

# Computer Science E-259

XML with J2EE

## Lecture 4: XPath 1.0 and XSLT 1.0

23 February 2006

David J. Malan

`malan@post.harvard.edu`

1

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# Computer Science E-259

## Last Time

- DOM Level 3
- JAXP 1.3 and Xerces 2.7.1
- My First XML Parser

2

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# Last Time

## DOM Level 3

```
<!-- This is an example document -->
```

```
<students>
```

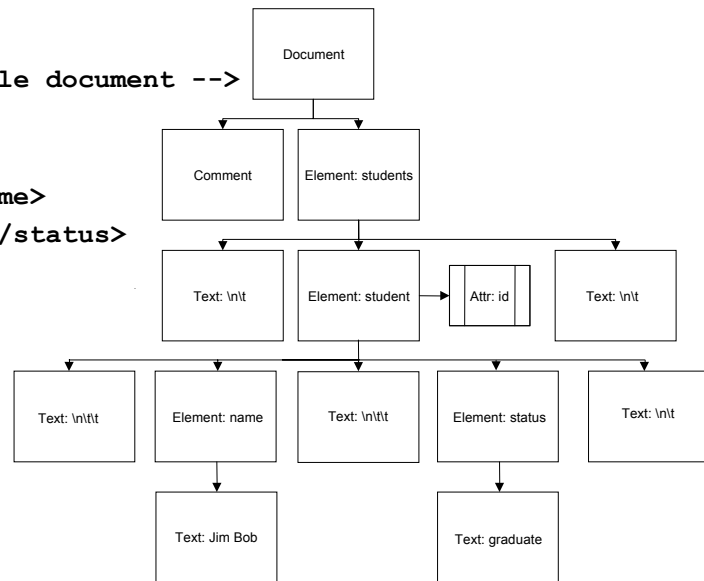
```
  <student id="0001">
```

```
    <name>Jim Bob</name>
```

```
    <status>graduate</status>
```

```
  </student>
```

```
</students>
```



3

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# Last Time

## JAXP 1.3 and Xerces 2.7.1

```
javax.xml.parsers.DocumentBuilderFactory
javax.xml.parsers.DocumentBuilder
org.w3c.dom.*
...
```

4

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# Last Time

## My First XML Parser

`cscie259.project1.mf.*`

5

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# Computer Science E-259

## This Time

- CSS Level 2
- XPath 1.0
- XSLT 1.0
- TrAX
- Project 2

6

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# CSS Level 2

## By Example

```
<?xml-stylesheet type="text/css" href="myblockbuster.css"?>
```

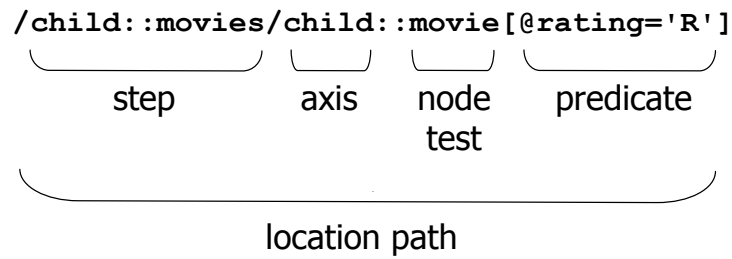
# XPath 1.0

## History

- XML Path Language (XPath) Version 1.0 is a Recommendation since 11/99
- Version 2.0 is a Candidate Recommendation since 11/05

# XPath 1.0

## Location Paths



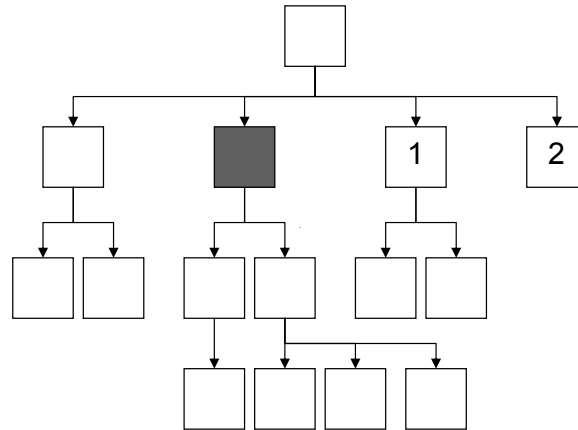
# XPath 1.0

## Axes

- ancestor, ancestor-or-self
- attribute
- child
- descendant, descendant-or-self
- following, following-sibling
- namespace
- parent
- preceding, preceding-sibling
- self

