

CmpE 473 Internet Programming

Pinar Yolum
pinar.yolum@boun.edu.tr

Department of
Computer Engineering
Boğaziçi University

XSL

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

XML Query Languages

- XSL: e**X**tensible **S**tylesheet **L**anguage
 - XPath: Defining parts of XML documents
 - XSLT: For transforming XML documents
 - XSL-FO: For formatting XML documents

XPath

- Address parts of XML documents
- Uses the logical structure of XML documents
- Model XML documents as trees with nodes
 - Element nodes
 - Attributes nodes
 - Text nodes (PCDATA)
 - Comments
 - Root node (document node)
 - Namespace nodes
 - Processing instruction nodes
 - Comment nodes

Data Model

- Parent in XPath is like parent as traditionally in computer science
- XPath basics:
 - Element nodes, comment nodes, processing instruction nodes and text nodes are children
 - An attribute is not the child of its parent
 - Makes a difference for certain kinds of recursion (e.g., apply-templates discussed in XSLT)

Fall 2007— Pinar Yolum

5

XPath Paths

- Path expressions to select nodes in XML document

Expression	Description
element	All child nodes of the node
/	From the root node
//	Nodes in the document from the current node that matches the selection
.	The current node
..	The parent of the current node
@	Attributes

Fall 2007— Pinar Yolum

6

XPath Paths

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book>
    <title lang="eng">Harry Potter</title>
    <price>29.99</price>
  </book>
  <book>
    <title lang="eng">Learning XML</title>
    <price>39.95</price>
  </book>
</bookstore>
```

Fall 2007— Pinar Yolum

7

Examples

- bookstore
 - Selects all child nodes of bookstore
- /bookstore
 - Selects the root element bookstore
- bookstore/book
 - Select all book elements that are children of bookstore
- //book
 - Select all book elements
- bookstore//book
 - Select all books that are under bookstore
- //@lang
 - Select all attributes with name lang

Fall 2007— Pinar Yolum

8

XPath Navigation

- Predicates used to find specific nodes (with a specific value)
- Predicates are embedded in square brackets
- Select children according to position, e.g., [j], where j could be 1 ... last()
- //title[@lang]: All titles with attribute lang
- /bookstore/book[price>35.00]: All the book elements of bookstore with price>35
- Wildcard, *:
 - * matches any element node
 - @*: finds all attribute values
- AND
 - //book/title | //book/price: Title AND price elements of all book elements

Fall 2007— Pinar Yolum

XPath Queries

- Incorporate selection conditions in XPath
 - Attributes: //Song[@genre="jazz"]
 - Elements: //Song[starts-with(./group, "Led")]
 - Existence of attribute: //Song[@genre]
 - Existence of subelement: //Song[group]
 - Boolean operators: and, not, or
 - Set operator: union (); none others
 - Arithmetic operators: >, <, ...
 - String functions: contains(), concat(), length(),
 - Aggregates: sum(), count()
- Example:
 - http://www.w3schools.com/xpath/xpath_examples.asp
 - http://www.w3schools.com/xpath/tryit.asp?filename=try_xpath_select_cdnodes

Fall 2007— Pinar Yolum

10

XPath Axes

- Define a set of nodes based on the current node
- Select operations
 - Ancestor, child, descendant, parent (all ancestors/children/descendants/parent)
 - Attribute (all attributes of the current node)
 - Following (all nodes following the closing tag)
 - Preceding (all nodes before the opening tag)
 - Self (current node)
- axisname::nodetest[predicate]

Fall 2007— Pinar Yolum

11

XPath Axes Examples

child::book	Selects all book nodes that are children of the current node
child::*	Selects all children of the current node
attribute::*	Selects all attributes of the current node
ancestor::book	Selects all book ancestors of the current node
child::*/child::price	Selects all price grandchildren of the current node

Fall 2007— Pinar Yolum

12

XQuery

- Querying language for XML (what SQL is for databases)
- Uses XPath to find nodes
- FLWOR (pronounced flower) Expressions for detailed queries

Fall 2007— Pinar Yolum

13

FLWOR Expressions (1)

- For
 - for \$v in \$doc//video return \$v
 - Return all videos in doc
- Where
 - for \$v in \$doc//video
where \$v/year = 1999
return \$v/title
 - Return all videos with year 1999
 - Identical to \$doc//video[year=1999]/title

Fall 2007— Pinar Yolum

14

FLWOR Expressions (2)

- Let
 - let \$maxCredit := 3000
 - let \$overdrawnCustomers :=
//customer[overdraft > \$maxCredit]
 - return count(\$overdrawnCustomers)
 - Identical to count(//customer[overdraft > 3000])
- Order by
 - for \$x in //video
 - order by \$x/year ascending,
number(\$x/user-rating) descending
 - return \$x/title
 - Ascending year, descending user rating

Fall 2007— Pinar Yolum

15

FLWOR Expressions (3)

- Return
 - Nodes that are returned by the expression
 - for \$v in //video[genre="comedy"]
return //actor[@id = \$v/actorRef]
 - Return all actors in comedy videos

Fall 2007— Pinar Yolum

16

Conditional

- If-then-else
 - Example code
for \$x in doc("books.xml")/bookstore/book
return if (\$x/@category="CHILDREN")
then <child>{data(\$x/title)}</child>
else <adult>{data(\$x/title)}</adult>
 - Result
<adult>Everyday Italian</adult>
<child>Harry Potter</child>
<adult>Learning XML</adult>
<adult>XQuery Kick Start</adult>

Fall 2007— Pinar Yolum

17

XQuery with HTML (1)

- Embed XQuery in HTML
- Example

```
<ul> {  
  for $x in doc("books.xml")/bookstore/book/title order by $x  
  return <li>{$x}</li>  
}
```
- Returns

```
<ul>  
<li><title lang="en">Everyday Italian</title></li>  
<li><title lang="en">Harry Potter</title></li>  
<li><title lang="en">Learning XML</title></li>  
<li><title lang="en">XQuery Kick Start</title></li>  
</ul>
```

Fall 2007— Pinar Yolum

18

XQuery with HTML (2)

- Example

```
<ul> {  
  for $x in doc("books.xml")/bookstore/book/title order by $x  
  return <li>{data($x)}</li>  
}
```
- Returns

```
<ul>  
<li>Everyday Italian</li>  
<li>Harry Potter</li>  
<li>Learning XML</li>  
<li>XQuery Kick Start</li>  
</ul>
```

Fall 2007— Pinar Yolum

19

Examples

- doc("books.xml")/bookstore/book/title

```
<title lang="en">Everyday Italian</title>  
<title lang="en">Harry Potter</title>  
<title lang="en">XQuery Kick Start</title>  
<title lang="en">Learning XML</title>
```
- doc("books.xml")/bookstore/book[price<30]

```
<book category="CHILDREN">  
<title lang="en">Harry Potter</title> <author>J K.  
Rowling</author> <year>2005</year>  
<price>29.99</price>  
</book>
```

Fall 2007— Pinar Yolum

20