

Intelligent Internet Technologies

Lecture 10.

XSL Technology

Alexandra V. Vitko





What is XSL?

- XSL eXtensible Stylesheet Language
- XSL includes:
 - XSLT XSL Transformation
 - XSL FO XSL Formatting Objects
- Used for transformation and formatting of XML documents
- W3C standard

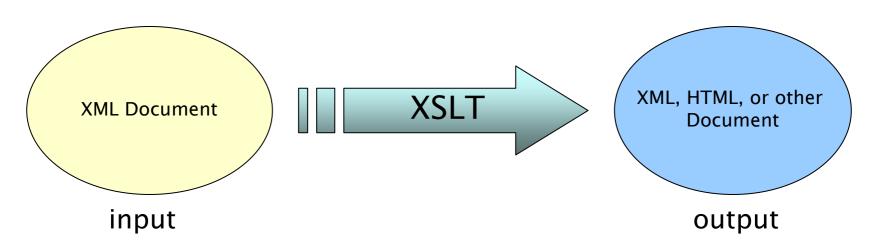


Part 1. XSLT

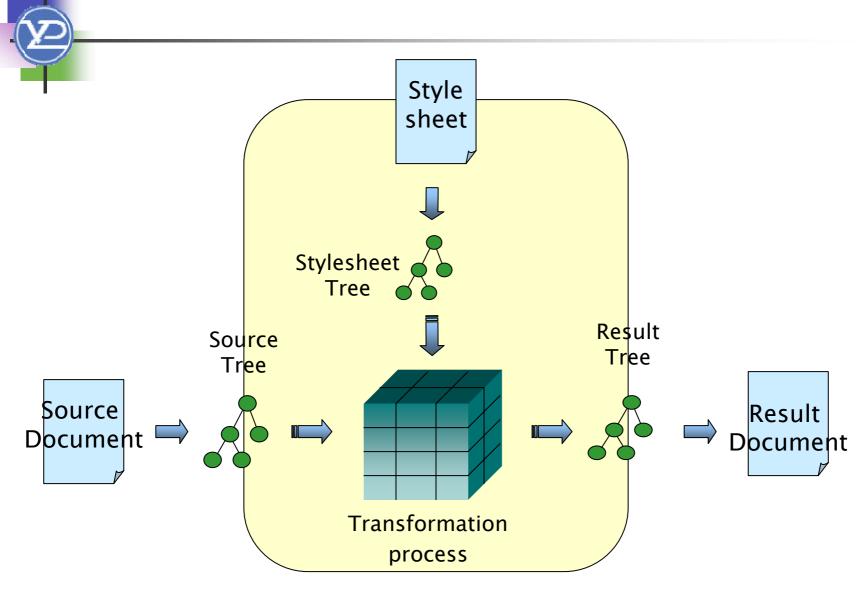


What is XSLT?

- eXtensible Stylesheet Language Transformation
- A language (an XML application) to define the appearance and behaviour of an XML document.
- Transformation: The logical restructuring of a source document to produce a result document. (For example: XML ⇒ HTML)



Transformation Process





Transformation Process (2)

 An XML parser takes the source XML document and turns it into a tree representation

The XSLT processor transforms this tree into another tree

 A serializer takes the result tree and turns it into a XML document or some other document



The XSLT Language

- XML syntax (using the xsl: namespace)
 - xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 - XSL language consists of directives (elements in this namespace)
- Rule-based
 - stylesheets consist of a series of templates that contain rules for the processing of a particular element.
 - XSL stylesheets are not a sequential programming language - rules are applied depending upon the logical structure of the document.
 - Rules may be conditional
- XSL may contain variables (numeric or string), it may perform arithmetic calculations.
- There is a library of parameterised functions

XSL



XSLT as a Programming Language

- XML Syntax
 - XSLT stylesheet = XML document
 - Use the angle brackets and tags of XML

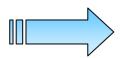
In a conventional language

In a XML language



XSLT as a Programming Language (2)

- Rule-based
 - The rules of XSLT:
 - Templates contain transformation rules
 - Either XSL directives or valid XML output
 - <xsl:template> elements in the stylesheet
 - Each rule has a match pattern & template body
- Templates are matched to an XML node.



The pattern is "price", which matches any <price>element.

→To output the value of the <price> element, formatted as a number with two decimal digits



XPath and XSLT

- XPath is a sub-language within XSLT that identifies document components.
- XPATH expressions match a template or element contents.

Example: < xsl:template match = "book/chapter" > Example XPath expressions:

- /invoice/billing-address/postcode
- .../@title
- /book/chapter[3]/section[2]/para[1]

XSL



XSLT as a Programming Language (3)

- Functional Programming
 - XSLT instructions can be executed in any order
 - Not depend on how the previous node was handled
- Data Type
 - Defined by the query language, XPath
 - Character strings, numbers(0,1..), Boolean values(T/F)
 - Variables
 - "write-once" variables, just names for values
 - hold the result of any XPath query

Y

Variables: examples

```
<xsl:variable name="x" select="//item" />
<xsl:variable name="y" select="count(@*)" />
<xsl:variable name="z" select="@type='T' " />
<xsl:variable name="tree">
      <img src="button1.gif" />
        <img src="button2.gif" />
         table>
</xsl:variable>
```

