

# XSLT & XQUERY

## Introduction to

- XSLT (25 min)
- XQuery (15 min)

(24 slides)

David Nykjær, davidn@diku.dk

1

# XSL Overview

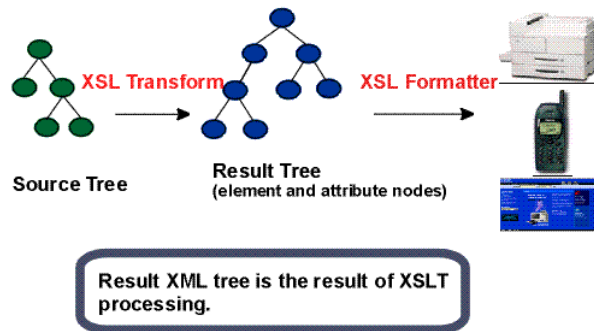
## What is XSL ?

- XSLT
- XPATH
- XSL Formatting Objects (FO)\*
  
- XSL is used for translating an XML document to another XML/HTML/XXXML format

David Nykjær, davidn@diku.dk

2

## XSL Ideal concept



David Nykjær, davidn@diku.dk

3

## XSL real-life concept

Two main purposes:

1) Generating Web pages/portals/(pda)

XML + XSL = HTML

2) Business Objects interaction

XML + XSL = (Another format) XML

David Nykjær, davidn@diku.dk

4

## XSL Example

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <body>
        <h2>My CD Collection</h2>
        <table> <tr> <th>Title</th> <th>Artist</th> </tr>
          <xsl:for-each select="catalog/cd">
            <tr>
              <td><xsl:value-of select="catalog/cd/title"/></td>
              <td><xsl:value-of select="catalog/cd/artist"/></td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

David Nykjær, davidn@diku.dk

5

## XML Document for XSL

```
<?xml version="1.0"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
  </cd>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
  </cd>
</catalog>
```

David Nykjær, davidn@diku.dk

6

# XML Example output

## My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler

David Nykjær, davidn@diku.dk

7

# XSL Commands

```
<xsl:for-each>  
  <xsl:for-each select="catalog/cd[artist='Bob Dylan']">  
    <xsl:value-of select="title"/>  
  </xsl:for-each>  
  
<xsl:sort>  
  <xsl:sort select="artist"/>  
  
<xsl:value-of>  
  <xsl:value-of select="catalog/cd[artist='Bob Dylan']/title">  
  
<xsl:copy-of>  
  <xsl:copy-of select="catalog/cd[artist='Bob Dylan']">
```

David Nykjær, davidn@diku.dk

8

## XSL Commands continued...

```
<xsl:if>
  <xsl:if test="price = 10">
    <xsl:value-of select="title"/>
  </xsl:if>

<xsl:choose>
  <xsl:choose>
    <xsl:when test="price = 10"> ... some code ...
  </xsl:when>
  <xsl:otherwise> ... some code ....
  </xsl:otherwise>
</xsl:choose>
```

