

CSE@UTA

## Introduction to XML

CSE3302 Programming Languages Fall2007 ©Weimin He Introduction to XML 1

CSE@UTA

## What is XML?

- XML stands for eXtensible Markup Language
- A markup language much like HTML
- Simplified and extended SGML

CSE3302 Programming Languages Fall2007 ©Weimin He Introduction to XML 2

CSE@UTA

## Why Isn't HTML Enough?

- HTML
  - Focuses on presentation only
  - Has no well-defined or definable structural rules
  - Can not express rich structures and semantics
- XML
  - Describe the data
  - Self-described, semi-structured data
  - Can be queried

CSE3302 Programming Languages Fall2007 ©Weimin He Introduction to XML 3

CSE@UTA

## HTML vs. XML

HTML	XML
<code>&lt;h1&gt; Bibliography &lt;/h1&gt;</code>	<code>&lt;bibliography&gt;</code>
<code>&lt;p&gt; &lt;i&gt; Programming Languages &lt;/i&gt; Kenneth C. Louden</code>	<code>&lt;book&gt; &lt;title&gt; Programming Languages</code>
<code>&lt;br&gt; Thomson, 2003</code>	<code>&lt;/title&gt; &lt;author&gt; Kenneth C. Louden</code>
<code>&lt;p&gt; &lt;i&gt; Thinking in Java &lt;/i&gt; Bruce Eckel</code>	<code>&lt;/author&gt; &lt;publisher&gt; Thomson</code>
<code>&lt;br&gt; Prentice Hall, 2000</code>	<code>&lt;/publisher&gt; &lt;year&gt; 2003 &lt;/year&gt; &lt;/book&gt;</code>
	<code>... &lt;/bibliography&gt;</code>

CSE3302 Programming Languages Fall2007 ©Weimin He Introduction to XML 4

CSE@UTA

## A Complete XML Document

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE auction SYSTEM "auction.dtd">
<!-- This document describes auction information
on ebay website -->
<auction>
  <vendor>
    <company>ebay</company>
    <email>webmaster@ebay.com</email>
    <phone>1-800-333-4444</phone>
  </vendor>
  <item id = "item1">
    <name>Mountain Bicycle</name>
    <description>
      Used for 2 months, but looks like new
    </description>
    <seller>
      <username>bicycle seller</username>
      <feedback>99% positive</feedback>
      <email>bseller@yahoo.com</email>
    </seller>
    <payment>
      credit card, money order, check
    </payment>
    <price>65.00</price>
    <city>Dallas</city>
    <state>TX</state>
  </item>
  <item id = "item2">
    <name>Thinkpad Laptop Keyboard</name>
    <description>
      Used IBM thinkpad keyboard, good condition
    </description>
    <![CDATA[ <h1> Keyboard image</h1>
      The keyboard image is
      
    </description>
    <model>X31</model>
    <seller>
      <username>
        thinkpad factory
      </username>
      <feedback>95% positive</feedback>
      <email>thinkpad_factory@gmail.com</email>
    </seller>
    <payment> credit card, money order</payment>
    <price>50.00</price>
    <city>BROOKLYN </city>
    <state>NY</state>
  </item>
</auction>

```

CSE3302 Programming Languages Fall2007 ©Weimin He Introduction to XML 5

CSE@UTA

## XML Declaration

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE auction SYSTEM "auction.dtd">
<!-- This document describes auction information
on ebay website -->
<auction>
  <vendor>
    <company>ebay</company>
    <email>webmaster@ebay.com</email>
    <phone>1-800-333-4444</phone>
  </vendor>
  <item id = "item1">
    <name>Mountain Bicycle</name>
    <description>
      Used for 2 months, but looks like new
    </description>
    <seller>
      <username>bicycle seller</username>
      <feedback>99% positive</feedback>
      <email>bseller@yahoo.com</email>
    </seller>
    <payment>
      credit card, money order, check
    </payment>
    <price>65.00</price>
    <city>Dallas</city>
    <state>TX</state>
  </item>
  <item id = "item2">
    <name>Thinkpad Laptop Keyboard</name>
    <description>
      Used IBM thinkpad keyboard, good condition
    </description>
    <![CDATA[ <h1> Keyboard image</h1>
      The keyboard image is
      
    </description>
    <model>X31</model>
    <seller>
      <username>
        thinkpad factory
      </username>
      <feedback>95% positive</feedback>
      <email>thinkpad_factory@gmail.com</email>
    </seller>
    <payment> credit card, money order</payment>
    <price>50.00</price>
    <city>BROOKLYN </city>
    <state>NY</state>
  </item>
</auction>

