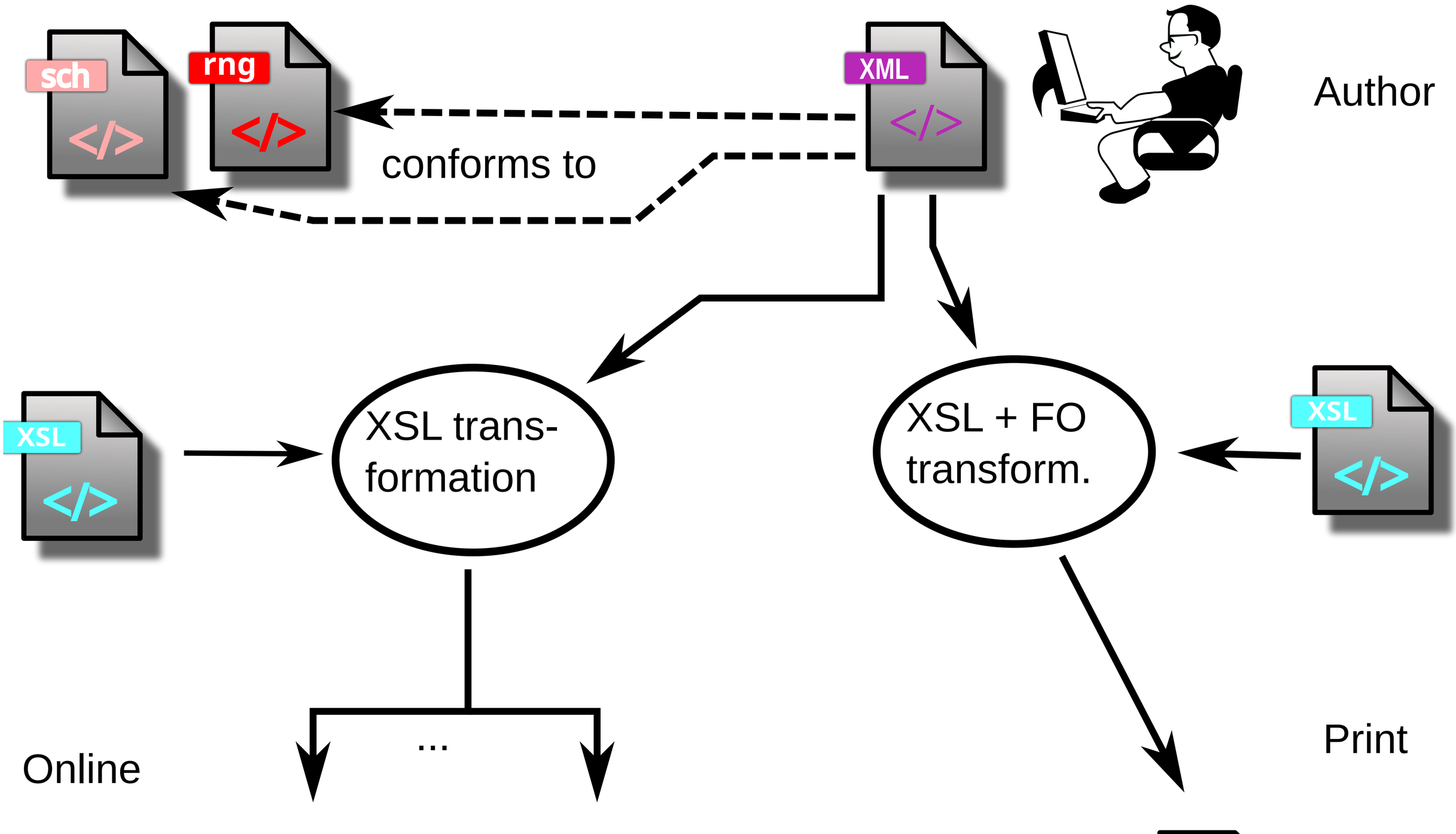
What is Docbook?

- Focus on technical documentation
- Excellent authoring user interface
- Semantic markup language
  - XML based

- Modular document xinclude support
- Topic support (Assemblies)
- MathML support:

$$\int_{-\infty}^{+\infty} e^{-x^2}\,\mathrm{dx} = \sqrt{\pi}$$

Authoring and publishing



Document representation

```
<section version="5.1"
  xmlns="http://docbook.org/ns/docbook"
  ...>

  <title>A Title</title>

  <para>A paragraph</para>
</section>
```

Software centric schema

Software specific support:

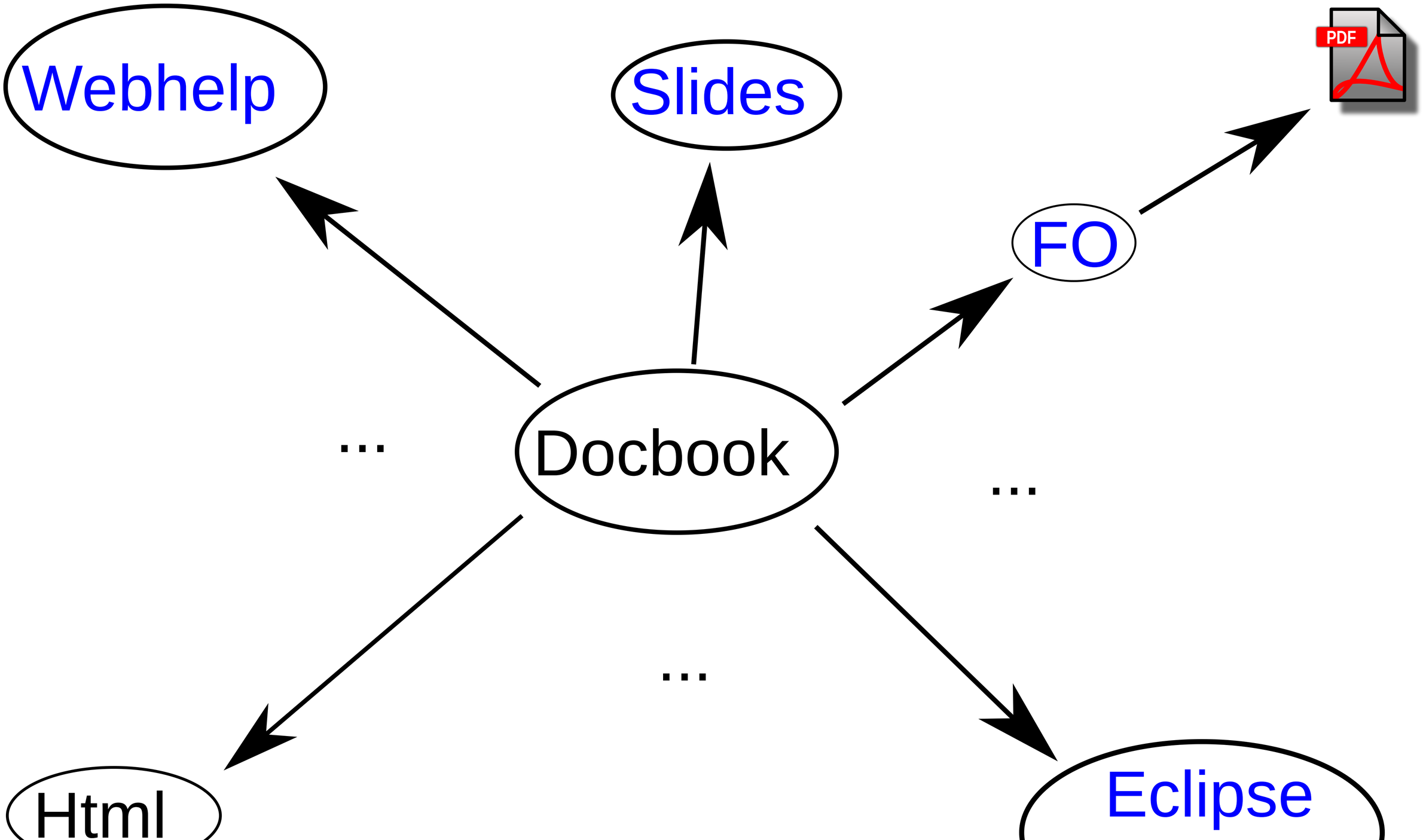
```
<xsl:stylesheet ❶ xmlns:xsl ❷ ="http://www.w
                version="2.0" ❸ >

  <xsl:output method="text" ❹/>

  <xsl:template ❺ match ❻ ="/memo">
    <xsl:value-of ❼ select ❽ ="from" />
  </xsl:template>

</xsl:stylesheet>
```

Document targets



Docbook components

• Document grammar
  ◦ RelaxNG based schema
  ◦ Schematron rules
	• Target format generators
	  ◦ XSL style sheets targeting HTML and FO
	  ◦ CSS and JavaScript for generated HTML

Target format overview

• HTML
  ◦ Standard
  ◦ Webhelp
  ◦ Mobile friendly
  ◦ ...
• Eclipse help, e.g. “Oxygen” documentation
	• PDF
	• Epub(3)
	• Slides
	• ...

