

CmpE 473 Internet Programming

Pınar Yolum
pinar.yolum@boun.edu.tr

Department of
Computer Engineering
Boğaziçi University

XSLT

Based largely on
Service-Oriented Computing: Semantics, Processes, Agents
– Munindar P. Singh and Michael N. Huhns, Wiley, 2004
Examples from www.w3schools.com and www.stylusstudio.com

XSLT

- A functional programming language
- Transform one XML into another format
XML, HTML, XHTML
- Add (or remove) new elements
- Rearrange or sort elements
- Use XPath to define matching pattern
- Firefox and IE has built-in support

XSLT Stylesheets

- A stylesheet specifies transformations on a document

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <main-tag>
  ...
  </main-tag>
```
- XML source tree → XML result tree
- Use the XSLT namespace, conventionally abbreviated as xsl
- Includes primitives:
 - Copy-of
 - <for-each select="...">
 - <if test="...">
 - <choose >

XSLT Templates: 1

- Match
 - A pattern to specify where a given transform should apply
 - This match only works on the root:
`<xsl:template match="/">`
...
`</xsl:template>`
- Value-of Element
 - Select the value and add it to the transformation

Fall 2007— Pinar Yolum

5

XSLT Templates: 2

- `<xsl:for-each>`
 - Apply to every node of a specified element
 - `<xsl:for-each select="catalog/cd">`
- Filtering output
 - Add extra constraint
 - `<xsl:for-each select="catalog/cd[artist='Bob Dylan']">`
 - = (equal)
 - != (not equal)
 - < (less than)
 - > (greater than)

Fall 2007— Pinar Yolum

6

XSLT Templates: 3

- Sort the selected nodes
 - `<xsl:sort select="artist"/>`
- If test: Apply based on condition

```
<xsl:if test="price &gt; 10">
<tr>
  <td> <xsl:value-of select="title"/> </td>
  <td><xsl:value-of select="artist"/></td>
</tr>
</xsl:if>
```

Fall 2007— Pinar Yolum

7

XSLT Templates: 4

- Choose (if-then-else)
 - Used with `<xsl:when>` and `<xsl:otherwise>`
- ```
<xsl:choose>
 <xsl:when test="price > 10">
 <td bgcolor="#ff00ff">
 <xsl:value-of select="artist"/></td>
 </xsl:when>
 <xsl:otherwise>
 <td><xsl:value-of select="artist"/></td>
 </xsl:otherwise>
</xsl:choose>
```

Fall 2007— Pinar Yolum

8

## XSLT Templates: 5

- Can be applied recursively on the et-children via

```
<xsl:apply-templates/>
```

- By default, if no other template matches, recursively apply to et-children of current node (ignores attributes) and to root:

```
<xsl:template match="*/">
 <xsl:apply-templates/>
</xsl:template>
```

Fall 2007—Pinar Yolum

9

## Example XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl"
 href="cdcatalog.xsl"?>
<catalog>
<cd>
 <title>Empire Burlesque</title>
 <artist>Bob Dylan</artist>
 <country>USA</country>
 <company>Columbia</company>
 <price>10.90</price>
 <year>1985</year> </cd>
</catalog>
```

Fall 2007—Pinar Yolum

## Example XSL

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
 <html>
 <body>
 <h2>My CD Collection</h2>
 <table border="1">
 <tr bgcolor="#9acd32">
 <th align="left">Title</th>
 <th align="left">Artist</th> </tr>
 <xsl:for-each select="catalog/cd">
 <tr>
 <td><xsl:value-of select="title"/></td>
 <td><xsl:value-of select="artist"/></td> </tr>
 </xsl:for-each>
 </table>
 </body>
 </html>
</xsl:template>
</xsl:stylesheet>
```

Fall 2007—Pinar Yolum