About Multimodal ZVON

Jiri Jirat [Jiri.Jirat@idoox.com]
ZVON project, http://zvon.org

Formats: HTML , PDF

Table of contents

List of figures

Abstract: Multimodal ZVON is a demonstration of a site powered by XML/XSLT technology. The same XML sources have been used to create several different output formats (including graphics). Moreover, a sophisticated search and a site map have been very easily implemented, since all sources are XML.

Keywords: XML, XSLT, CSS, PDF

Introduction [Go top]

The "Multimodal view of ZVON" is a demonstration project which shows how XML with XSLT can be used to create and maintain a website. Multiauthoring is very easy, thanks to the transparency of the XML format. Now the website is completely built using XML and in the following few pages we will briefly describe the framework.
The whole process consists of the following main steps:

- Defining data layout - storage of different data types in XML files, directory and data structure definition.
- Creating presentation layout: pictures (SVG), various text formats (HTML, XML, PDF)
- Using secondary information (metadata) - creating specialized search and a site map.

Data layout [Go top]

Storing metadata (ZVON metadata) [Go top]

Metadata about each item (reference, tutorial, ...) are stored in one central file. These metadata contain:

- General description - short titles, type of item
- History data - dates and description of changes (announcements)
- Address (URL)
- Keywords for database searching (Glossary and Referrers entries)
- Keywords for index

Here is an example of such a tree fragment:

```
<zvonItem id="z20010208" defaultGlossary="DOM" defaultPriority="5" referrers="DOM2+Reference"
download="yes">
  <title>
    <frontPage/>
  </title>
  <shortTitle>DOM2 Reference</shortTitle>
  <url>
    <basePath download="yes" mime="text/html"
 server="http://www.zvon.org">/xxl/DOM2reference/</basePath>
    <startFile mime="text/html">Output/index.html</startFile>
    <alias mime="text/html">/xxl/DOM2reference/Output/index.html</alias>
  </url>
  <announce year="2000" month="02" day="08">
    <shortDescription> - complete DOM2 core reference</shortDescription>
    <description>
      <frontPage/>
    </description>
  </announce>
  <type idrefs="reference"/>
  <category idrefs="xml html"/>
  <priority value="5" idrefs="xml html"/>
  <index>
    <keyword>DOM</keyword>
    <keyword>JavaScript</keyword>
  </index>
</zvonItem>
```

(X)HTML fragments [Go top]

We need to use some HTML fragments (or, better say, some fragments from HTML namespace), because XML does not support forms yet. We keep these fragments in an XML file, from where they can be easily copied into the output file. The following example shows a stored fragment with a form for subscribing to the ZVON monthly newsletter:

```
<box id="newsletter" phpfile="site/mailreg_en.php">
  <table class="box" xmlns="http://www.w3.org/TR/REC-html40">
    <tr>
      <td align="center">
 Free monthly newsletter
        <form action="site/news_reg.php" method="post">
          <input type="text" name="email" size="15" value="your email"/>
          <input type="submit" value="go"/>
        </form>
      </td>
    </tr>
  </table>
</box>
```

Defining the page layout [Go top]

Once the metadata and fragments have been written, the composition of the page can be described by only a few tags. The code below defines the layout for the References page:

```
<page id="references" mime="html" title="References" old_php_id="17">
  <section>
    <menu idref="top"/>
    <menu idref="main"/>
  </section>
  <section>
    <box id="search"/>
    <box id="links"/>
    <box id="newsletter"/>
```

```
  </section>
  <section>
    <items type="reference">
      <group category="xml"/>
      <group category="html"/>
      <group category="other"/>
    </items>
  </section>
  <section>
    <menu idref="bottom"/>
  </section>
</page>
```

Such a structure definition is common for all output formats: HTML with CSS, XML with CSS, XML with XSLT, HTML with graphics (PNG created from SVG), HTML with SVG. The XSLT stylesheets are different.