# XPath 1.0

`following-sibling`

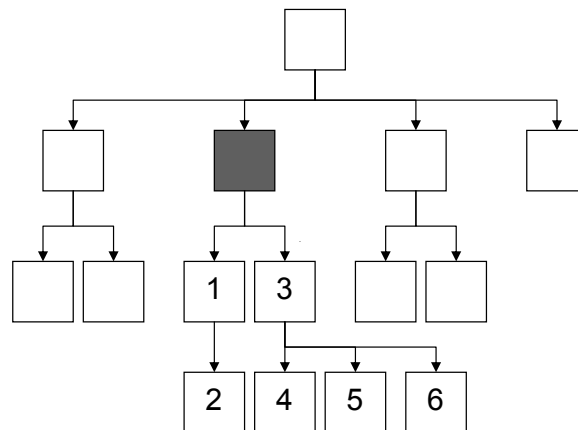


11

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# XPath 1.0

`descendant`

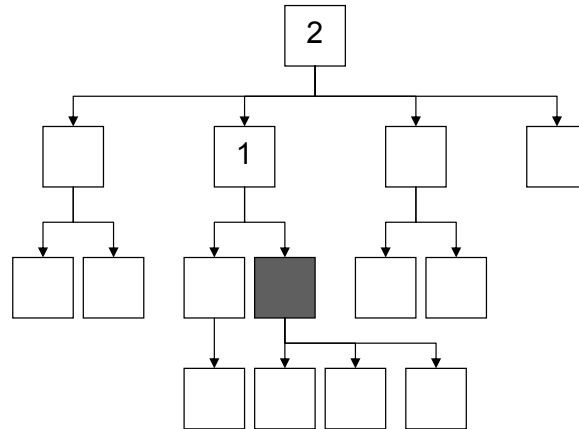


12

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# XPath 1.0

ancestor

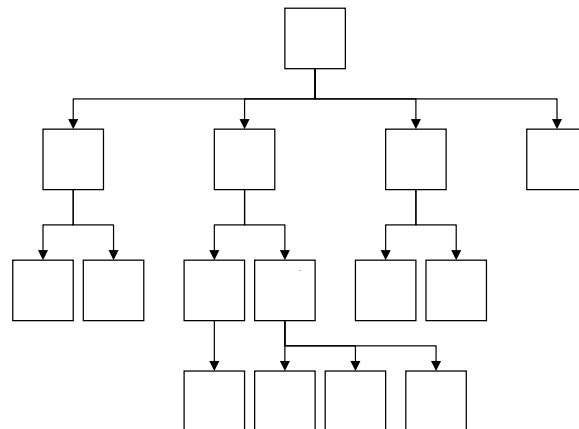


13

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# XPath 1.0

...



14

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# XPath 1.0

## Node Tests

- `foo`
- `foo:bar`
- `foo:*`
- `*`
- `node()`
- `comment()`
- `text()`
- `processing-instruction()`

15

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# XPath 1.0

## Abbreviated Syntax

- |  |   |                  |
|--|---|------------------|
| ▪ <code>child::</code>                     | ≡ | <code>' '</code> |
| ▪ <code>attribute::</code>                 | ≡ | <code>@</code>   |
| ▪ <code>/descendant-or-self::node()</code> | ≡ | <code>//</code>  |
| ▪ <code>self::node()</code>                | ≡ | <code>.</code>   |
| ▪ <code>parent::node()</code>              | ≡ | <code>..</code>  |

16

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.



# XPath 1.0

## Data Types

- boolean
- number
- string
- node-set
- external object

17

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# XPath 1.0

## boolean

- `true()`, `false()`
- `=`, `!=`, `<`, `>`, `<=`, `>=`
- `and`, `or`
- `not()`

18

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# XPath 1.0

## number

- `=, !=, <, >, <=, >=`
- `+, -, *, div, mod, -`
- `floor(), ceiling()`
- ...

# XPath 1.0

## string

- `"foo", 'foo'`
- `concat(), contains(), starts-with(), string-length(), substring(), substring-after(), substring-before(), translate()`
- ...