XSL



Sylesheets, Templates, Patterns

```
<xsl:stylesheet
     xmlns:xsl=http://www.w3.org/1999/XSL/Tramsform
     version="1.0">
</xsl:stylesheet>
```

<xsl:template> element : like the procedures & functions



The contents of a template

- Contents of an<xsl:template> form a template body
- XSLT instructions : a subset of XSLT elements

```
<xsl:fallback>
<xsl:apply-imports>
                               <xsl:for-each>
<xsl:apply-templates>
                               <xsl:if>
<xsl:attribute>
                               <xsl:message>
<xsl:call-tempalte>
                               <xsl:number>
<xsl:choose>
                               <xsl:<pre>cessing-
<xsl:comment>
                               instruction>
<xsl:copy>
                               <xsl:text>
<xsl:copy-of>
                               <xsl:value-of>
<xsl:element>
                               <xsl:variable>
```



Sorting

- Sorting: by specifying <xsl:sort> elements
 - <xsl:sort> element specifies a sort key and has a number of attributes to control sorting



Top-level Elements

- Can be used as children of the <xsl:stylesheet> or <xsl:transform> element
- Appear in any order, except <xsl:import> elements have to come before any others

<xsl:attribute-set>
<xsl:decimal-format>
<xsl:param>
<xsl:preserve-space>
<xsl:include>
<xsl:strip-space>
<xsl:template>
<xsl:namespace-alias>

<xsl:variable>



Part 2. XSL FO



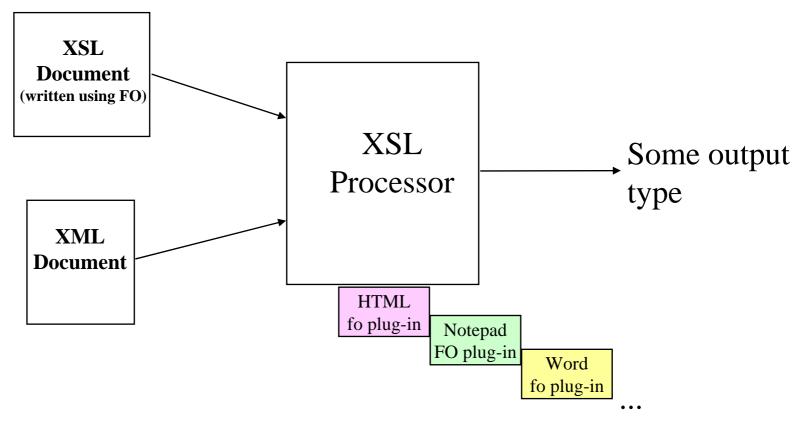
Formatting Objects (FO)

- Formatting objects are a predefined set of macros for formatting.
 - e.g., use a list FO to format a list

Note: when you use a FO you need to specify what kind of FO is desired - an HTML FO, or a paper FO, etc.

Potential of Using FO





To change the output just change the plug-in!



Why XSL FO?

- XSL-FO supports high-quality printing better than CSS+HTML/XML:
 - Multiple column layout
 - Formatting conditional on document content
 - Placement of footnotes, headers, etc., into margins
 - Automatic generation and crossreference of page numbers



XSL FO Overview

 An XSL-FO document describes the layout of a series of nested boxes, called areas.

These are placed on one or more pages.



Boxes

These look like the CSS box model: they have

- content,
- inside some padding,
- inside a border,
- inside a margin.



Pages

There is also a general page model.

Pages have

- margins all around
- region-before and region-after (i.e., headers and footers)
- region-start and region-end (columns on the side of the main content)
- region-body (main content)

XSL FO Page Regions

		top-margin		
	region-before			
l e f t - m a r g i n	region-start	region-body	r e g i o n - e n d	right-margin
		region-after		
bottom-margin				



The XSL FO Document

- fo:root
 - top node of the formatting object tree
- First describe some page masters fo:simple-page-master
 - These describe the structure of pages.
- Then provide some page sequences
 fo:page-sequence
 - These provide the content to put in the pages.

A Simple Example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<fo:root
  xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="my-page">
      <fo:region-body margin="lin"/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="my-page">
    <fo:flow flow-name="xsl-region-body">
      <fo:block>Hello, world!</fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```



Contents

- fo:flow children of fo:page-sequence specify content
 into xsl-region-body, xsl-region-start, etc.
- These generally contain a sequence of (perhaps nested) block, list-block, block-container, table, tableand-caption.

```
<fo:flow flow-name="xsl-region-body">
    <fo:block>This text will appear as a block
    </fo:block>
    <fo:block>This will be another block.</fo:block>
    ...
</fo:flow >
```



Implementations

- Various, some of them cost much
- Most popular is Apache FOP
- FOP (Formatting object to PDF) by J. Tauber
 - XML Apache project (http://xml.apache.org/fop/)
 - open-source freeware
 - Java-based XML/XSL-FO to PDF (or MIF/PCL/TXT) processor



Summary

- XSLT as transformation language for XML documents
- XPath queries within an XSLT stylesheet
- How XSLT & SQL database can work together
 - an application gets XML out of the database, and then puts that XML through XSLT stylesheet for display

Read More in

World Wide Web Consortium.

XSL Transformations (XSLT) Version 2.0

W3C Recommendation 23 January 2007

Available at http://www.w3.org/TR/xslt20

Extensible Stylesheet Language (XSL)

Version 2.0

W3C Working Draft 17 January 2012

Available at http://www.w3.org/TR/xslfo20