UNIT IV

Representing Web Data

XML-Documents and Vocabularies-Versions and Declaration - Namespaces JavaScript and XML: Ajax-DOM based XML processing Event-oriented Parsing: SAX-Transforming XML Documents-Selecting XML Data:PATH-Template- based Transformations: XSLT-Displaying XML Documents in Browsers-Case Study- Rlated Technologies. Separating Programming and Presentation: JSP Technology Introduction-JSP and Servlets-Running JSP Applications Basic JSP-JavaBeans Classes and JSP-Tag Libraries and Files-Support for the Model-View-Controller Paradigm-Case Study-Related Technologies.

Representing Web Data: XML

XML

XML stands for eXtensible Markup Language, developed by W3C in 1996. XML 1.0 was officially adopted as a W3C recommendation in 1998. XML was designed to carry data, not to display data. XML is designed to be self-descriptive. XML is a subset of SGML that can define your own tags. A Meta Language and tags describe the content. XML Supports CSS, XSL, DOM.

The Difference between XML and HTML

1. HTML is about displaying information, where as XML is about carrying information. In other words, XML was created to structure, store, and transport information. HTML was designed to display the data.
2. Using XML, we can create own tags where as in HTML it is not possible instead it offers several built in tags.
3. XML is platform independent neutral and language independent.
4. XML tags and attribute names are case sensitive where as in HTML it is not.

5. XML attribute values must be single or double quoted where as in HTML it is not compulsory
6. XML elements must be properly nested
7. All XML elements must have a closing tag
8. XML is used to create new internet languages. Here are some examples:
    - WSDL for describing available web services
    - WAP and WML as markup languages for handheld devices
    - RSS languages for news feeds
    - RDF and OWL for describing resources and ontology
    - SMIL for describing multimedia for the web

Well Formed XML Documents

XML with correct syntax is "Well Formed" XML. XML validated against a DTD is "Valid" XML.

A "Well Formed" XML document must have the following correct XML syntax:

- XML documents must have a root element
- XML elements must have a closing tag
- XML tags are case sensitive
- XML elements must be properly nested
- XML attribute values must be quoted

Example for XML Document

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me! </body>
</note>
```

Xml document begins with XML declaration statement: <? xml version="1.0" encoding="ISO-8859-1"?> . The next line describes the root element of the document: <note>. This element is "the parent" of all other elements. The next 4 lines describe 4 child elements of the root: to, from, heading, and body. And finally the last line defines the end of the root element : < /note>

XML Element

An XML element is everything from (including) the element's start tag to (including) the element's end tag.

An element can contain:

- other elements
- text
- attributes
- Or a mix of all of the above...

XML vocabulary

XML vocabulary is used to define

-element and attribute names

- element content
- Semantics of elements and attributes

Some of the xml vocabularies are XHTML, RSS, XSL, DTD, and Schema

XML DTD

Document Type Definition purpose is to define the structure of an XML document. It defines the structure with a list of defined elements in the xml document.

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]
```

Where PCDATA refers parsed character data. In the above xml document the elements to, from,

heading, body carries some text, so that, these elements are declared to carry text in DTD file. This definition file is stored with .dtd extension.

XML Schema

It is an alternative to DTD to define the structure of an XML document.

<xs:element name="note">

<xs:complexType>

<xs:sequence>

<xs:element name="to" type="xs:string"/>

<xs:element name="from" type="xs:string"/>

<xs:element name="heading" type="xs:string"/>

<xs:element name="body" type="xs:string"/>

</xs:sequence>

</xs:complexType>

</xs:element>

This definition file is stored with .xsl extension.

XML DTD vs XML Schema:-

The schema has more advantages over DTD. A DTD can have two types of data in it, namely

the CDATA and the PCDATA. The CDATA is not parsed by the parser whereas the PCDATA is parsed. In a schema you can have primitive data types and custom data types like you have used in programming.

XML Parsers

An XML parser converts an XML document into an XML DOM object - which can then be manipulated with a JavaScript.

Two types of XML parsers:

– Validating Parser

• It requires document type declaration

• It generates error if document does not

– Conform with DTD and

– Meet XML validity constraints

– Non-validating Parser

• It checks well-formedness for xml document

• It can ignore external DTD

XML Namespaces

It is a collection of element and attributes names associated with an XML vocabulary. XML Namespaces provide a method to avoid element name conflicts.