Tooling / Software

Editing / office

- XMLmind XML Editor
- Oxygenxml XML Author

Editing / programming

emacs, vi, notepad, XML IDE, ...

XSLT processors

Saxon 6.5.5, Xalan, ...

FO (PDF) processors

- Apache FOP (Open Source)
- RenderX xep
- Antenna House formatter

Different schema languages

Docbook 5.x

Based on RelaxNG grammar

Docbook 4.x (old / outdated)

Based on DTD grammar

Overview

Software independent considerations

- Docbook
- Target formats
- HTML

Plain HTML

- Different HTML versions
- Static text
- Single or chunked output
- No full text search

Web help

- HTML 5 based
- Client side full text search index by virtue of JavaScript (Apache Lucene)
- JavaScript based navigation
- 3-rd party tool integration e.g. MathJax

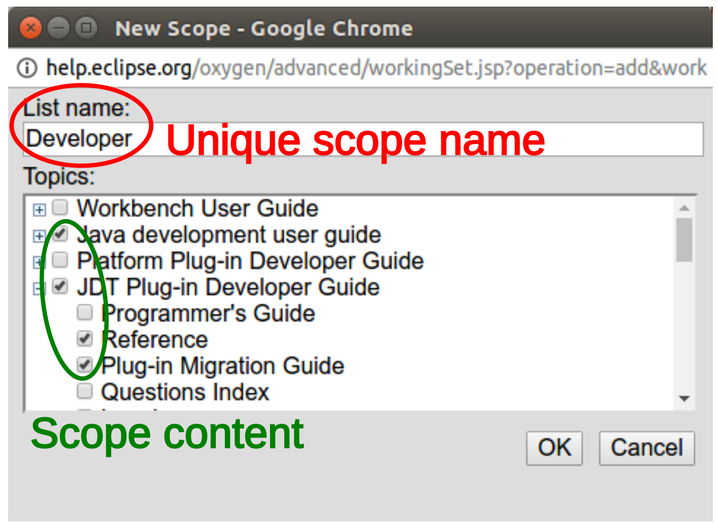
Overview

Software independent considerations

- Docbook
- Target formats
  - Eclipse Help

Eclipse help

- Application server based
- Server based full text search
  - Search scope definitions
- Standalone or centralized
- Plugin model, Web App deployable



Overview

Software independent considerations

- Docbook
- Target formats
  - Printed output

Printed output

- Focus on Formatting Objects
- Multiple formatting engines
- Multiple print formats

Overview

Software independent considerations

- Docbook
- Selected elements
  - Paragraph elements

Paragraph

View	Docbook	HTML
Some text.	`<para>Some text</para>`	`<p style='color:red' >Some text.</p>`

Caution: No style / formatting related parameters in Docbook.

This is by design and on purpose.

Reference: See Paragraph elements.

Overview

Software independent considerations

- Docbook
- Selected elements
- List elements

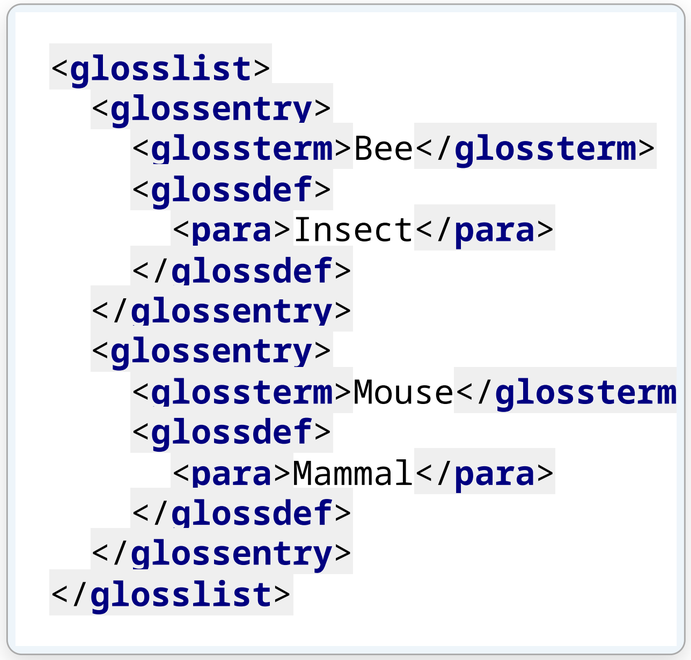
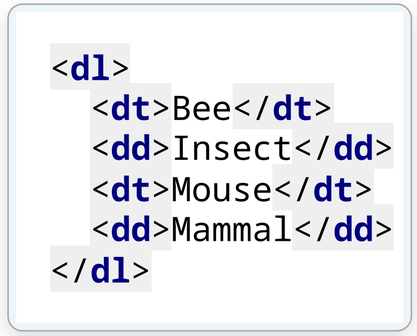
Itemized list

View	Docbook	HTML
• Bee • Ant	<itemizedlist> <listitem> <para>Bee</para> </listitem> <listitem> <para>Ant</para> </listitem> </itemizedlist>	 Bee Ant

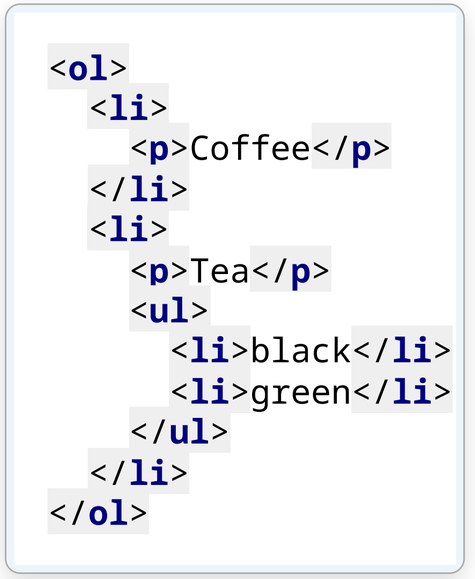
Ordered list

View	Docbook	HTML
”” • 1. Bee 2. Ant	<orderedlist> <listitem> <para>Bee</para> </listitem> <listitem> <para>Ant</para> </listitem> </orderedlist>	 Bee Ant

Glossary list

View	Docbook	HTML
Bee Insect **Mouse** Mammal	 <glosslist> <glossentry> <glossterm>Bee</glossterm> <glossdef> <para>Insect</para> </glossdef> </glossentry> <glossentry> <glossterm>Mouse</glossterm <glossdef> <para>Mammal</para> </glossdef> </glossentry> </glosslist>	 <dl> <dt>Bee</dt> <dd>Insect</dd> <dt>Mouse</dt> <dd>Mammal</dd> </dl>

Nested lists

View	Docbook	HTML
• 1. Coffee 2. Tea • black • green		

```
<orderedlist>
  <listitem>
    <para>Coffee</para>
  </listitem>
  <listitem>
    <para>Tea</para>
    <itemizedlist>
      <listitem>
        <para>black</para>
      </listitem>
      <listitem>
        <para>green</para>
      </listitem>
    </itemizedlist>
  </listitem>
</orderedlist>
```

```
<ol>
  <li>
    <p>Coffee</p>
  </li>
  <li>
    <p>Tea</p>
    <ul>
      <li>black</li>
      <li>green</li>
    </ul>
  </li>
</ol>
```

Reference

See List elements.