Organizing the references [Go top]

Each reference (tutorial, ...) acts as a nearly independent unit. This brings the benefits to authors, who can organize their source data as they want and need. There are two bridges among navigation pages and each piece (reference, tutorial, ...) - ZVON metadata and the frontpage of the reference. Here is a sample frontpage:

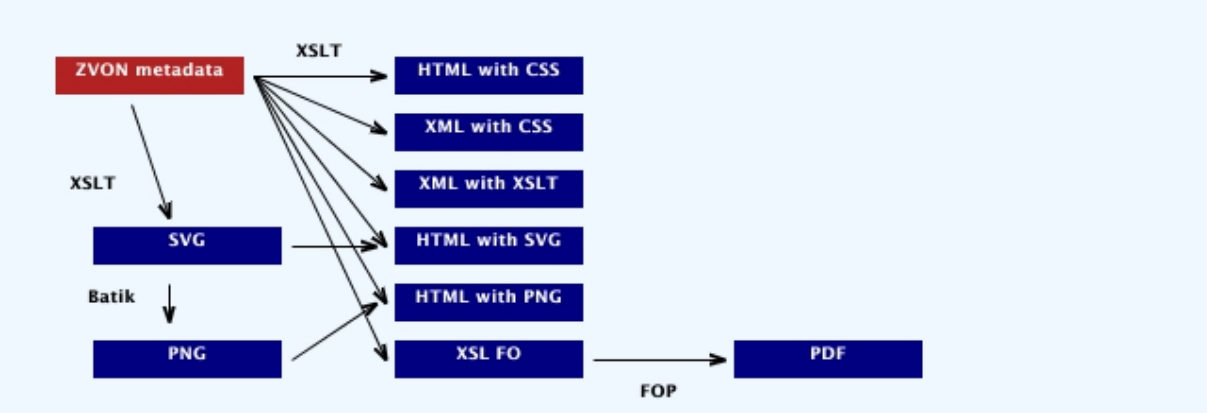
```
<frontPage xml:lang="en" xmlns:html="http://www.w3.org/TR/REC-html40" idref="z19991208">
  <head>
    <author id="nicmila"/>
    <title>XPath Tutorial</title>
  </head>
  <body>
    <section>
      <title>Introduction</title>
      <para>
 XPath is described in
        <html:a href="http://www.w3.org/TR/xpath">XPath 1.0 standard</html:a>
 . In this tutorial selected XPath features are demonstrated on many examples.
      </para>
    </section>
    <section>
      <para>You can start from :</para>
      <list>
        <item>
          <html:a href="../Output/example1.html">Example 1</html:a>
        </item>
      </list>
      <para>or from the left-side panel.</para>
    </section>
    <section>
      <title>See also</title>
      <list>
        <item>
          <html:a href="http://www.zvon.org/cgi-bin/xlab/bin/index.py">XLab</html:a>
 - interactive XPath experiments
        </item>
        <item>
          <html:a href="http://zvon.org/xxl/XMLTutorial/General/book.html">XML
tutorial</html:a>
        </item>
        <item>
          <html:a href="http://zvon.org/xxl/DTDTutorial/General/book.html">DTD
tutorial</html:a>
        </item>
        <item>
          <html:a href="http://zvon.org/xxl/NamespaceTutorial/Output/index.html">XML
Namespace tutorial</html:a>
        </item>
        <item>
          <html:a href="http://zvon.org/xxl/xhtmlReference/Output/index.html">XHTML
1.0 reference</html:a>
        </item>
        <item>
          <html:a
 href="http://zvon.org/xxl/xhtmlBasicReference/Output/index.html">"XHTML Basic"
reference</html:a>
        </item>
      </list>
    </section>
    <zvonXSLT/>
    <keywords/>
    <zvon/>
    <downloads/>
  </body>
  <appendix>
    <referrer id="XPath+Tutorial"/>
    <copyright id="idoox" year="2000"/>
    <zvonLink id="zvonLink1"/>
    <licence id="FDL"/>
    <licence id="GPL"/>
    <program id="saxon"/>
```

```
    <program id="python"/>
  </appendix>
  <indexFrame upbarlink="../Output/_upbar.html">
    <links/>
    <frameset cols="250,*">
      <frame name="mainIndex" src="../Output/examples.html"/>
      <frame name="mainWindow" src="../Output/introduction.html"/>
    </frameset>
  </indexFrame>
</frontPage>
```

Presentation layout [Go top]

Figure **1** shows the process of creating a multimodal view in a shortcut. Sources are XML files, which are then transformed to several different output formats.

Figure **1**: Summary of the build process.



Creating pictures [Go top]

Thanks to SVG (XML based graphical format), it is easy to create graphics from the text (e.g. buttons, graphical date format, fancy titles, ...) using XSLT transformations (see Figure **2**). Not all browsers are able to display SVG graphics and thus it may be better to make a conversion of SVG to some classical format (PNG, JPEG). Of course, you are losing a lot of information.

Figure **2**: Creating pictures using SVG.