# XPath 1.0

## node set

- `count()`, `current()`, `last()`, `name()`, `position()`, `sum()`
- `|`
- `...`

# XPath 1.0

## Converting Types

- Explicit Conversion
  - `boolean()`, `string()`, `number()`
- Implicit Conversion
  - `false` » 0, `true` » 1
  - `false` » 'false', `true` » 'true'
  - 0 » false, other » true
  - '' » false, other » true
  - empty » false, other » true
  - ...

# XSLT 1.0

## Motivation

- XML provides a syntax for structured data formats
- No one format is likely to enable all possible uses for data
- Transforming XML can be useful in two different scenarios
  - Data conversion: transforming one format to another
    - Multiple data formats for B2B purchase orders
    - Different description formats for family trees
  - Publishing: transforming data to a human viewable form
    - Displaying XML data on the Web as HTML
    - Displaying XML data in print using PDF

23

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# XSLT 1.0

## History

- XSL Transformations (XSLT) Version 1.0 is a recommendation since 11/99 (first draft dates back to 8/98)
- Version 2.0 is a Candidate Recommendation since 11/05

24

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# XSLT 1.0

## XHTML 1.0

```
<?xml version="1.0" encoding="iso-8859-1"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:param name="foo"/>
  <xsl:output doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"
    doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"
    encoding="iso-8859-1" indent="yes" method="xml"/>
  <xsl:template match="/">
    <html>
      <head>
        <title>My First XSLT-Generated Webpage</title>
      </head>
      <body/>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

# XSLT 1.0

## Nodes

- root node
- element nodes
- attribute nodes
- text nodes
- comment nodes
- PI nodes
- ...

# XSLT 1.0

## Elements

- `xsl:stylesheet`
- `xsl:apply-templates`, `xsl:call-template`,  
`xsl:template`, `xsl:with-param`
- `xsl:for-each`
- `xsl:value-of`
- ...

# XSLT 1.0

## Matching Templates

```
<xsl:template match="/">
    ...
</xsl:template>

<xsl:template match="foo">
    ...
</xsl:template>
```

# XSLT 1.0

## Named Templates

```
<xsl:template name="foo">
  <xsl:param name="bar" select="'baz'"/>
  ...
</xsl:template>

<xsl:call-template name="foo">
  <xsl:with-param name="bar" select="'not bar'"/>
  ...
</xsl:template>
```

# XSLT 1.0

## Applying Templates

```
<xsl:apply-templates select="..."/>
```

# XSLT 1.0

## Built-In Templates

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

```
<xsl:template match="text()|@*">  
  <xsl:value-of select="."/>  
</xsl:template>
```

```
<xsl:template match="comment()|processing-instruction()"/>
```

# XSLT 1.0

## Values of Nodes

```
<xsl:value-of select="...">
```



# XSLT 1.0

## Recursive Descent Processing

- Start from the root node
- Find a matching template
- Instantiate the body of the template
  - Send literal result elements (*i.e.*, non-XSLT elements) to standard output
  - Interpret XSLT elements as instructions
- In other words, `<xsl:apply-templates/>` selects nodes to which the processor recursively applies the same algorithm: match templates and instantiate their bodies

# XSLT 1.0

## Processors

- Xalan-J 2.7.0
  - `java org.apache.xalan.xslt.Process -IN foo.xml -XSL foo.xsl`
- Microsoft XML Core Services (MSXML) 3.0 (SP4)
  - `<?xml-stylesheet type="text/xsl" href="foo.xsl"?>`
- SAXON 8.6.1
- Stylus Studio 2006 XML Enterprise Edition
- XMLSpy 2006 Enterprise Edition
- ...

# XSLT 1.0

## Versus CSS Level 2

Why two stylesheet languages?

35

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# TrAX

## APIs

```
javax.xml.transform.*  
javax.xml.transform.stream.*  
javax.xml.transform.dom.*  
javax.xml.transform.sax.*
```

36

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# Project 2

## Overview

- It's B2B Time!
- My Blockbuster
- XTube

37

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# Next Time

## XPath 1.0 and XSLT 1.0, Continued

- XPath 1.0, Continued
- XSLT 1.0, Continued

38

Copyright © 2006, David J. Malan <malan@post.harvard.edu>. All Rights Reserved.

# Computer Science E-259

XML with J2EE

## Lecture 4: XPath 1.0 and XSLT 1.0

23 February 2006

David J. Malan

`malan@post.harvard.edu`