XPath Patterns and Expressions

Updated: 8/30/2012

Based on Goldberg, Chapter 3
Versions of XPath

• Goldberg's Chapter 3 uses XPath 1.0
• Chapter 15 introduces XPath 2.0
Templates and XPath

• A template definition uses a pattern to specify which node(s) the template can apply to.
  \[<\text{xsl:template match="pattern"/>}\]

• When a template is applied, an expression specifies the node to be processed.
  – a conditional expression may be used
  \[<\text{xsl:apply-template select="expression"/>}\]

• XPath is the language used to create patterns and expressions
Types of XPath Nodes

• root nodes
• element nodes
• text nodes
• attribute nodes
• comment nodes
• processing instruction nodes
• namespace nodes
Location Path

• Defines a hierarchical relationship that describes the locations of nodes
  – conceptually a node tree

• An **absolute location** path begins with /

• A **relative location** path is relative to the current node
  – Example: if the current node is /wonders/wonder, a relative path could be name/@language
XML Node Tree

- Root node (document node)
- Child node(s)
  - each can have its own child nodes
Determining the Current node

• The **current node** is the node specified by the currently active template
  – it is identified by the template's *match* attribute
• In a *xsl:for-each* instruction, each matched node becomes, in turn, the current node
• When *xsl:apply-templates* returns a node set, each node in the set becomes the current node
Node References

- Current node: . (dot)
  - an attribute of the current node: /@name
  - all attributes of the current node: /@*
- All child nodes: * (asterisk)
- All descendants of the current node: //
- Parent node: .. (double dot)
- Sibling: ../name
- Attribute of the parent node: ../@attrib
Conditionally Selecting Nodes

• Expressions called *predicates* can be used to compare values, test for existence, and calculate (see Chapter 4)

• General format:  \textit{path[expression]} \\

• Examples:
  
  – testing for existence of an attribute named language:
    \texttt{[@language]} \\
  – selecting all attributes of nodes on the path when \textit{expression} is true:
    \texttt{path[expression]/@*}
Example

• Select all nodes that have an attribute named file:
  <xsl:apply_templates>select="//*@file"