

XML Concepts

CSE 486/586
Sept 7, 2004

References:

<http://www.bloomington.in.us/~mongin/xml-tutorial/>
<http://studio.telme.com/general/xmlprimer.html>
<http://java.sun.com/xml/index.jsp>

Introduction

- eXtensible Markup Language (XML): De facto standard for structured documents and data on the web
- Represents a hierarchical set of entities, where an entity consists of one or more elements.
- An element can consist of attributes and/or text
`<day>Sunday</day>`
OR
`<drink name="Coffee"></drink>`
OR
`<drink name="Coffee"/>`
`<name gender="male">Frodo Baggins</name>`
- XML is case-sensitive

Elements v/s Attributes

- Elements and their properties can be described in two ways:
 - `<book itemId="999">`
`<name>Learning JAXB</name>`
`</book>`
 - `<book>`
`<itemId>999</itemId>`
`<name>Learning JAXB</name>`
`</book>`
- As a general guideline, elements correspond to nouns and attributes to adjectives
- Attributes cannot hold multiple values and cannot describe structure
- No strict rule

Introduction

- Elements may be arranged in infinitely nested hierarchy, but only one element can be designated as root document element.
- The root document element is the first element that appears in the document.
- A comment is enclosed between "`<!--`" and "`-->`". Comments may appear as a child of any element or after the root element.
- XML reserves certain characters. To express these characters in your document, use the following:
 - less than (`<`) `<`
 - greater than (`>`) `>`
 - ampersand (`&`) `&`

XML v/s HTML

- XML is a structured way of representation of an entity, whereas HTML can be called an implementation of XML that is used for the world wide web.
- Parsing is quite strict in XML. Moreover, if the DTD is supplied, unrecognized tags are rejected. On the other hand, HTML parsing is extremely forgiving.
- In XML, all tags must be closed. In HTML, some tags, such as IMG and INPUT, need not be closed.
- In XML, attributes must be associated with a value, whereas in HTML this is not necessary. (e.g., SCRIPT/@DEFER).
- Attribute values must be surrounded by quotes. This is needed in HTML only if the value contains embedded spaces.

Namespaces

- Namespaces are a simple and straightforward way to distinguish names used in XML documents, no matter where they come from.
- Example:

```
<n:html xmlns:xdc="http://www.xml.com/books"
xmlns:h="http://www.w3.org/HTML/html4"
<h:head><h:title>Book Review</h:title></h:head>
<h:body>
  <xdc:bookreview>
    <xdc:title>XML: A Primer</xdc:title>
    <h:table>
      .....
    </h:table>
  </xdc:bookreview>
</h:body>
</h:html>
```
- 'h' and 'xdc' are shortcuts to namespace URLs

Namespaces

An HTML browser may look at only the tags prefixed by "h" when parsing an XML document. Another application may just look at only the "xdc" tags.

- Namespaces can also be unnamed as follows:

```
<html xmlns="http://www.w3.org/html4"
      xmlns:xdc="http://www.xml.com/books">
<head><title>Book Review</title></head>
.....
```
- A URL is used to name namespace. They may not be real addresses. For example, "http://www.w3.org/html", "http://www.xml.com/books" are both namespaceURIs.

Schema definition

- The schema of XML files is usually defined using Languages such as:
 - Document Type Definition (.dtd)
 - W3C XML Schema Language (.xsd)
- .xsd files are more structured and complete in defining the schema over the .dtd files.
- Even individual attributes can have namespaces associated with them.

Java and XML

- Extensible Markup Language (XML) is a cross-platform, extensible, and text-based standard for representing data.
- It is a key technology in the development of Web services.
- Following are the chief components provided by Java, related to XML or that use XML:
 - Java API for XML Processing (JAXP)
 - Java API for XML Binding (JAXB)
 - Java API for XML Registries (JAXR)
 - Java API for XML-based RPC (JAX-RPC)
 - SOAP with Attachments API (SAAJ)

Java and XML

- The **Java API for XML Processing (JAXP)** enables applications to parse and transform XML documents independent of a particular XML processing implementation.
- **Java Architecture for XML Binding (JAXB)** provides a convenient way to bind an XML schema to a representation in Java code. Hence, easier to incorporate XML data and processing functions in Java applications.
- The **Java API for XML Registries (JAXR)** gives you a uniform way to use business registries that are based on open standards or industry consortium-led specifications (such as UDDI).

Java and XML

- You can use the **Java API for XML-based RPC (JAX-RPC)** to build Web applications and Web services, incorporating XML-based RPC functionality according to the SOAP 1.1 specification.
- The **SOAP with Attachments API for Java (SAAJ)** provides a standard way to send XML documents over the Internet from the Java platform.