XML document 1

```
<table>
<tr>
<td>Apples</td>
<td>Bananas</td>
</tr>
</table>
```

XML document2

```
<table>
<name> Coffee Table</name>
<width>80</width>
<length>120</length>
</table>
```

If these XML fragments were added together, there would be a name conflict. Both contain a <table> element, but the elements have different content and meaning. Such name conflicts in XML can easily be avoided using a name prefix as shown below:

```
<h:table>
<h:tr>
<h:td>Apples</h:td>
<h:td>Bananas</h:td>
</h:tr>
</h:table>
<f:table>
<f:name>African Coffee Table</f:name>
<f:width>80</f:width>
```

<f:length>120</f:length>

</f:table>

When using prefixes in XML, a so-called namespace for the prefix must be defined. The namespace is defined by the xmlns attribute in the start tag of an element. The namespace declaration has the following syntax.

xmlns:prefix="URI"

For example,

<h:table xmlns:h="http://www.w3.org/table">

<h:tr>

<h:td>Apples</h:td>

<h:td>Bananas</h:td>

</h:tr>

</h:table>

<f:table xmlns:f="http://www.w3.org/furniture">

<f:name>African Coffee Table</f:name>

<f:width>80</f:width>

<f:length>120</f:length>

</f:table>

Default namespace

Default xml namespace uri for elements of a document is

xmlns:http://www.w3.org/1999/xhtml

RSS:-

“RSS stands for Rich Site Summary”

RSS is a format for delivering regularly changing web content. Many news-related sites and other online publishers syndicate their content as an RSS Feed to whoever wants it.

- RSS allows you to syndicate your site content
- RSS defines an easy way to share and view headlines and content
- RSS files can be automatically updated
- RSS allows personalized views for different sites
- RSS is written in XML

- With RSS it is possible to distribute up-to-date web content from one web site to thousands of other web sites around the world.
- RSS allows fast browsing for news and updates. RSS was designed to show selected data.
- Distributing your content using RSS will involve creating one file that contains your content.
- This file will reside on your server to enable other web sites to display your channel. You can update your channel simply by updating your file.
- Without RSS, users will have to check your site daily for new updates. This may be too timeconsuming for many users. With an RSS feed, they can check your site faster using an RSS aggregator (a site or program that gathers and sorts out RSS feeds) since RSS data is small and fast-loading

RSS is useful for web sites that are updated frequently, like:

1. News sites - Lists news with title, date and descriptions
2. Companies - Lists news and new products
3. Calendars - Lists upcoming events and important days
4. Site changes - Lists changed pages or new pages

Creating an RSS File

Your first step will be to identify your file. To do this, place the following code at the top of your text file.

```
<?xml version="1.0"?>
<rss version="0.91">
```

Your next step will be to create your channel header. The "channel" tag indicates that you are beginning a new channel.

JavaScript and XML: AJAX

AJAX:

- AJAX stands for Asynchronous JavaScript and XML
- AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
- Using Ajax,

o Update a web page with new data without reloading the page

o Request data from a server after the page has loaded

o Receive data from a server after the page has loaded

o Send data to a server in the background

- AJAX allows updating parts of a web page, without reloading the whole page.

- With Ajax, web applications can also retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page.

- Ajax is combination of

o HTML

o XML

o CSS

o JavaScript

o JavaScript DOM

o XMLHttpRequest in asynchronous mode

1. Asynchronous

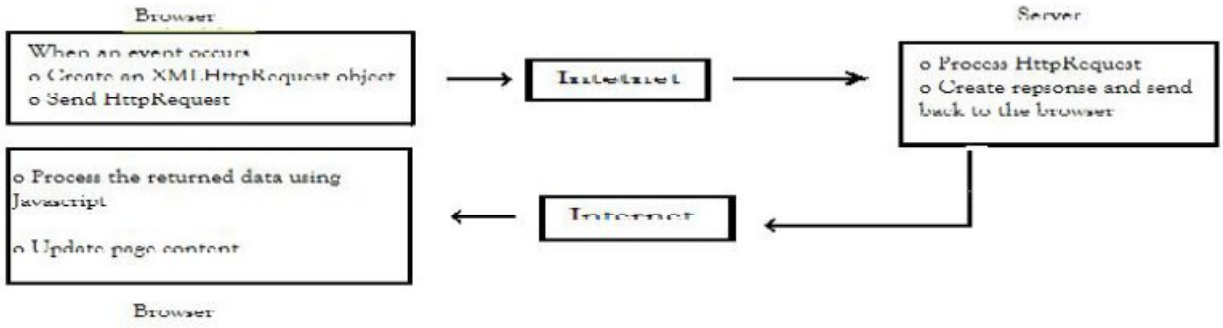
 This means that when you send a request, you wait for the response to come back, but are free to do other things while you wait. The response probably won't come back immediately, so you set up a function that will wait for the response to be sent back by the server, and react to it once that happens.

