0301402 INTRODUCTION TO XML

UNIT	MODULES	WEIGHTAGE
1	Introduction to XML	20 %
2	Document Type Definition (DTD)	20 %
3	XML Namespace	20 %
4	XML Schema	20 %
5	Extensible StyleSheet Language (XSL)	20 %

UNIT - 5 XSL (Extensible StyleSheet Language)

- Need of XSL
- XSL: Three Parts
- XSLT Language Characteristics
- XSLT Features
- XSL Transformation (XSLT)
- XML To HTML Transformation
- Looping
- Conditional Processing
- Numbers and Sorting

Need of XSL

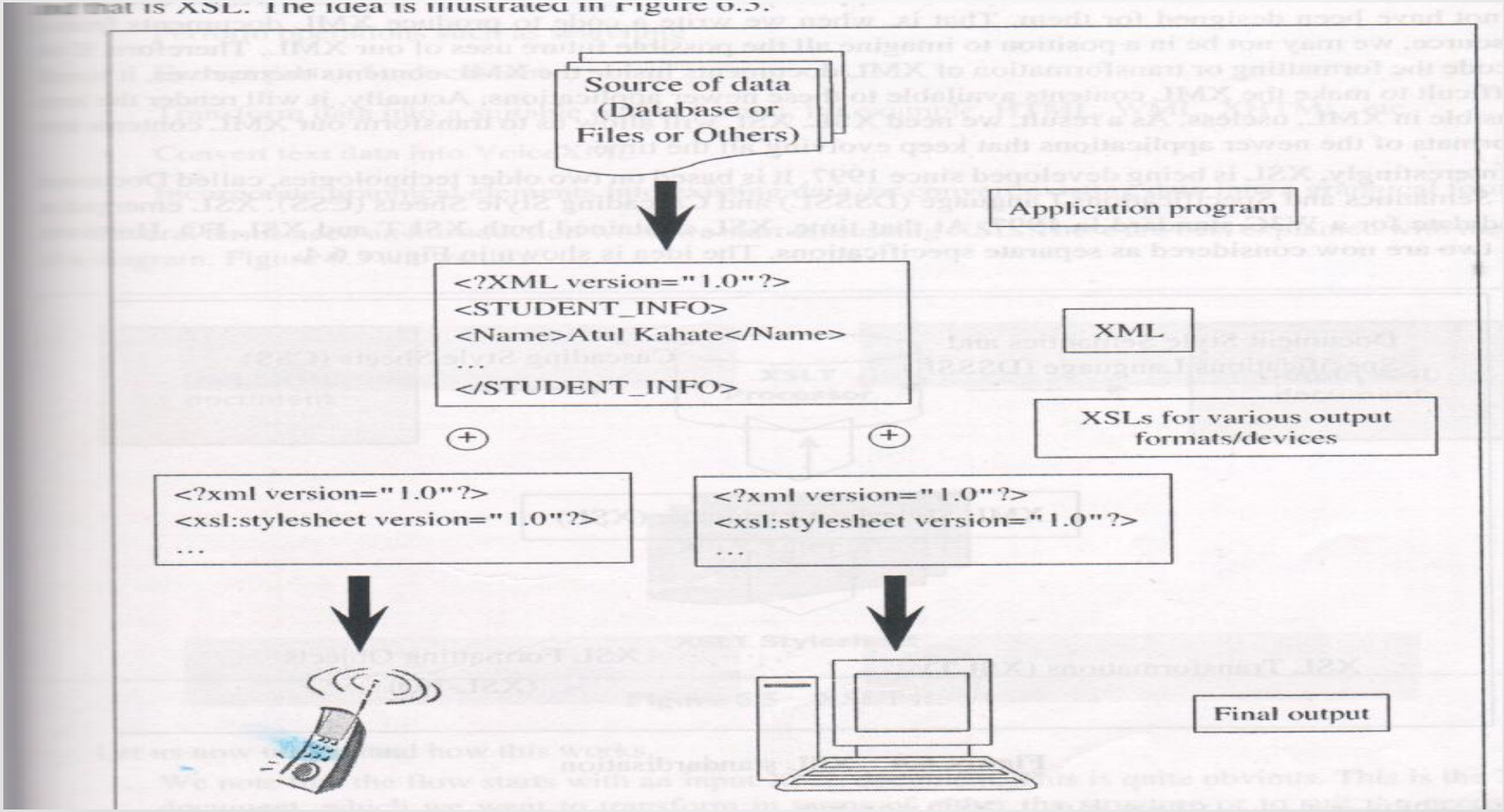
- **XSL is a language**.
- It uses stylesheets for manipulating XML documents.
- It can be said that **XSL is made up of two different** but related technologies.
  - **XSLT (XSL Transformation)**
  - **XSL-FO (XSL Formatting Object)**

