

XSLT

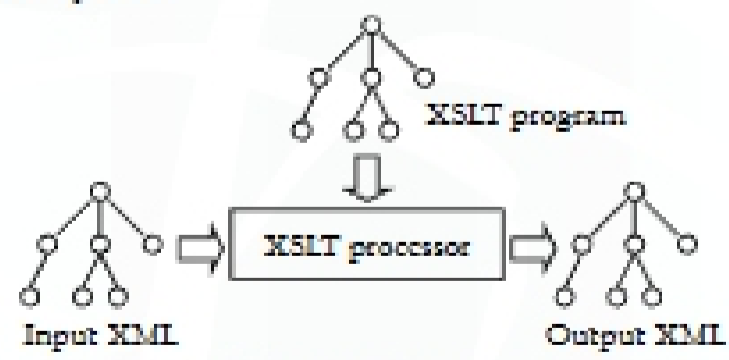
CPS 116
Introduction to Database Systems

Announcements (October 21)

- ◆ Midterm has been graded
 - Sample solution is available
 - Please verify your score on Blackboard
- ◆ Homework #2 should be graded by Thursday
- ◆ Homework #3 will be handed out on Thursday
- ◆ Project milestone #2 due in 3 weeks
 - Email feedback on milestone #1 by this weekend

XSLT

- ◆ XML-to-XML rule-based transformation language
 - Used most frequently as a stylesheet language
 - An XSLT program is an XML document itself
 - Current version is 2.0; W3C recommendation since January 2007



The diagram illustrates the XSLT transformation process. At the top, an 'XSLT program' is represented as a tree structure. An arrow points down to a box labeled 'XSLT processor'. To the left of the processor is 'Input XML' (a tree structure), and to the right is 'Output XML' (another tree structure). Arrows indicate the flow from Input XML to the processor, and from the processor to Output XML. A note at the bottom states: 'Actually, output does not need to be in XML in general'.

XSLT program

- ❖ An XSLT program is an XML document containing
 - Elements in the <xsl: namespace
 - Elements in user namespace
- ❖ The result of evaluating an XSLT program on an input XML document =
the XSLT document where each <xsl: element has been replaced with the result of its evaluation
- ❖ Basic ideas
 - Templates specify how to transform matching input nodes
 - Structural recursion applies templates to input trees recursively
- ❖ Uses XPath as a sub-language

XSLT elements

- ❖ Element describing transformation rules
 - <xsl:template>
- ❖ Elements describing rule execution control
 - <xsl:apply-templates>
 - <xsl:call-template>
- ❖ Elements describing instructions
 - <xsl:if>, <xsl:for-each>, <xsl:sort>, etc.
- ❖ Elements generating output
 - <xsl:value-of>, <xsl:attribute>, <xsl:copy-of>, <xsl:text>, etc.

XSLT example

- ❖ Find titles of books authored by "Abiteboul"

```
<?xml version="1.0"?> Standard header of an XSLT document
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="2.0">
  <xsl:template match="book[author='Abiteboul']">
    <booktitle>
      <xsl:value-of select="title"/>
    </booktitle>
  </xsl:template>
</xsl:stylesheet>
```
- ❖ Not quite; we will see why later

<xsl:template>

```
<xsl:template match="book[author='Abiteboul']">
  <booktitle>
    <xsl:value-of select="title"/>
  </booktitle>
</xsl:template>
```

- ✦ <xsl:template match="match_expr"> is the basic XSLT construct describing a transformation rule
 - match_expr is an XPath-like expression specifying which nodes this rule applies to
- ✦ <xsl:value-of select="xpath_expr"/> evaluates xpath_expr within the context of the node matching the template, and converts the result sequence to a string
- ✦ <booktitle> and </booktitle> simply get copied to the output for each node match

Template in action

```
<xsl:template match="book[author='Abiteboul']">
  <booktitle>
    <xsl:value-of select="title"/>
  </booktitle>
</xsl:template>
```

✦ Example XML fragment

```
<book ISBN="1234-10" price="10.00">
  <title>Foundations of Databases</title>
  <author>Abiteboul</author>
  <author>Silber</author>
  <author>Korth</author>
  <publisher>Addison-Wesley</publisher>
  <year>1995</year>
  <section_1/>
</book>
<book ISBN="1234-20" price="10.00">
  <title>A First Course in Databases</title>
  <author>Silber</author>
  <author>Korth</author>
  <publisher>Practice-Hall</publisher>
  <year>2002</year>
  <section_1/>
</book>
```

Template applies

```
<booktitle>
  Foundations of Databases
</booktitle>
```

Template does not apply; default behavior is to process the node recursively and print out all text nodes

```
A First Course in Databases
Silber
Korth
Practice-Hall
2002
...
```

Removing the extra output

✦ Add the following template:

```
<xsl:template match="text()|@*"/>
```

✦ This template matches all text and attributes

✦ XPath features

- text() is a node test that matches any text node
- @* matches any attribute
- | means "or" in XPath

✦ Body of the rule is empty, so all text and attributes become empty string

- This rule effectively filters out things not matched by the other rule