Overview

Software independent considerations

- Docbook
- Selected elements
  - Formal elements

A table

View

Col 1	Col 2
A1	A2
B	

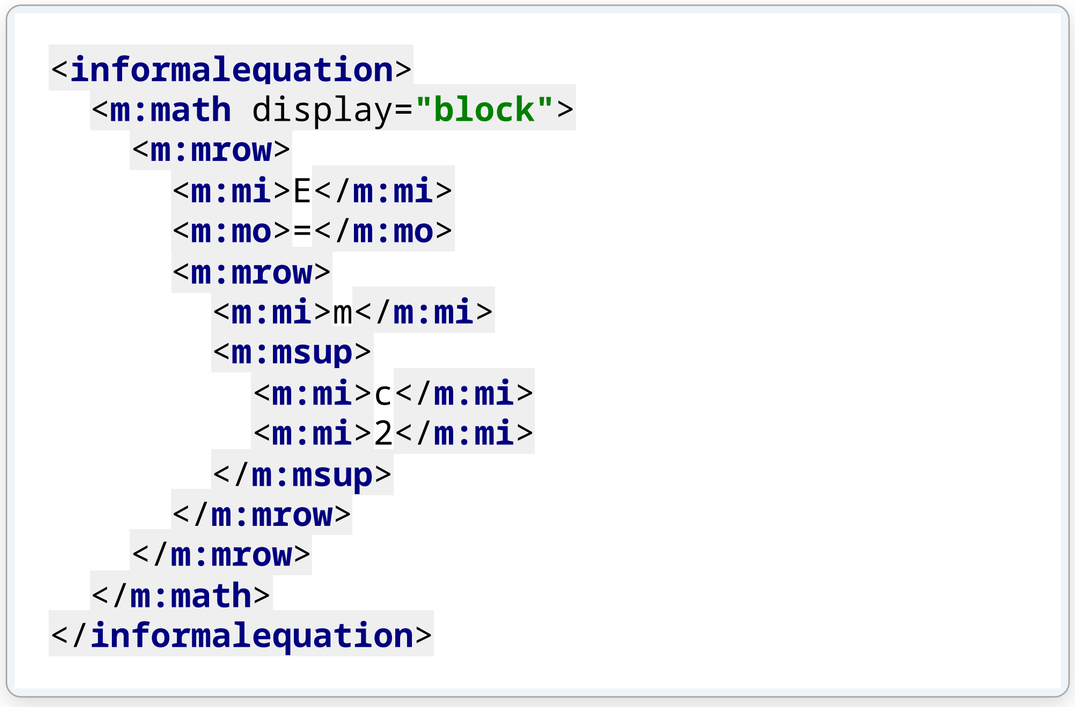
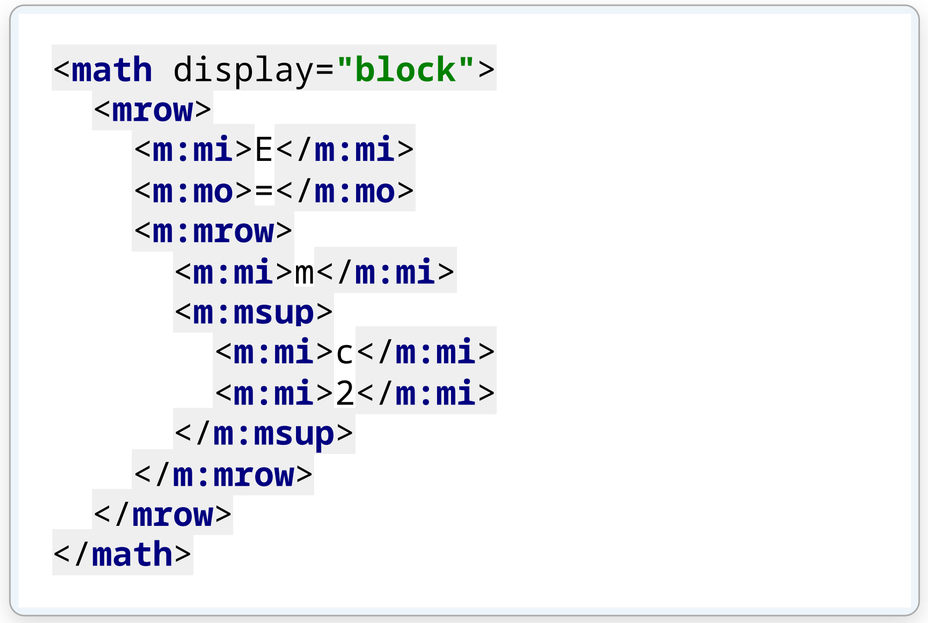
Docbook

```
<informaltable border="1">
  <tr>
    <th>Col 1</th>
    <th>Col 2</th>
  </tr>
  <tr>
    <td>A1</td>
    <td>A2</td>
  </tr>
  <tr>
    <td colspan="2">B</td>
  </tr>
</informaltable>
```

HTML

```
<table border="1">
  <tr>
    <th>Col 1</th>
    <th>Col 2</th>
  </tr>
  <tr>
    <td>A1</td>
    <td>A2</td>
  </tr>
  <tr>
    <td colspan="2">B</td>
  </tr>
</table>
```

A MathML equation

View	Docbook	HTML
$E = mc^2$		

```
<informalequation>
  <m:math display="block">
    <m:mrow>
      <m:mi>E</m:mi>
      <m:mo>=</m:mo>
      <m:mrow>
        <m:mi>m</m:mi>
        <m:msup>
          <m:mi>c</m:mi>
          <m:mi>2</m:mi>
        </m:msup>
      </m:mrow>
    </m:mrow>
  </m:math>
</informalequation>
```

```
<math display="block">
  <mrow>
    <m:mi>E</m:mi>
    <m:mo>=</m:mo>
    <m:mrow>
      <m:mi>m</m:mi>
      <m:msup>
        <m:mi>c</m:mi>
        <m:mi>2</m:mi>
      </m:msup>
    </m:mrow>
  </mrow>
</math>
```

A TeX equation

Docbook

```
<informalequation>
  <mathphrase>
  $ |x| = \left\{
   \begin{array}{rl}
    -x & \mbox{if $x&lt;0$} \
     x & \mbox{otherwise}
   \end{array}\right.$
  </mathphrase>
</informalequation>
```

HTML

```
<span class="mathphrase">
  $ |x| = \left\{
   \begin{array}{rl}
    -x & \mbox{if $x&lt;0$} \
     x & \mbox{otherwise}
   \end{array}\right.$
</span>
```

$$|x| = \begin{cases} -x & \text{if } x < 0 \\ x & \text{otherwise} \end{cases}$$

Reference

See Formal elements.