Need of XSL

- XML Do
  - Capturing and storing data
  - Providing data validation facilities
  - Make data handling plaftorm independent
  - Make data exchange easier
- XSL Do
  - Transforming data from one formate to another
  - Selecting and reorganising parts of an XML document in a desired manner
  - Adding new contents to an XML document
  - Forating XML documents as per the needs

Need of XSL

- Positioning of XSL



XSL : Three Parts

- XSL Provides the action to XML . There are several technologies that fall under the XSL umbrella
  - **XSLT**
    - A language to transform XML documents
  - **XSL Formatting Object**
    - A vocabulary for formatting specifics which is part of the actual XSL specification
  - **Xpath**
    - A language for addressing any part of an XML document.

XSLT

- As the name suggest, **XSLT is used to transform XML data from one structure to another.**
- The **transformation can be with respec**t to the following:
  - Create new content
  - Add to the exsting content
  - Modify XML content into another XML format
  - Perform operations such as searching
  - Extract parts of a document
  - Transform data into suitable output format
  - Convert text data into voice XML

XSLT

- XSLT Flow

diagram. Figure 6.5 shows this.

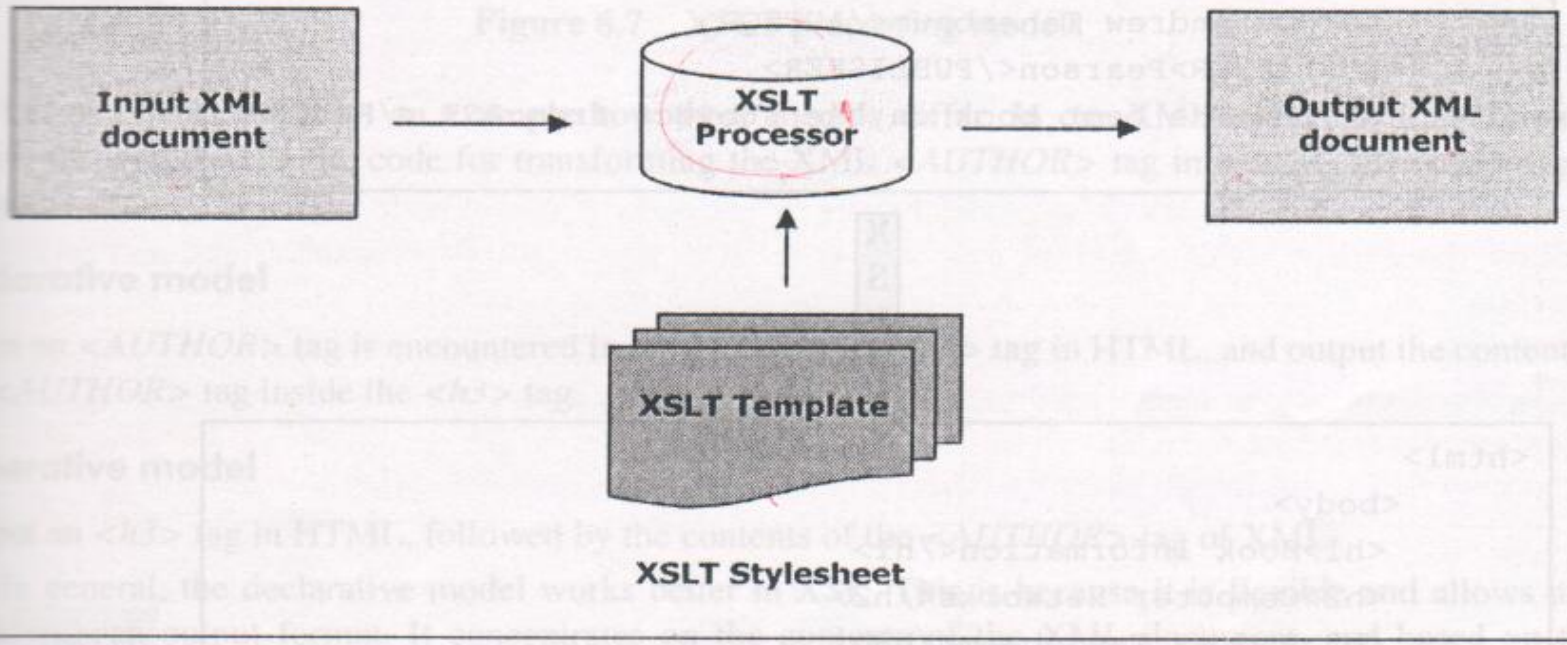


Figure 6.5 XSLT flow

XSLT

- Transformating XML into HTML

For example, suppose we have an XML document containing information about books. We want to use XSLT to transform this into HTML for displaying it on a Web browser, as shown in Figure 6.6.

```
<?xml version = "1.0"?>
<BOOK>
      <TITLE>Computer Networks</TITLE>
      <AUTHOR>Andrew Tanenbaum</AUTHOR>
      <PUBLISHER>Pearson</PUBLISHER>
      <REVIEW>The best book on the subject from AST </REVIEW>
</BOOK>
```