- XML metadata + XSLT -> SVG
- SVG -> PNG

Another advantage which SVG can bring: it is very easy to automatically determine the width and height of the picture directly from XSLT. Here is an example of XSLT code:

```
<xsl:template match="@title" mode="heading">
  <p>
    <img
 width="{document(concat('page_title_',ancestor::page/@id,'.svg'))/svg:svg/@width}"
height="{document(concat('page_title_',ancestor::page/@id,'.svg'))/svg:svg/@height}"
src="/SVGOutput/page_title_{ancestor::page/@id}.png" alt="{.}"/>
  </p>
</xsl:template>
```

Creating HTML pages [Go top]

We have produced the following HTML formats, as a representatives of possible variants:

- HTML with CSS
- HTML with graphics (PNG created from SVG)
- HTML with SVG

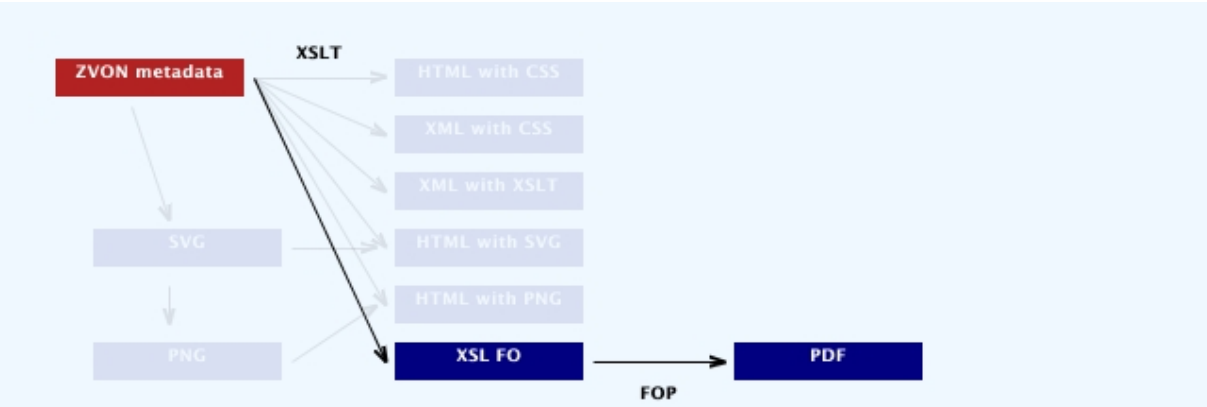
ZVON website consists not only from pure HTML files, but uses also PHP code for serving pages or connecting to databases (XML glossary, Namespace database). Thus we have used the following approach, which Eric van der Vlist [1] described in his article: Our metadata are transformed using XSLT to HTML; PHP code or (X)HTML forms are stored in separate file, from which they are extracted and then inserted into the resulting files. The process is shown in Figure **3** in detail. Note, that some parts of the PHP code (namely navigation arrays) are generated too, enabling easier maintenance and seamless integration of different parts of the site, owned by different groups (ZVON and IDOOX).

Figure **3**: Creating HTML version - detailed description.

Creating the PDF version [Go top]

Creation of the PDF version is a two-stage process. First, the XML sources are transformed to XSL FO documents and these are then converted to PDF files (using FOP), see Figure **4**.

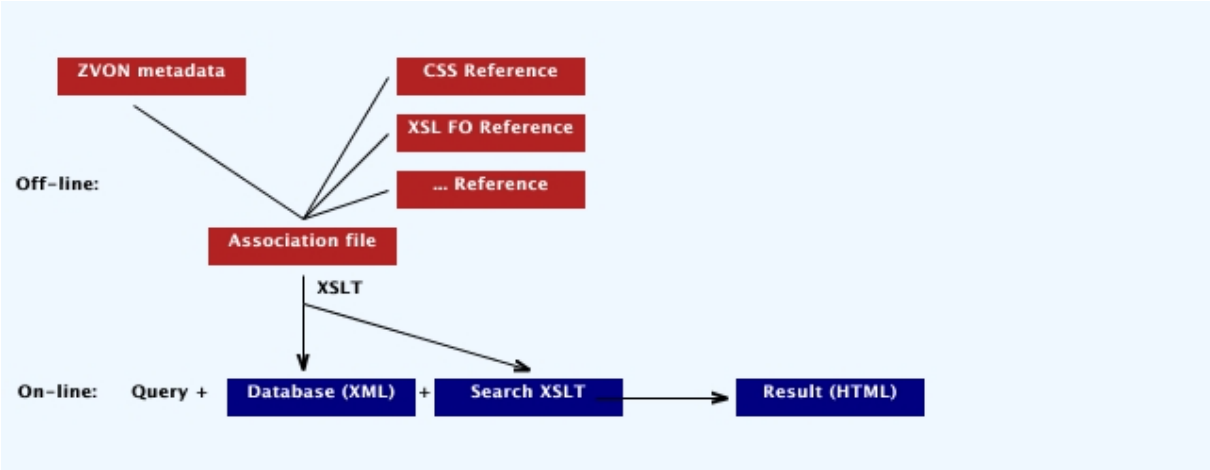
Figure **4**: Creating PDF version.



