XSL Transformations

Lecture 8, 07/08/02

The whole `<xsl:template>` element is a template.

- The match pattern determines where this template applies.
- Result element(s) come from non-XSL namespace(s).
- XSLT elements come from the XSL namespace.

### Document Tree

![Diagram of Document Tree]

### Default Template

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

### Text Node Template

```
<xsl:template match="TextNodeName">  
  <xsl:value-of select="text()"/>
</xsl:template>
```

### Relative XPaths

The path is relative to the element that is currently being processed in a given template.

```
<xsl:template match="poems">  
  <xsl:value-of select="poem/title"/>
</xsl:template>
```

is equivalent to

```
<xsl:template match="poems">  
  <xsl:value-of select="/poems/poem/title"/>
</xsl:template>
```
Pattern Examples

- **STANZA**
  Matches all <para> children in the current context
- **STANZA/LINE**
  Matches all <emphasis> elements that have a parent of <para>
- **/**
  Matches the root of the document
- **STANZA/LINE**
  Matches all <LINE> elements that have an ancestor of <STANZA>

Pattern Examples (contd..)

- **STANZA/LINE[1]**
  Matches the first <LINE> child of all the <STANZA> children in the current context
- **/title**
  Matches all <title> elements anywhere in the document
- **//title**
  Matches all <title> elements that are descendants of the current context

Predicates

- `<xsl:template match="LINE[contains(text(), 'brimming')]">
  <B> <xsl:apply-templates/></B>
</xsl:template>`

Matching Rules

- **node[1]**
  Matches the first node
- **node[position()=last()]**
  Matches the last node
- **node[position() mod 2 = 0]**
  Matches even nodes
- **element[@id='foo']**
  Matches the element(s) with id attributes whose value is "foo"