2. JavaScript

 JavaScript is used to make a request to the server. Once the response is returned by the server, you will generally use some more JavaScript to modify the current page's document object model in some way to show the user that the submission went through successfully.

3. XML

 The data that you receive back from the server will often be packaged up as a snippet of XML, so that it can be easily processed with JavaScript. This data can be anything you want, and as long as you want.



XMLHttpRequest - Constructor for other host objects allow a JavaScript program to send an HTTP request to a server and receive back a response containing an XML document. The following program illustrates the use of XMLHttpRequest.

XML DOM

Document Object Model is for defining the standard for accessing and manipulating XML documents. XML DOM is used for

 Loading the xml document

 Accessing the xml document

 Deleting the elements of xml document

 Changing the elements of xml document

According to the DOM, everything in an XML document is a **node**. It considers

 The entire document is a document node

 Every XML element is an element node

 The text in the XML elements are text nodes

 Every attribute is an attribute node

 Comments are comment nodes

DOM Levels

 Level 1 Core: W3C Recommendation, October 1998

o It has feature for primitive navigation and manipulation of XML trees

o other Level 1 features are: All HTML features

 Level 2 Core: W3C Recommendation, November 2000

o It adds Namespace support and minor new features

o other Level 2 features are: Events, Views, Style, Traversal and Range

 Level 3 Core: W3C Working Draft, April 2002

o It supports: Schemas, XPath, XSL, XSLT

We can access and parse the XML document in two ways:

o Parsing using DOM (tree based)

o Parsing using SAX (Event based)

Parsing the XML doc. using DOM methods and properties are called as tree based approach whereas using SAX (Simple Api for Xml) methods and properties are called as event based approach.

DOM based XML Parsing:(tree based)

JAXP is a tool, stands for Java Api for Xml Processing, used for accessing and manipulating xml document in a tree based manner.

In this approach, to access XML document, the document object model implementation is defined in the following packages:

javax.xml.parsers

 org.w3c.dom

The following DOM java Classes are necessary to process the XML document:

 DocumentBuilderFactory class creates the instance of DocumentBuilder.

 DocumentBuilder produces a Document (a DOM) that conforms to the DOM specification

The following methods and properties are necessary to process the XML document:

Property Meaning

nodeName Finding the name of the node

nodeValue Obtaining value of the node

parentNode To get parnet node

childNodes Obtain child nodes

attributes For getting the attributes values

Method Meaning

getElementByTagName(name) To access the element by specifying its name

appendChild(node) To insert a child node removeChild(node) To remove existing child node

An XML tree is not viewed as a data structure, but as a stream of **events** generated by the parser.

The kinds of events are:

- the **start** of the document is encountered
- the **end** of the document is encountered
- the **start tag** of an element is encountered
- the **end tag** of an element is encountered
- **character data** is encountered
- a **processing instruction** is encountered

Scanning the XML file from start to end, each event invokes a corresponding **callback** method that the programmer writes.