XML

XSLT

```
<html>
      <body>
       <h1>Book Information</h1>
       <h2>Computer Networks</h2>
       <h3>Andrew Tanenbaum</h3>
       <h4>Pearson</h4>
       <p>The best book on the subject from AST</p>
      </body>
</html>
```

HTML

XSLT Language Characteristics

- XSLT uses the XML syntax
- No side- effects
- Pattern matcing rule based / declarative
  - A stylesheet is a sequence of template rules each of which says how a particular node should be processed.
  - No ordering is needed for the template rules.
- Closure
  - Means the output has the same data structure as the input
- Recursive
  - XSLT supports recursion with built-in constructs

XSLT Features

- Multiple input sources
- Abilit to select document fragments using Xpath expressions
- Named and / or pattern based templetes
- Parameterized templates.

XSL Transformation - Template

- XSLT is used to transform one XML document from one from to another.
- **Templates**
  - An XSLT document is an XSLT document which has:
    - **A root element called Stylesheet**
    - **A file extension of .xsl**

XSL Transformation - Template

- **Converting XML to HTML**
  - In XML doc. **We would need to specify that we want to make use of a specific XSLT doc.**
  - **XSLT doc would contain appropriate rules to display the contents** of the above XML document in the HTML formte.
  - **To view the outcome, we need to open our source XML in a Web browser**. The web browser would applay the XSLT stylesheet to the XML document and show us the output in the desired HTML formate.

XSL Transformation - Template

- **Demo**
  - Book.xml
  - Book.xsl
  - two.xml
  - two.xsl
  -
  - Three.xml
  - Three.xsl

Creating Elements and Attributes

- We can convernt One formate of XML document to another formate XML document using XSL.
- For this we can creat new element and attributes using XSL.

Creating Elements and Attributes

- **Formmate ONE XML FILE**

<students>

<student first_name*="raju">*

<id>*101</id>*

<remarks> *A student who is not at all sincere! </remarks>*

</student>

</students>

Creating Elements and Attributes

- **Formmate TWO XML FILE**

  <raju id="101">

  <notes> *A student who is not at all sincere! </notes>*

  </raju>

- ***DEMO***
  - *Student.xml*
  - *Student.xsl*

Looping using<xsl:for-each>

- The XSLT <xsl:for-each> syntax is used to loop through an XML document.
- It allows us to embed one template inside another.
- It can act as an alternative to an <xsl:apply-templates> syntax.
- **DEMO**
  - **Customer.xml**
  - **Customer.xsl**
  - **Customer1.xsl**

Looping using<xsl:for-each>

- **DEMO**
  - **portfolio.xml**
  - **portfolio.xsl**
  - **portfolio1.xsl**

sorting using<xsl:sort>

- XSLT provides features for sorting of elements and attributes.
- Syntax

  <xsl:sort select =”@id” order=”ascending”/>
- DEMO
  - customer_sort.xml
  - customer_sort.xsl

Conditional processing<xsl:if>

- The <xsl:if> statement is used to specify the condition, based on which the processing will happen.

  <xsl:if test=condition>

  </xsl:if>

- The <xsl:value-of> statement is used to select the value of a particular element.

Conditional processing<xsl:if>

- DEMO
  - portfolio_if.xml
  - portfolio_if.xsl
  - portfolio_if_sort.xsl

  - performance_nest_if.xml
  - performance_nest_if.xsl

Numbers <xsl:number>

- <xsl:number> is used to:
  - Allocate a sequential number to the current node.
  - Formate number
- <xsl:number level=”lvl” value=”nbrExp” count=”node” format=”nbrFmt”/>
  - Level = attribute determines depth
  - Value= attribute determines the numbering value
  - Count= attribute determines which elements are counted
  - format=attribute determines how the numbers are formatted

Numbers <xsl:number>

Attribute Value	Description
1	Use standard number
A	Use standard capital letters
a	Use standard lowercase latters
i	Use lowercase Roman numerals
I	Use capital roman numerals

Numbers <xsl:number>

- Demo
  - customer_number.xml
  - customer_number.xsl