Overview

Software independent considerations

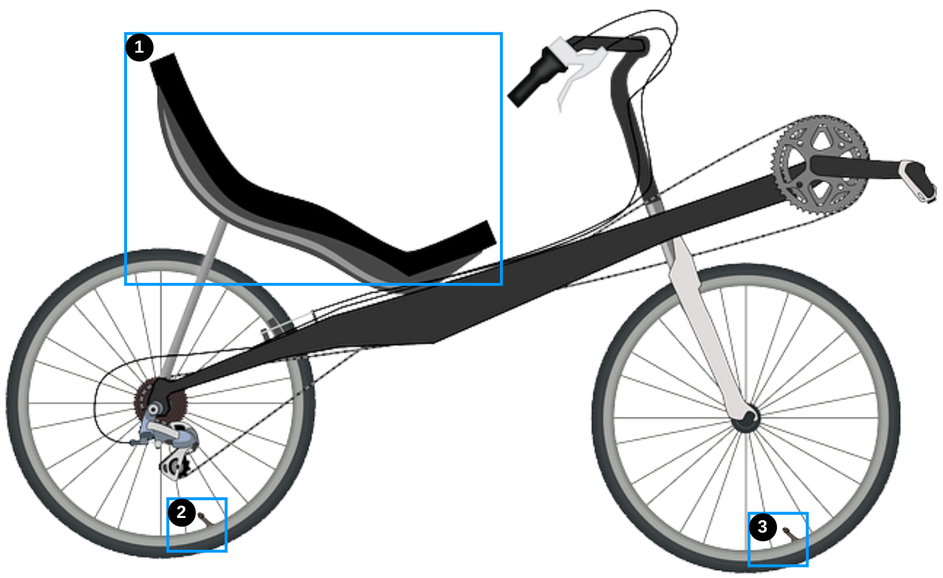
- Docbook
- Selected elements
  - Graphic elements

Figure

Mountain spring

```
<figure >
  <title>Mountain spring</title>
  <mediaobject>
    <imageobject>
      <imagedata fileref=
        "Ref/DbookIntro/mountain.jpg"/>
    </imageobject>
  </mediaobject>
</figure>
```

Image map + calloutlist



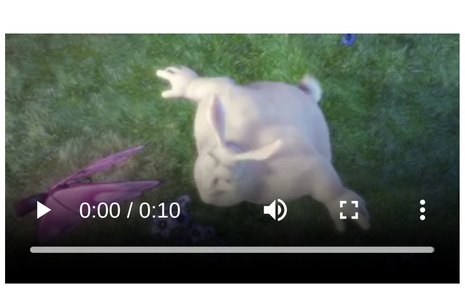
1 Seat

3 2 Valves

```
<mediaobject>
  <imageobjectco>
    <areaspec ...>
      <area coords="83,16,340,187"
          xml:id="a1" linkends="c1"/>
      ...
    </areaspec>
    <imageobject>
      <imagedata fileref="recumbent.png.svg"/>
    </imageobject>
    <calloutlist>
      <callout arearefs="a1" xml:id="c1">
        <para>Seat</para>
      </callout>
      <callout arearefs="a1 a2" xml:id="c1">
        <para>Valves</para>
```

Video

Video courtesy of Big Buck Bunny.




```
<videoobject>
  <videodata
      fileref="buckBunny.mp4"
       format="video/mp4">
    <multimediaparam
          name="controls"
         value="controls"/>
  </videodata>
</videoobject>
```

Overview

Software independent considerations

- Docbook
- Selected elements
  - Admonition elements

A warning

View	Docbook

Caution
Beware of overheating!
	`<caution>`
	`  <para>Beware of overheating!</para>`
	`</caution>`

Reference

See Admonition elements: important, note, tip, warning.

Overview

Software independent considerations

- Docbook
  - Selected elements
    - Sectioning elements

Recursive sections

```
<chapter version="5.1"
  xmlns="http://docbook.org/ns/docbook
  <title>Top</title>
  <section>
    <title>Level 1</title>
    <section>
      <title>Level 2</title>
      <section>
        <title>Level 3</title>
        <para>Hello!</para>
      </section>
    </section>
  </section>
</chapter>
```

```
<html>
  ...
  <body>
    <h1>Top</h1>
    <h2>Level 1</h2>
    <h3>Level 2</h3>
    <h4>Level 3</h4>
    <p>Hello!</p></body>
</html>
```

Non-recursive sections

```
<chapter version="5.1"
  xmlns="http://docbook.org/ns/docbook
  <title>Top</title>
  <sect1>
    <title>Level 1</title>
    <sect2>
      <title>Level 2</title>
      <sect3>
        <title>Level 3</title>
        <para>Hello!</para>
      </sect3>
    </sect2>
  </sect1>
</chapter>
```

```
<html>
  ...
  <body>
    <h1>Top</h1>
    <h2>Level 1</h2>
    <h3>Level 2</h3>
    <h4>Level 3</h4>
    <p>Hello!</p></body>
</html>
```

See <chapter>, <section>, <sect1>, <sect2>, <sect3>, <sect4>, <5>, <sect5>, <sect6>, <simplesect>, <refentry>.

Overview

Software independent considerations

- Docbook
- Selected elements
- Links

Two different link flavours

Internal document links

Referential integrity by ID / IDREF constraints:

```
<chapter id="intro">
...
<chapter> ...
See <xref linkend="intro"/> ...
```

External links

These are “usual” hypertext links:

```
<para>See
<link href="http://tdg.docbook.org">Docbook</link>
.</para>
```

Followup exercise

243. Internal document links

Overview

Software independent considerations

- Docbook
  - Selected elements
    - Top level elements

Choosing a top level element

- Root element is purpose dependent
- Schema based options in Docbook 5.x (RelaxNG) requiring an `<info>` child in 5.1.
- No limitation in Docbook 4.x (DTD).

Allowed 5.1 top level elements

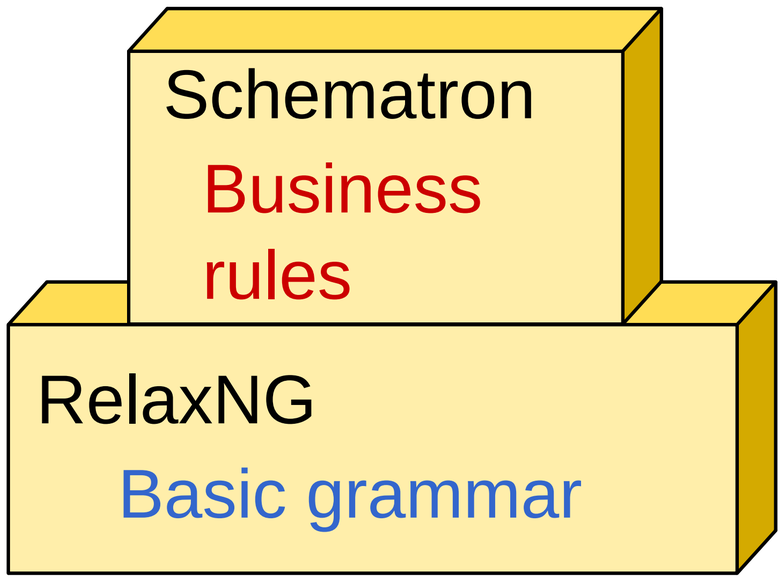
Structure	Big
chapter section (recursive), sect1, sect2, sect3, sect4, sect5 refsection (recursive), refsect1, refsect2, refsect3	set, book, part

Component	Medium
acknowledgements, appendix, bibliography, colophon, dedication, glossary, index, para, preface, refentry, reference, setindex, toc	article

Overview

- Software independent considerations
  - Docbook
    - Schematron

Schematron on top of RelaxNG



Each <title> must contain at least one word

Each <chapter> starts with a <title>

Example: `xml:id` and permalink

```
<chapter id="introduction" (1)> ...
  <section xml:id="firstSection" (1)>
    <title>First section</title>
    <para>A stable link</para>
  </section>
  <section> <!-- no xml:id attribute-->
    <title>Second section</title>
    <para>No stable link</para> ...
```

```
<!-- file introduction.html -->
<html>
  ...
<h2 id="firstSection" (1)>First section</h2>
<p>A stable link</p>

<h2 id="d03213" (2)>Second section</h2>
<p>No stable link</p>
```

(1) Defining chunk's base name `introduction.html`.

