COIN-078B
Internet Programming with XML

XSLT (eXtensible Stylesheet Language Transformations)

Module Objectives

You'll learn...

- what XSL is and its role on XML
- the differences between XSLT and XSL-FO
- how to use XSLT to transform your XML documents from one structure to another
- how to use XSLT to reorder the output according to a specific criteria
- how to use XSLT to only display certain pieces of information

Reading Assignments:

- **Textbook:** Chapter 8

What is XSL?

XSL is an XML-based language for expressing stylesheets. Designers use an XSL stylesheet to express their intentions about how their XML documents should be presented; that is, how the source content should be styled, laid out, and paginated onto some presentation medium, such as a window in a Web browser or a hand-held device, or a set of physical pages in a catalog, report, pamphlet, or book.

It consists of two parts:

1. **XSLT** (XSL Transformations) which is a language for transforming XML documents, and
2. **XSL-FO** (XSL Formatting Objects) that consists of an XML vocabulary for specifying formatting semantics.

An XSL stylesheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an XML document that uses the formatting vocabulary.

How Does It Work?

An XSL **stylesheet processor** reads an XML document and an XSL stylesheet and produces the presentation of that XML source content in the form that was intended by the designer of that stylesheet. There are two aspects of this presentation process: first, constructing a result tree by applying a set of transformation rules to the XML source tree and second, interpreting the result tree by adhering to the formatting object namespace to produce formatted results suitable for presentation on a display, on paper, in speech, or onto other media. The first aspect is called **tree transformation** and the second is called **formatting**. The process of formatting is performed by the **formatter** - a formatting object interpreter. This formatter may simply be a rendering engine inside a browser. Figure 1 illustrates this process. (Click it for a larger diagram)

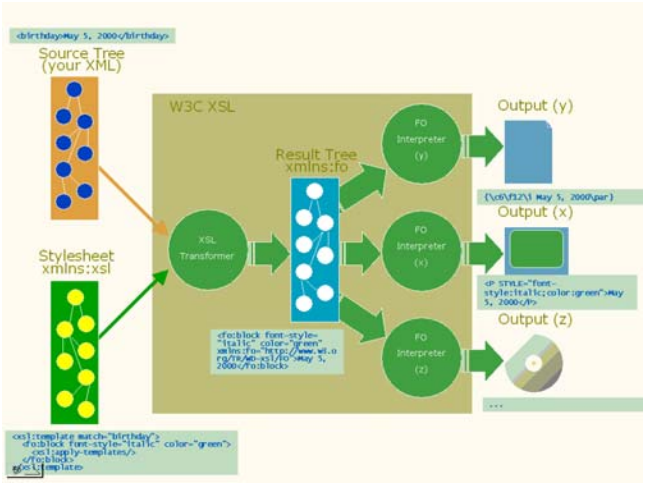

Figure 1: The two parts of the W3C XSL presentation process: First the input XML document is transformed into a result tree, and then the result tree is interpreted by a formatting object interpreter optimized for a particular output.

Microsoft's Implementation of XSL

It took years since its proposal submission in August 1997 for XSL 1.0 to become a W3C Recommendation in October 2001. The formatting object libraries, which are based on prior work on CSS and DSSSL - the Document Style Semantics & Specification Language, are complex, and many outstanding issues needed resolution. In 1998, Microsoft felt that the transformation piece of XSL was stable enough and implemented only that part of XSL from a working draft of the XSL specification. Microsoft introduced this part of XSL in the MSXML parser, which shipped with Microsoft Internet Explorer 5. This gave developers access to a mechanism that allowed general-purpose, XML-to-XML transformation language.

Although some people criticized the Microsoft XSL implementation as an incomplete part of a W3C specification, many other people used the implementation to understand the power of a declarative transformation language. As a result of this wide understanding, the W3C XSL Working Group extracted the transformation part from XSL and created a new W3C Recommendation, XSL Transformations (XSLT) Version 1.0 in November 1999. XSLT is illustrated in Figure 2. (Click it for a larger diagram)

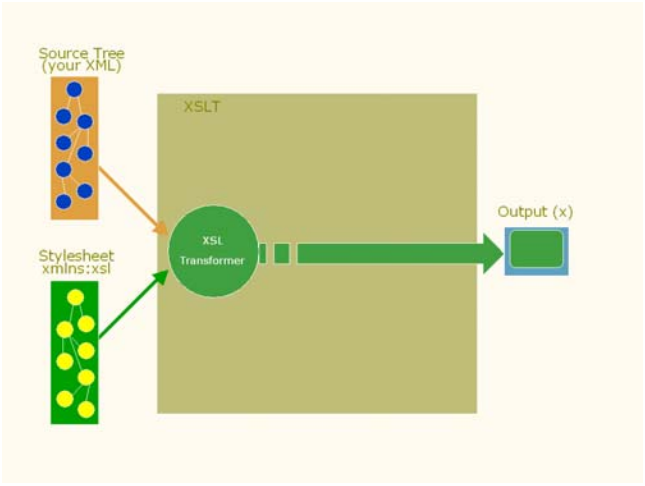

Figure 2: The XSLT model contains only the transformation part. An XSLT stylesheet directly contains information about the output device.