David Nykjær, davidn@diku.dk

9

## XSL Templates

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html><body><h2>My CD Collection</h2>
  <xsl:apply-templates/>
</body>
</html>
</xsl:template>
<xsl:template match="cd">
  <p>
    <xsl:apply-templates select="title"/>
    <xsl:apply-templates select="artist"/>
  </p>
</xsl:template>
<xsl:template match="title">
  Title:<span style="color:#ff0000"> <xsl:value-of select="."/></span><br />
</xsl:template>
<xsl:template match="artist">
  Artist: <span style="color:#00ff00"> <xsl:value-of select="."/></span> <br />
</xsl:template>
</xsl:stylesheet>
```

David Nykjær, davidn@diku.dk

10

# XSL - Functions

## String

- substring()
- concat()

## Boolean

- not()

## Integer

- power()

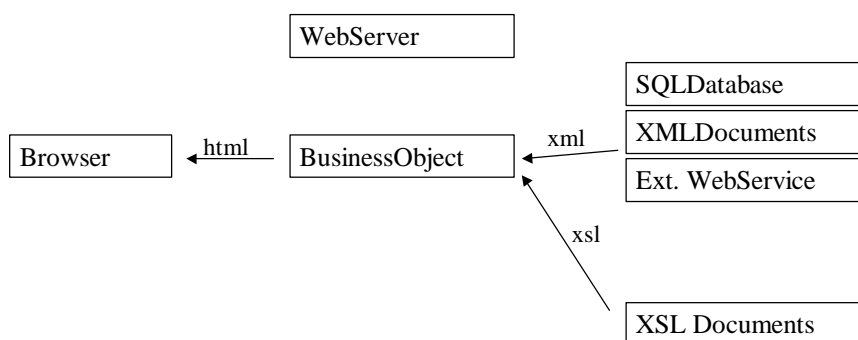
## Nodeset

- count()
- position()

David Nykjær, davidn@diku.dk

11

# XSL Real-Life Solution



David Nykjær, davidn@diku.dk

12

# XSL Questions

David Nykjær, davidn@diku.dk

13

# XQuery

What is XQuery ?

- XQuery = SQL for XML

XQuery is also an extension of Xpath:

- `document("zoo.xml")//chapter[2]//figure[caption = "Tree Frogs"]`

David Nykjær, davidn@diku.dk

14

## XQuery - FLWR (Flower)

FOR-LET-WHERE-RETURN:

```
FOR $p IN document("bib.xml")//publisher
LET $b := document("bib.xml")//book[publisher = $p]
WHERE count($b) > 100
RETURN $p
```

(generalizes SELECT-FROM-HAVING-WHERE from SQL)

## XQuery - Distinct example

Distinct:

```
FOR $p IN distinct (document("bib.xml")//publisher)
LET $b := document("bib.xml")//book[publisher = $p]
WHERE count($b) > 100
RETURN $p
```

## XQuery - join example

Join:

```
FOR $p IN document("www.irs.gov/taxpayers.xml")//person
FOR $n IN document("neighbors.xml")//neighbor[ssn = $p/ssn]
RETURN
  <person>
    <ssn> { $p/ssn } </ssn>
    { $n/name }
    <income> { $p/income } </income>
  </person>
```

David Nykjær, davidn@diku.dk

17

## XQuery - average example

Average:

```
FOR $p IN distinct(document("bib.xml")//publisher)
LET $a := avg(document("bib.xml")//book[publisher = $p]/price)
RETURN
  <publisher>
    <name>{ $p/text() }</name>
    <avgprice>{ $a }</avgprice>
  </publisher>
```

David Nykjær, davidn@diku.dk

18

## XQuery - if-then-else

### If-then-else:

```
FOR $h IN document("library.xml")/holding
RETURN
  <holding> { $h/title,
    IF ($h/@type = "Journal")
    THEN $h/editor
    ELSE $h/author }
</holding>
```

David Nykjær, davidn@diku.dk

19

## XQuery - satisfies

### Some-in-satisfies:

```
FOR $b IN document("bib.xml")/book WHERE
  SOME $p IN $b//paragraph SATISFIES
  (contains($p,"sailing") AND contains($p,"windsurfing"))
RETURN $b/title
```

### Every-in-satisfies:

```
FOR $b IN document("bib.xml")/book
WHERE EVERY $p IN $b//paragraph SATISFIES
  contains($p,"sailing")
RETURN $b/title
```

David Nykjær, davidn@diku.dk

20

## XQuery - other features

### Sort:

- `document("bib.xml")//book[price > 100]`  
`SORTBY (author[1],title)`

### Other features:

- unions, intersections, differences, subranges of lists
- lots of operators and functions

## XQuery - Missing parts ?

### Are these functionalities available ?

- Indexes ?
- Scaling issues ? (1.000.000 nodes)
- Updates ?
- Inserts ?

## XQuery - Cons & Pros

### Pros:

- + SQL like = Easy to use and readable
- + Extends Xpath
- + Advanced functionality available

### Cons:

- Why not just use SQL syntax ?
- Who uses XML-databases for data-storage ?

David Nykjær, davidn@diku.dk

23

## References

- <http://www.w3.org/TR/xlt/>
- <http://www.w3.org/TR/xquery/>
- Google: xsl tutorial
- Google: xquery tutorial

David Nykjær, davidn@diku.dk

24