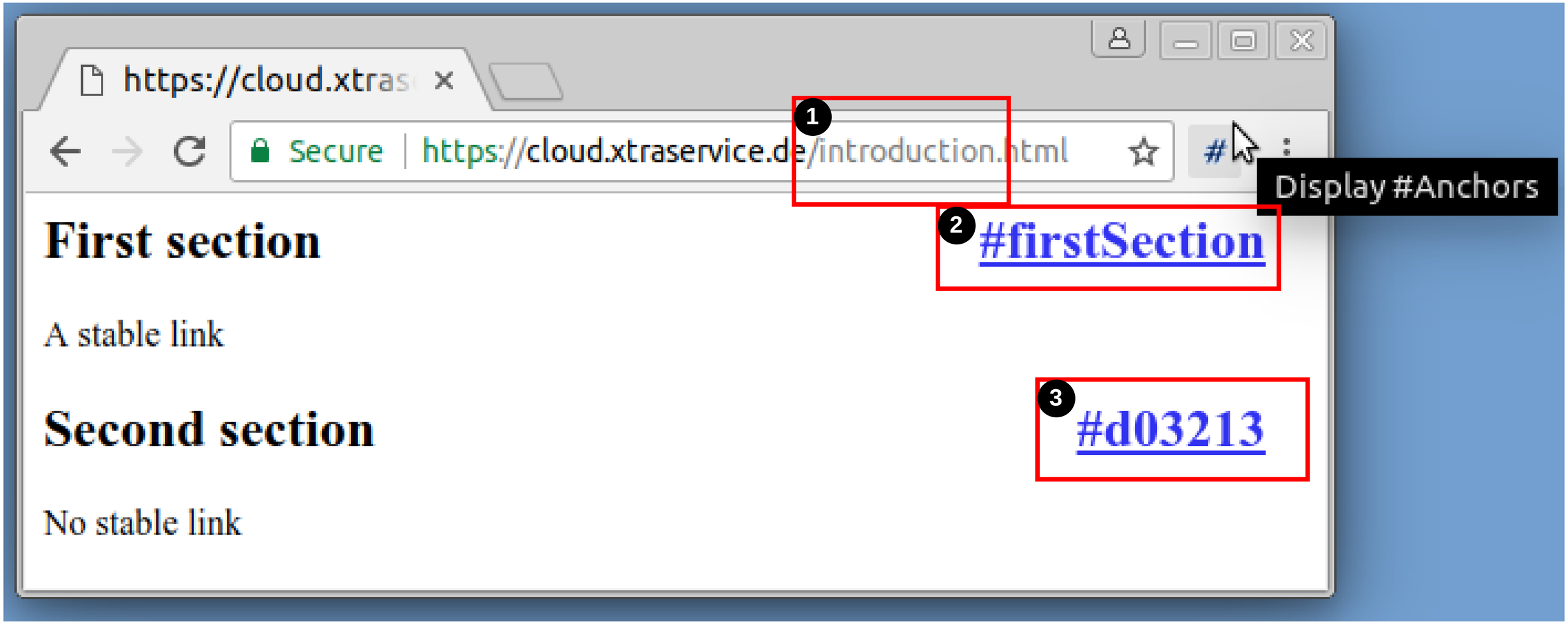
---

(1) Stable target `http://...introduction.html#firstSection`.

---

(2) Instable target `http://...introduction.html#d03213`.

Using Display #Anchors



1. The page's URI based on `xml:id` value `introduction`.

2. Stable `https://.../introduction.html#firstSection`.

Considerations author based permalink

Requirement

Important elements (`<chapter>`, `<section>`, `<table>`...) must provide an xml:id value.

Implementation choices

- Modify underlying RelaxNG schema.

  Result: Restricted schema (Inheritance relationship)
- Add Schematron integrity rule on top of schema.

Schematron permalink rule

```
<s:pattern>
  <s:title>Mandatory Id definition constraint</s:title>
  <s:rule context="db:chapter|db:section|db:table|db:qandaset">
    <s:assert test="@xml:id"
       >Each chapter,section, table ... must have a unique id.</s:assert>
  </s:rule>
</s:pattern>
```

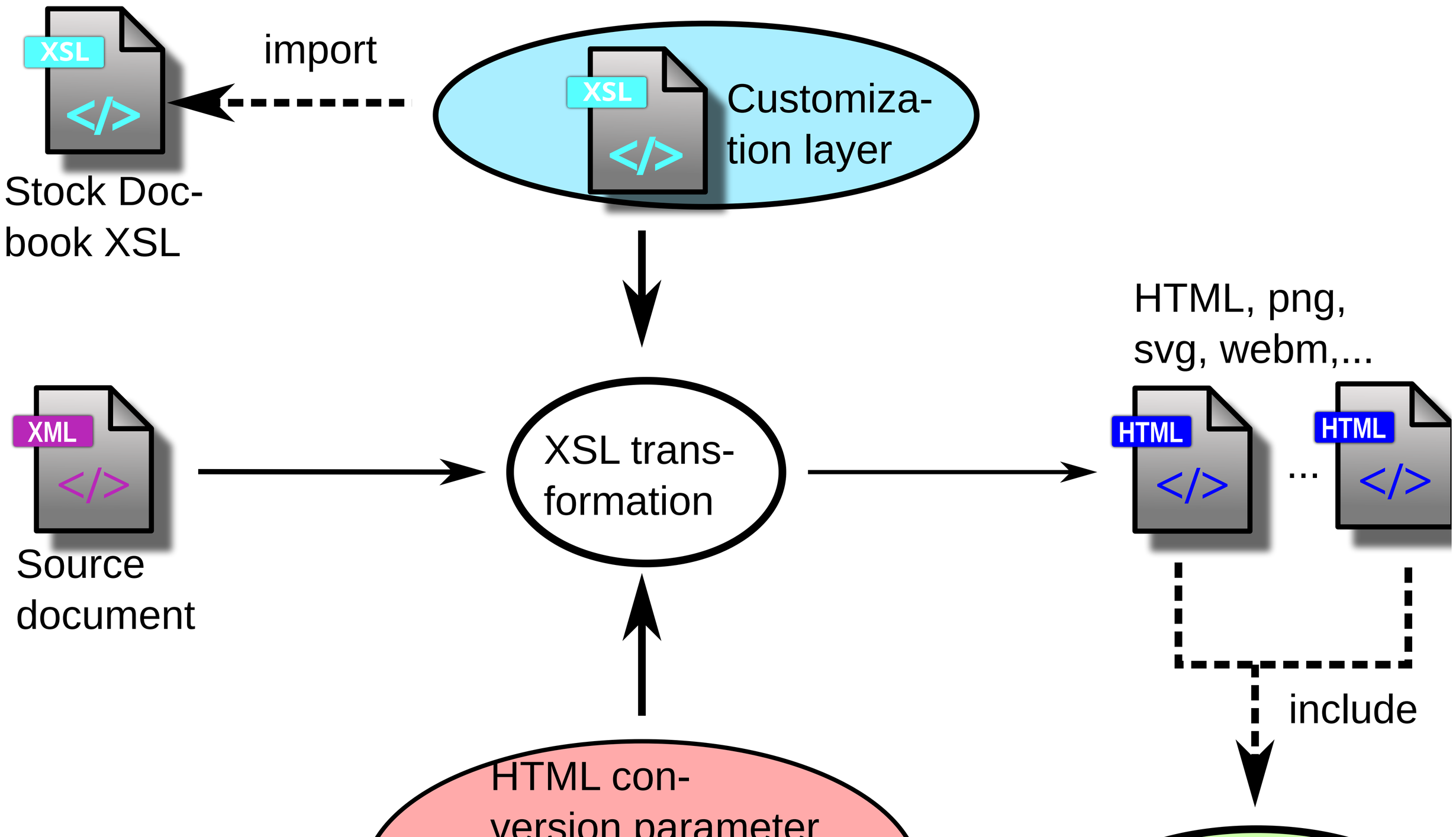
Overview

Software independent considerations

➟ Docbook

➟ Customizing

HTML customization overview



Overview

Software independent considerations

- Docbook
- Customizing
  - Docbook XSL built in parameters

Target specific configuration

- XSL transformation configuration parameters.
- Separate categories:
  - HTML
  - FO
  - Slides
  - Website
- Tool support (XMLMind, OxygenXml, ...)