SAX packages

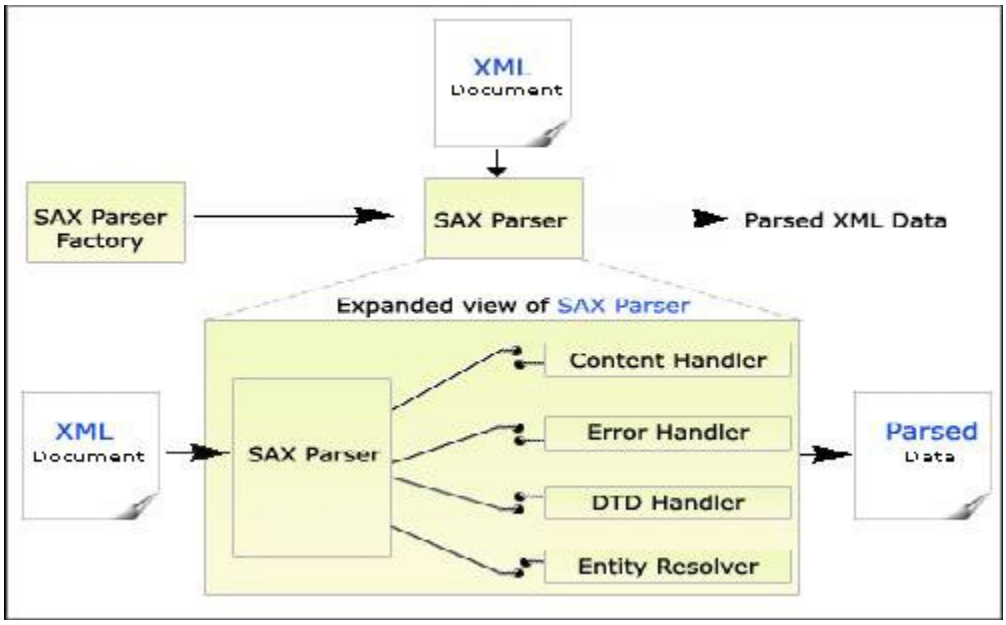
- javax.xml.parsers: Describing the main classes needed for parsing
- org.xml.sax: Describing few interfaces for parsing

SAX classes

- **SAXParser** Defines the API that wraps an XMLReader implementation class
- **SAXParserFactory** Defines a factory API that enables applications to configure and obtain a SAX based parser to parse XML documents
- **ContentHandler** Receive notification of the logical content of a document.
- **DTDHandler** Receive notification of basic DTD-related events.
- **EntityResolver** Basic interface for resolving entities.
- **ErrorHandler** Basic interface for SAX error handlers.
- **DefaultHandler** Default base class for SAX event handlers.

SAX parser methods

- **startDocument()** and **endDocument()** – methods called at the start and end of an XML document.
- **startElement()** and **endElement()** – methods called at the start and end of a document element.



XPATH

Is the syntax for specifying a collection of elements on other info contained within xml doc.

XPATH expression will be applied to parse the xml doc. which constructs a tree like representation of xml doc. similar to DOM tree.

The root of the XPath parse tree is ‘document’

Location Paths

<xsl:template match="/"> which represents the XPath document root. An XPath expression such as this represents one or more nodes within XPath parse tree is known as location path.

This location path consists a ‘location step’

Location step has two parts:

- An axis name followed by (::)
- Node test

child::Emp_Id is an example of a location step and has two parts:

Axis name specifies the direction to which we can search a node

Node test specifies an element name selected for transformation

Axis name:

The following are several axis names uses to search a particular element in the xml doc.

Name	Relationship with context node
self	The context node itself
child	Any immediate descendant
descendant	Any proper descendant
descendant-or-self	Any descendant, including the context node itself
parent	Immediate ancestor
ancestor	Any proper ancestor, including the document root (unless the context node is the document root)
ancestor-or-self	Any ancestor, including the context node itself
preceding-sibling	Any sibling of the context node that precedes the context node in the document
following-sibling	Any sibling of the context node that follows the context node in the document
attribute	Any attribute defined for the context node

Separating Programming and Presentation: JSP Technology

What is JSP?

 JSP Stands for "Java Server Pages". Using "JSP" we can use both, static HTML with dynamically-generated HTML. Web pages created using CGI programs are mostly static , dynamic part is limited to a few small locations. But using CGI and servlet , you can generate entire page through one program. Using JSP, you can built two parts separately.

JSP is the product of Sun Microsystems Inc.JSP has more advance features than Servlet . JSP separates the presentation logic from the business logic ,provide facility to developers to work separately without any trouble.JSP have the properties of Cold fusion and ASP and hence provide the flexibility to embed the business logic efficiently within the HTML content (presentation logic).

Advantages of JSP:-

 JSP is useful for server side programming

 JSP are translated and compiled into JAVA servlets but are easier to develop than JAVA servlets.

 JSP uses simplified scripting language based syntax for embedding HTML into JSP.

 JSP containers provide easy way for accessing standard objects and actions.

 JSP reaps all the benefits provided by JAVA servlets and web container environment, but they have an added advantage of being simpler and more natural program for web enabling enterprise developer

 JSP use HTTP as default request /response communication paradigm and thus make JSP ideal as Web Enabling Technology.