While XSL now points to XSLT as its transformation engine, XSL still contains all the formatting object support. XSLT requires a syntax that enables the selection of certain parts of an XML document. For example, to render a chapter title one way and a section title another way, two rules must be able to specify a path to the appropriate objects in a particular context. Because XSLT is not the only W3C standard that requires this syntax, the XML Path Language (XPath) was extracted from the XSLT specification. The W3C adopted both XPath and XSLT as Recommendations in November 1999. Two years later it recommended XSL along with XSL-FO as a complete specification.

What is XSLT?

XSLT is an event-driven, rules-based programming language, expressed in XML, that lets you create efficient programs for transforming XML documents into other formats. While XSL is a stylesheet language because it provides formatting information ability, XSLT is clearly not a stylesheet language. XSLT has nothing to do with formatting information. You can use XSLT to create a formatted representation of an information set. But, you can also use XSLT where no formatting is ever to take place. For example, in transforming a business document like an invoice or medical record, from one vendor's schema to another. XSLT can do this easily, and there is no intent to create a formatted output, in this case.

XSLT is nothing less than a programming language. But not a programming language like C or Java. XSLT is not a procedural language like ones you might be used to. XSLT is a declarative language based on rules. This makes XSLT "*rules-based*". Rules are self-contained objects that do something once they are started. But XSLT's rules are not started by a traditional function call or program invocation. XSLT rules are fired when certain events occur in the XML document being converted. This makes XSLT "*event-driven*"

XSLT rules are called "*templates*", and are specified using the `<xsl:template>` element. A template will only execute (fire) when an event occurs in the XML document being transformed. The event that the template is to catch is defined using the `match` attribute on the `xsl:template` start tag.

Each XSLT template contains processing instructions to tell the XSLT processor how to treat the object that fired the rule. Each template rule is defined by the match attribute. This is what the XSLT event-processor looks at when it is deciding which template to activate for a given event.

The XSLT namespace has the URI **http://www.w3.org/1999/XSL/Transform**.

In the following example, a simple XML document.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="xsl/assign4.xsl"?>
<weather type="Current conditions">
    <temp>76</temp>
    <wind>5</wind>
</weather>
```

The the XSLT style sheet is shown below.

```
 1  <?xml version="1.0"?>
 2  <xsl:stylesheet
 3      version="1.0"
 4      xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 5
 6      <xsl:template match="weather">
 7          <H1><xsl:value-of select="@type"/></H1>
 8          <xsl:apply-templates/>
 9              <HR/>
10      </xsl:template>
11
12      <xsl:template match="temp">
13          <LI>Temperature: <xsl:apply-templates/>&#x00B0;F</LI>
14      </xsl:template>
15
16      <xsl:template match="wind">
17          <LI>Wind: <xsl:apply-templates/>mph</LI>
18      </xsl:template>
19
20      <xsl:template match="text()">
21          <xsl:value-of select="."/>
22      </xsl:template>
23
24  </xsl:stylesheet>
```

Produces the following output.

```
<H1>Current conditions</H1>
<LI>Temperature: 76°F</LI>
<LI>Wind: 5 mph</LI>
<HR/>
```

XSLT Elements and Functions

XSLT and XPath Sample Viewer

Push vs. Pull

Push (Document Driven)

XSLT Processor "pushes" XML content to XSL document. Templates are defined for elements.

- xsl:apply-templates

Pull (Stylesheet Driven)

XSL document "pulls" content from XML document.

- xsl:for-each
- xsl:call-template
- xsl:value-of

XSLT Processors

- **Server-side**
  - "offline" build / procesing
    - Xalan* on command-line
    - Saxon** on command-line
    - Build process controlled by Ant, shell-script, etc.
  - "dynamic" processing at time of request
    - Cocoon*
    - AxKit*

  * Apache Software Foundation
  ** Michael Kay
- **Client-side**
  IE 5.5+ and NS 6+ and Mozilla have XSLT processors

Resources

- XML.com: What is XSLT?
- W3Schools: XSLT Tutorial
- XML for the World Wide Web: Chapter 10 - XSLT (eXtensible Stylesheet Language Transformation)
- W3C : Extensible Stylesheet Language (XSL) Version 1.0
- W3C : XSL Transformations (XSLT) Version 1.0
- XSL Concepts and Practical Use
- Hands-on XSL: XSL for fun and diversion
- XSLT and XPath Quick Reference
- MSDN XSLT Reference
- XSLT Quick Reference Guide
- ZVON XSLT Reference
- XSLT and XPath Sample Viewer (IE only)
- Saxon XSLT Processor

Hands-on Assignment:

- **Assignment 1** due **Week 3**