Link stability

```
<book ...>
  <title>XML for Newbies</title>
  <chapter xml:id="intro">
    <title>Introduction</title>
    <para>...</para>
  </chapter>
  <chapter xml:id="work">
    <title>Working with objects</title>
    <para>...</para>
  </chapter>
</book>
```

Navigation structure.

- Index.html
- Per chapter:
  - **ch01**.html
  - **ch02**.html

Synthetically generated filenames.

use.id.as.filename = 1

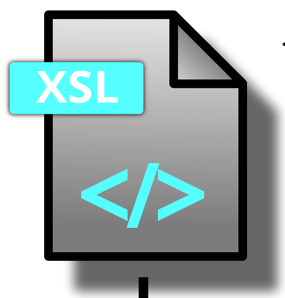
```
<book ...>
  <title>XML for Newbies</title>
  <chapter xml:id="intro">
    <title>Introduction</title>
    <para>...</para>
  </chapter>
  <chapter xml:id="work">
    <title>Working with objects</title>
    <para>...</para>
  </chapter>
</book>
```

Navigation structure.

- Index.html
- Per chapter:
  - **intro**.html
  - **work**.html

Providing link stability:

Parameter: use.id.as.filename




```
<xsl:when test =
  "$use.id.as.filename != 0>
  ...
</xsl:when>
```

XSL- trans-formation

...

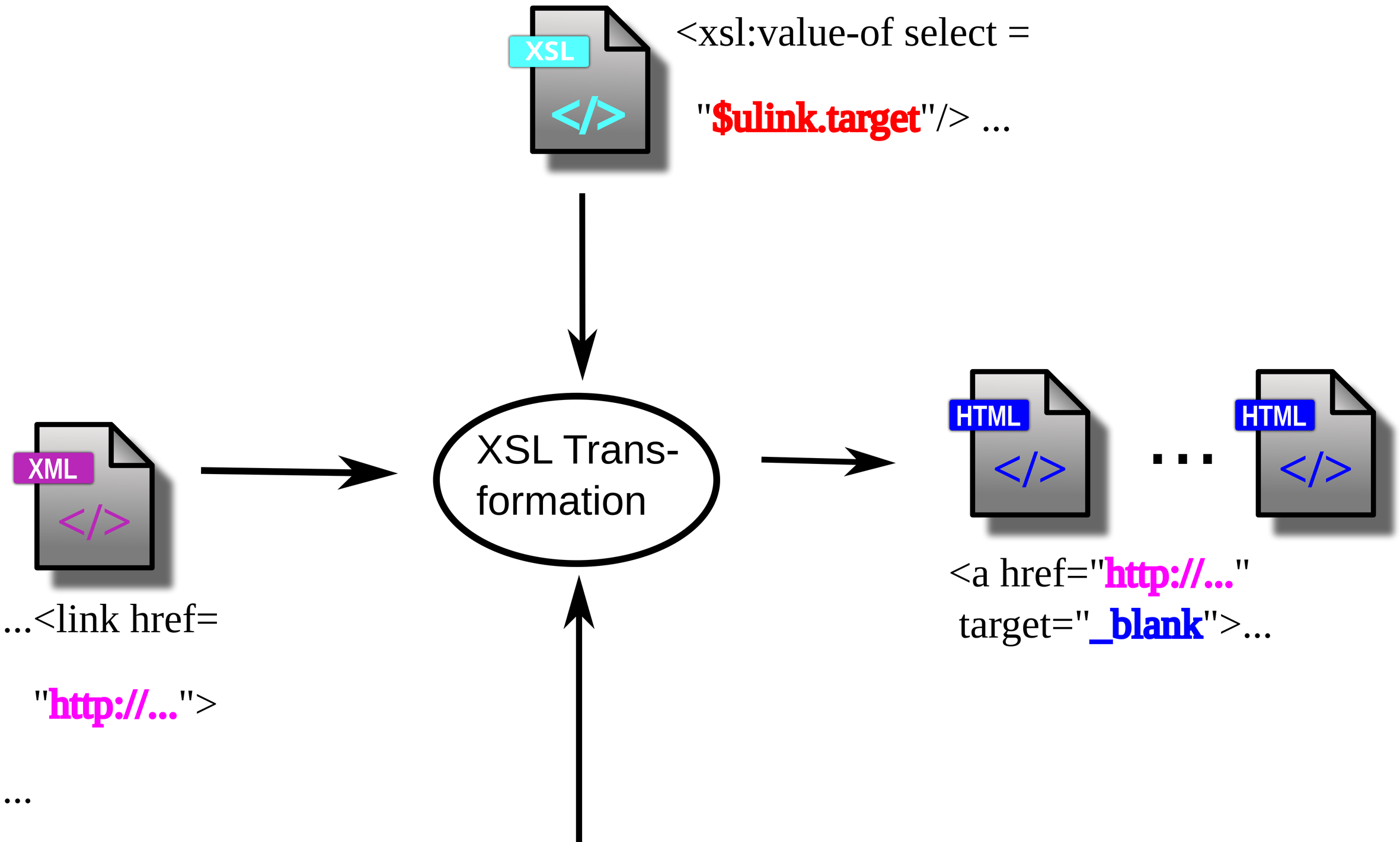

```
<chapter
  id="intro">...
<chapter
  id="work">...
```

intro.html

work.html

...

Customization parameter ulink.target



callout.unicode / *callout.graphics*

```
public class X { ❶
    void y (void) {...} ❷
}
```

❶ Class declaration

Followup exercise

Links

- DocBook XSL Stylesheets User Reference: Parameters

Overview

Software independent considerations

- Docbook
- Customizing
  - Docbook XSL style sheets

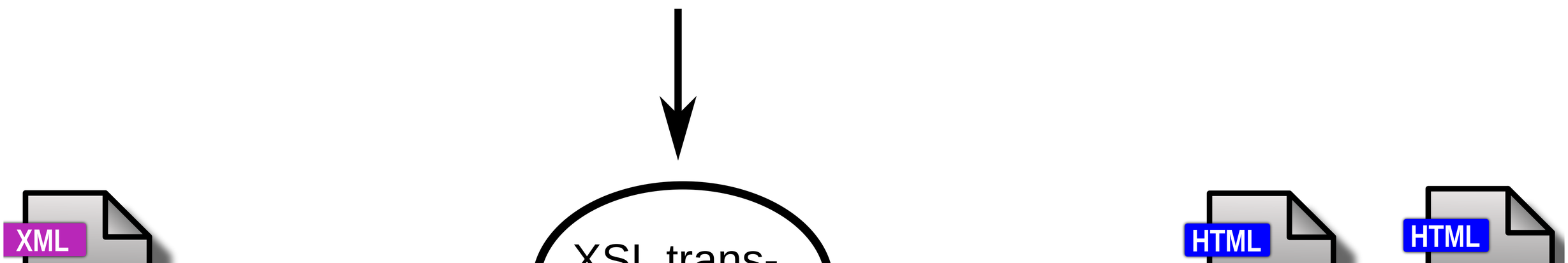
A sample customize.xsl

Stock Docbook webhelp.xsl

```
<xsl:import href="../../xhtml/chunk.xsl"/>
<xsl:include href="webhelp-common.xsl"/>
<xsl:include href="titlepage.templates.xsl"/>
```

Local customi-zation

```
<xsl:template
    name="webhelpheader.logo">
  <img src='mylogo.svg' alt="My site" />
</xsl:template>
```



Categories

• Adding Javascript
  ◦ Touch gestures
  ◦ Dynamic elements
• Embedded objects
  ◦ Videos
  ◦ MathML / LaTeX
	• Headers and footers
	  ◦ Company logo
	  ◦ Navigation icons
	• Front page