Specialized search and sitemap [Go top]

Since all references are written in XML, it was possible to build a specialized search of elements, attributes, properties etc. Figure **5** shows the process. As the references are added only from time to time, it is possible to pre-build the search database (XML format of course) and the "search engine" (XSLT stylesheet) off-line - upper part of Figure **5**. The search itself is then performed with these files using a servlet (see bottom of Figure **5**) with reasonable speed - try it yourself.

Figure **5**: Preparation of the reference search.



Metadata stored in the form of XML file are also very easily converted to a sitemap.
The following code shows that part of the association file. This fragment provides the mapping to interfaces of the DOM2 reference.

```
<group id="dom2" name="DOM 2" idref="z20010208">
  <item type="interface"
 xpath="document('../xxl/DOM2reference/DOM2/source/interfaces_XML/_interfaces.xml')
/interfaces/interface/@name" xpathToName="parent::*/@name"
 file="concat('http://www.zvon.org/xxl/DOM2reference/DOM2/Output/data/interfaces/',
parent::*/@name, '.html')"/>
...
</group>
```

As you can see, there is no further manipulation with the date once written - although the structures of XML sources are different and may be many months old, written by any author.

Technical tips and small tricks [Go top]

Serving XML files using PHP [Go top]

If you want to use PHP for serving XML files, then unlearn to use short PHP tags (**<?**, **<?=**) and disable them in your PHP configuration file. Otherwise the PHP module will be confused with XML processing instructions (e.g.: **<?xml ...>** or **<?xml-stylesheet ...>**).
The site http://www.php.net (excellent) contains information about compiling and configuring the PHP module.

Use of CVS, file sharing [Go top]

All source files are text files and the CVS (Concurrent Versions System) can be used for maintenance of the whole site (including graphics). All authors then can share XSLT

stylesheets, ZVON metadata, and scripts.

Software needed [Go top]

We use the following tools:
- Saxon (XSLT processor)
- FOP (XSL FO formatter)
- Batik (SVG viewer and transcoder)
- Any text or XML editor
- To build some references easily we use Perl or Python

References

1. Eric van der Vlist, XML.com, 2000,
http://www.xml.com/pub/a/2000/07/26/xslt/xsltstyle.html
Cited on pages: [5]

List of links

http://zvon.org
Cited on pages: [1]
http://zvon.org/index.php?nav_id=references
Cited on pages: [2]
http://zvon.org/index.php?nav_id=references&mime=html
Cited on pages: [3] , [5]
http://zvon.org/index.php?nav_id=references&mime=xml
Cited on pages: [3]
http://zvon.org/index.php?nav_id=references&mime=xml_xslt
Cited on pages: [3]
http://zvon.org/index.php?nav_id=references&mime=html_gr
Cited on pages: [3] , [5]
http://zvon.org/index.php?nav_id=references&mime=html_svg
Cited on pages: [3] , [5]
http://zvon.org/index.php?nav_id=173
Cited on pages: [5]
http://zvon.org/index.php?nav_id=172
Cited on pages: [5]
http://xml.apache.org/fop
Cited on pages: [5] , [7]
http://zvon.org:9001/saxon/cgi-bin/Search/Output/base.html?stylesheetFile=search.xsl
Cited on pages: [6]
http://zvon.org/index.php?nav_id=zvonindex
Cited on pages: [6]
http://www.zvon.org/xxl/DOM2reference/Output/index.html
Cited on pages: [6]
http://www.php.net
Cited on pages: [6]
http://www.cvshome.org/
Cited on pages: [6]
http://users.iclway.co.uk/mhkay/saxon/
Cited on pages: [7]
http://xml.apache.org/batik
Cited on pages: [7]
http://www.perl.org
Cited on pages: [7]
http://www.python.org
Cited on pages: [7]

http://www.xml.com/pub/a/2000/07/26/xslt/xsltstyle.html
Cited on pages: [7]