JSP and Servlets

• JSP documents are not executed directly

– When a JSP document is first visited,

1. first the server translates the JSP document to a servlet

2. Compiles the servlet

– Then the servlet is executed

– If any error occurs while executing, the exception will occur which gives the information for the servlet not for the JSP

– A JSP-generated servlet has a _jspService() method rather than doGet() or doPost()

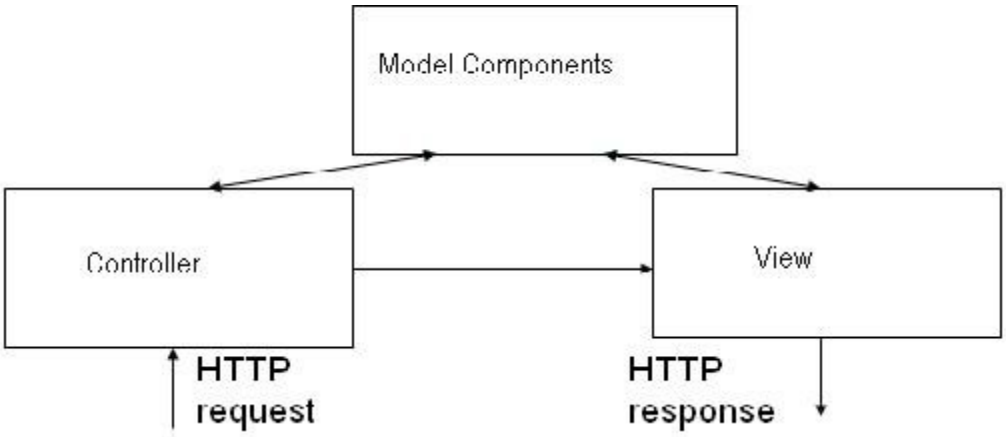
– This method begins to access the java code by creating a number of implicit objects.

MVC (Model -View - Controller)

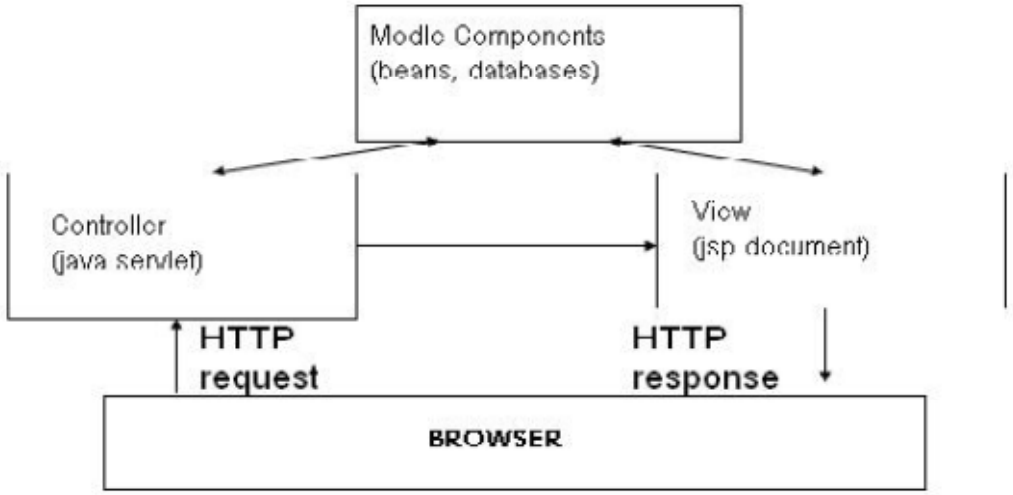
Many web applications are based on the Model-View-Controller (MVC) architecture pattern. That is, web apps contains three parts: 1) business logic 2) presentation and 3) request processing

The key motivation behind the MVC approach is the desire to separate the code that creates and manipulates the data (i.e business logic) from the code that presents the data (i.e presentation) and the code that process the request (i.e controller)

Business logic means the code applied for manipulation of application data and presentation refers the code written for look and feel of the web page such as background color, font style, placing of form controls and so on. Controller means that process the request message.



Typical JSP implementation of MVC



Advantage of using MVC

- MVC allows the developer to keep the separation between business logic, presentation and request processing. Due to this separation, any changes to the presentation can be made easy without disturbing business logic.