Example: videos

```
<xsl:template match="d:videodata">
  <video controls="controls" preload="auto">
    <xsl:attribute name="title">
      <xsl:value-of select="normalize-space(../../../d:title)"/>
    </xsl:attribute>

    <xsl:variable name="imageFilename">
      <xsl:call-template name="mediaobject.filename">
        <xsl:with-param name="object" select=".."/>
      </xsl:call-template>
    </xsl:variable>

    <source src="{$imageFilename}" type='video/mp4' />
    <source src="{$imageFilename}.ogv"/>
  </video>
</xsl:template>
```

Links

- Customizing DocBook XSL

Overview

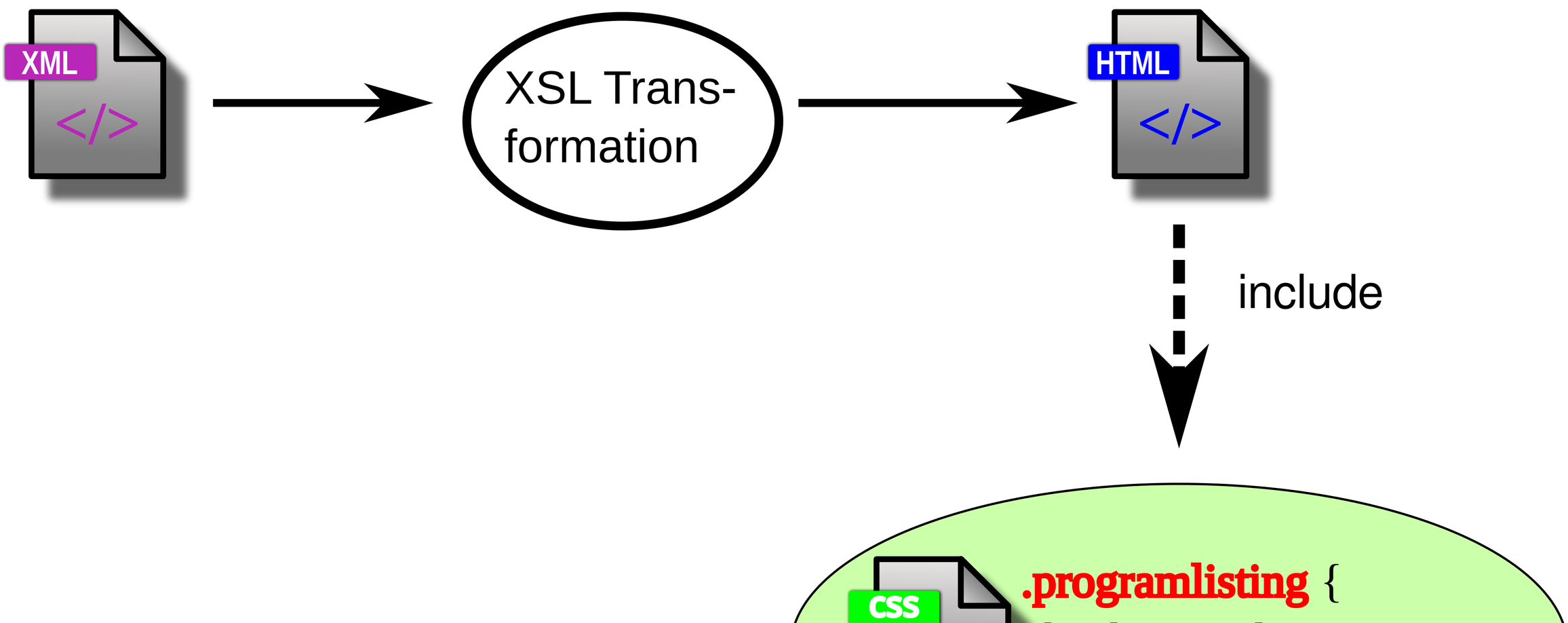
Software independent considerations

- Docbook
- Customizing
- CSS

Customize by CSS

<programlisting>
public class
Start {...}
</programlisting>

<pre class=
'**programlisting**'>
public class
Start {...}
</pre>



Example CSS modifications

```
div.example > p.title,
div.figure > p.title,fig
div.table > p.title,
div.procedure > p.title,
div.equation > p.title {
    color: #394986;
    font-weight: bold;
}
```

Followup exercise

245. Tweaking Docbook's default CSS.

Overview

Software independent considerations

➟ Docbook

➟ Styling the editor application

Styling the editor

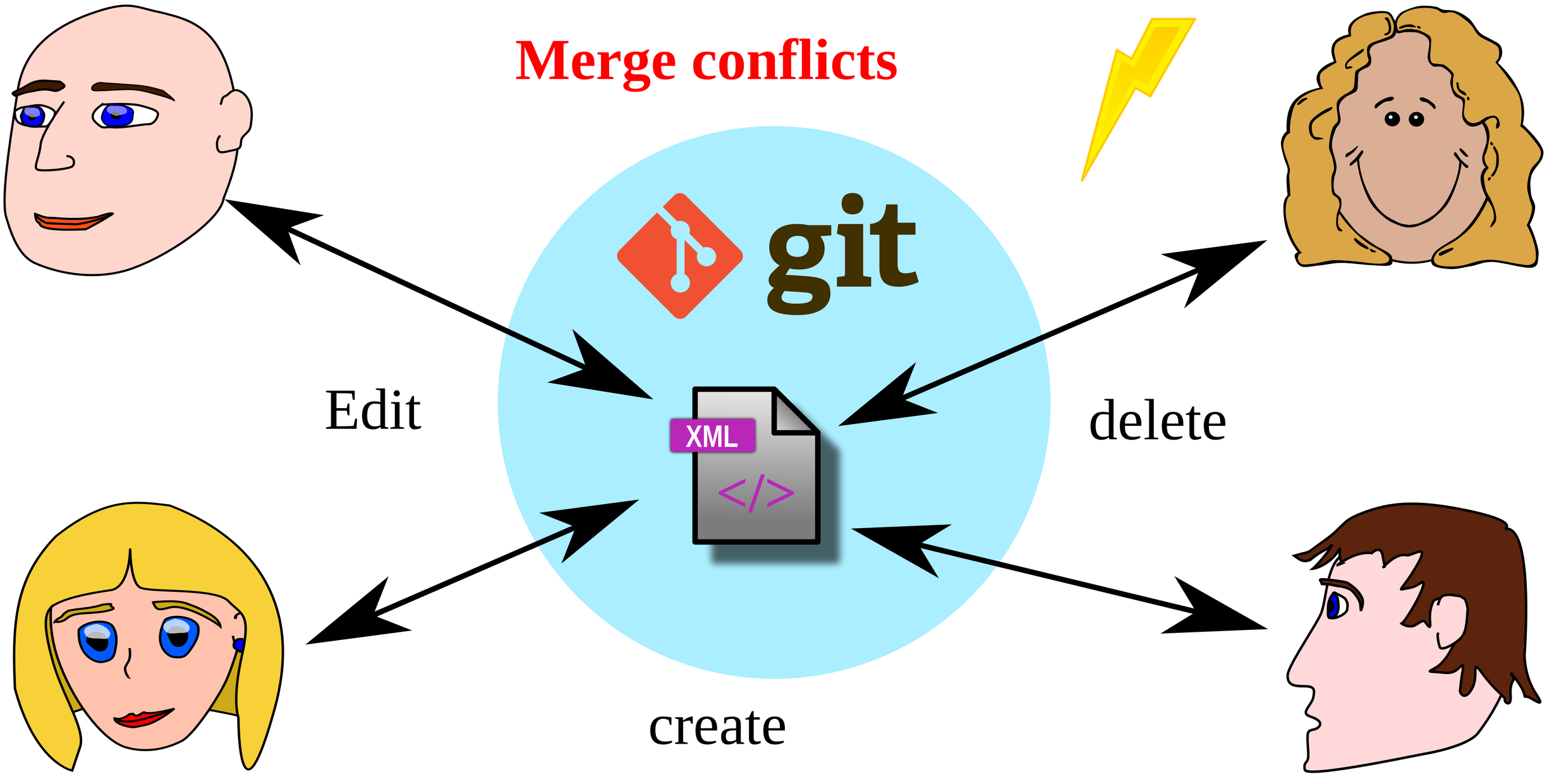
- CSS
- Plugins e.g. representing tables.
- Folding mode by CSS.

Overview

Software independent considerations

➟ Modular documents

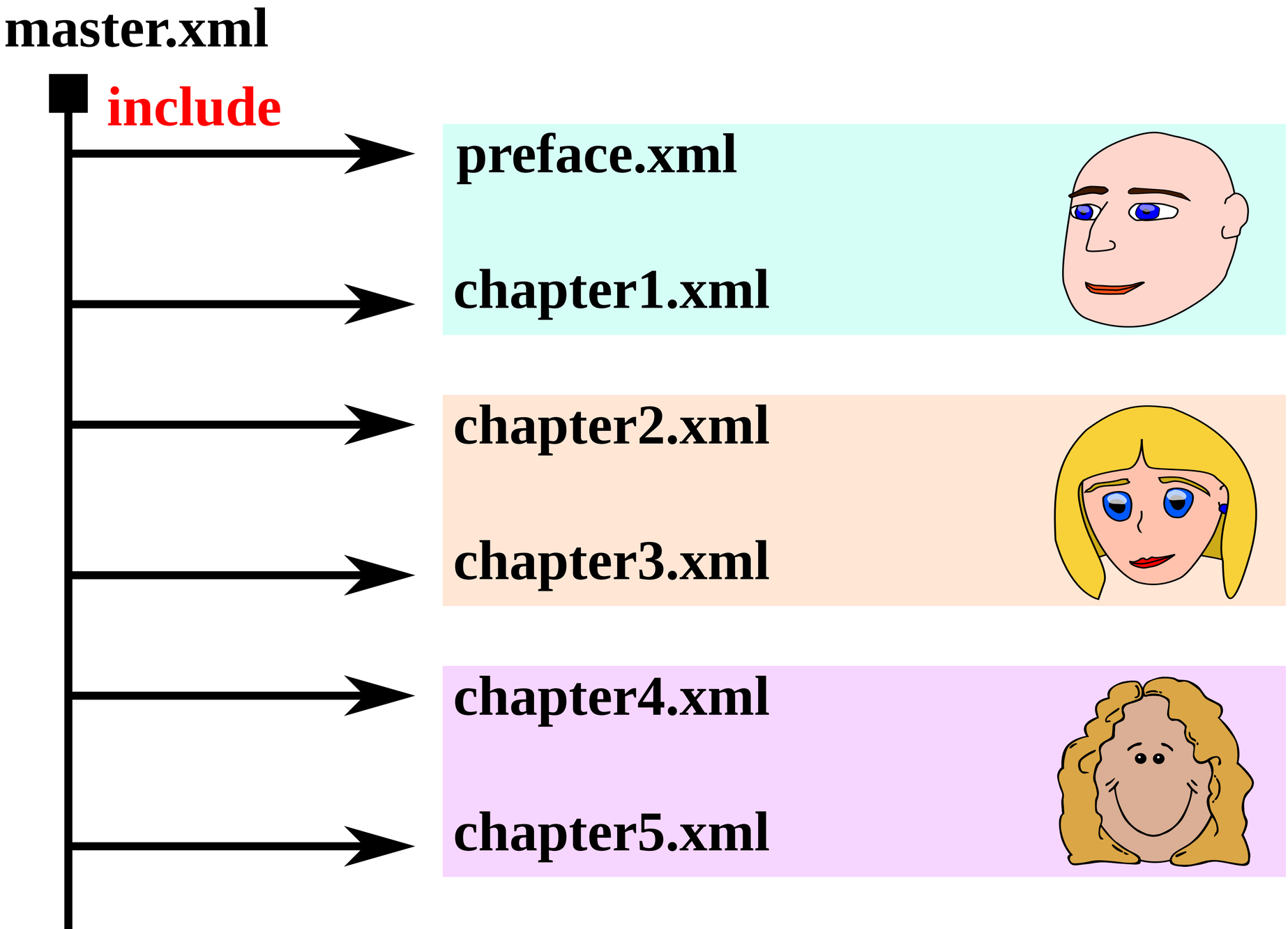
Motivating modular documents



Monolithic document problems

- Multiple author editing conflicts
- User interface limits
- No document component reuse

Document decomposition



A monolithic document

```
<book version="5.1"
  xmlns="http://docbook.org/ns/docbook">
  <chapter version="5.1" xml:id="start">
    <title>Start</title>
    <para>See <xref linkend="intro" /> (1).</para>
  </chapter>
  <chapter xml:id="intro" (2)>
    <title>Introduction</title>
    <para>Basic stuff.</para>
  </chapter>
</book>
```

(1) An internal link.

(2) Internal link target.

Decomposing documents

master.xml

```
<book version="5.1" (1)
  xmlns="http://docbook.org/ns/docbook"
  xmlns:xi="http://www.w3.org/2001/XInclude"> (2)
  <xi:include href="start.xml" (3)
     xpointer="element(/1)"/> (4)

  <xi:include href="intro.xml" (5)
     xpointer="element(/1)"/> (6)
</book>
```

start.xml

```
<chapter version="5.1" (1)
xmlns="http://docbook.org/ns/docbook">
  <title>Start</title>
  <para>See
     <xref linkend="intro"/>.</para>
</chapter>
```

intro.xml

```
<chapter version="5.1" (1)
xmlns="http://docbook.org/ns/docbook">
<title>Introduction</title>
  <para>Basic stuff.</para>
</chapter>
```

Followup exercise

246. Internal links and modular documents

Overview

Software independent considerations

➟ RelaxNG Schema

XML grammar defining languages

1. **RE**gular **LA**nguage for **X**ML **N**ext **G**eneration (RelaxNG)
2. Schematron
3. XML Schema (XSD)
4. **D**ocument **T**ype **D**efinition (DTD)

Address list schema

Schema

```
<element name="aBook">
  <zeroOrMore>
    <element name="person">
      <element name="fullName"
        <text/>
      </element>
      <element name="email">
        <text/>
      </element>
    </element>
  </zeroOrMore>
</element>
```

Doc instance

```
<aBook>
  <person>
    <fullName>Jim Bone</fullName>
    <email>bone@mycity.com</email
  </person>
</aBook>
```

Followup exercise

247. Inventing a <book> grammar

Overview

- Software independent considerations
  - Transforming documents
    - Target format HTML

Format conversion problem

Problem regarding Figure 671, “Single source publishing”:

```
<book version="5.1" ...>
  ...
  <chapter>
    <title>Introduction</title
    <para>First section.</para
  </chapter> ...
</book>
```

```
<html>
  <head>...</head>
  <body>
    <h1>Introduction</h1>
    <p>First section.</p> ..
  </body>
</html>
```

XSL template rules

```
<xsl:template match="/book">
  <html>
    <head> ... </head>
    <body>
      <h1>
        <xsl:value-of select="title"/>
      </h1>
    </body>
  </html>
</xsl:template>
```

Example: Formatting `<title>` elements

```
<xsl:template match="title">
  <h1>
    <xsl:value-of select="."/
  </h1>
</xsl:template>
```

```
<title>Some content</title>
```

gets converted to:

```
<h1>Some content</h1>
```

Followup exercises

248. Formatting <book> instances
249. Providing red background indicating foreign phrases
250. Splitting your document into chunks

Overview

- Software independent considerations
  - Transforming documents
    - Target format print

Basic FO introduction

- Further reading starting from Online and print versions.
- “Hello, world ...” style sample FO document.

Followup exercises