```

CSE3302 Programming Languages Fall2007 ©Weimin He Introduction to XML 6

### CSE@UTA XML Document Type

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE auction SYSTEM "auction.dtd">
<!-- This document describes auction information
on ebay website -->
<auction>
  <vendor>
    <company>ebay</company>
    <email>webmaster@ebay.com</email>
    <phone>1-800-333-4444</phone>
    <item id = "item1">
      <name>Mountain Bicycle</name>
      <description>
        Used for 2 months, but looks like new
      </description>
      <seller>
        <username>bicycle seller</username>
        <feedback>99% positive</feedback>
        <email>bseller@yahoo.com</email>
      </seller>
      <payment>
        credit card, money order, check
      </payment>
      <price>65.00</price>
      <city>Dallas</city>
      <state>TX</state>
    </item>
    <item id = "item2">
      <name>Thinkpad Laptop Keyboard</name>
      <description>
        Used IBM thinkpad keyboard, good condition
      </description>
      <![CDATA[ <h1> Keyboard image</h1>
        The keyboard image is
        
      </description>
      <model>X31</model>
      <seller>
        <username>
          thinkpad factory
        </username>
        <feedback>95% positive</feedback>
        <email>thinkpad_factory@gmail.com</email>
      </seller>
      <payment> credit card, money order</payment>
      <price>50.00</price>
      <city>BROOKLYN</city>
      <state>NY</state>
    </item>
  </vendor>
</auction>
  
```

CSE3302 Programming Languages Fall2007 @Weimin He Introduction to XML 7

### CSE@UTA XML Comments

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE auction SYSTEM "auction.dtd">
<!-- This document describes auction information
on ebay website -->
<!-- This document describes auction information
on ebay website -->
<!-- This document describes auction information
on ebay website -->
<!-- This document describes auction information
on ebay website -->
  
```

CSE3302 Programming Languages Fall2007 @Weimin He Introduction to XML 8

### CSE@UTA XML Elements

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE auction SYSTEM "auction.dtd">
<!-- This document describes auction information
on ebay website -->
<!-- This document describes auction information
on ebay website -->
  
```

CSE3302 Programming Languages Fall2007 @Weimin He Introduction to XML 9

### CSE@UTA XML Attributes

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE auction SYSTEM "auction.dtd">
<!-- This document describes auction information
on ebay website -->
<!-- This document describes auction information
on ebay website -->
  
```

CSE3302 Programming Languages Fall2007 @Weimin He Introduction to XML 10

### CSE@UTA XML PCDATA

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE auction SYSTEM "auction.dtd">
<!-- This document describes auction information
on ebay website -->
<!-- This document describes auction information
on ebay website -->
  
```

CSE3302 Programming Languages Fall2007 @Weimin He Introduction to XML 11

### CSE@UTA XML CDATA

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE auction SYSTEM "auction.dtd">
<!-- This document describes auction information
on ebay website -->
<!-- This document describes auction information
on ebay website -->
  
```

CSE3302 Programming Languages Fall2007 @Weimin He Introduction to XML 12

### More XML: Entity References

- Syntax: &entityname;
- Example:
 

```
<element> this is greater than
            &gt; </element>
```
- Some entities:

&lt;	<
&gt;	>
&amp;	&
&apos;	'
&quot;	"

CSE@UTA Programming Languages Introduction to XML  
Fall2007 ©Weimin He 13

### More XML: Name Spaces

- Avoid naming collisions
- Syntax: name ::= [prefix:]localpart

**Example 1:**

```
<school:subject>Math</school:subject>
<medical:subject>Diabetes</medical:subject>
```

**Example 2:**

```
<directory xmlns = "http://cse.uta.edu/cse3302/ns-text"
           xmlns:image = "http://cse.uta.edu/cse3302/ns-image">
  <file filename = "scores.xml">
    <description> Student scores </description>
  </file>
  <image:file filename = "curve.jpg">
    <image:description>The curve of student scores</image:description>
    <image:size width = "200" height = "100"/>
  </image:file>
</directory>
```

CSE@UTA Programming Languages Introduction to XML  
Fall2007 ©Weimin He 14

### Document Type Definition (DTD)

- XML document validation
- Allow checking structure and exchange data in standardized format
- ELEMENT type declaration
  - Defines structural rules
- ATTLIST attribute-list declaration
  - Defines an attribute

CSE@UTA Programming Languages Introduction to XML  
Fall2007 ©Weimin He 15

### An Example DTD

```
<!-- DTD document for auction.xml -->
<!ELEMENT auction(vendor+)>
<!ELEMENT vendor(company, email, phone, item*)>
<!ELEMENT company(#PCDATA)>
<!ELEMENT email(#PCDATA)>
<!ELEMENT phone(#PCDATA)>
<!ELEMENT item(name, description, seller, payment, price, city, state)>
<!-- ATTLIST item id #REQUIRED -->
<!ELEMENT name(#PCDATA)>
<!ELEMENT description(#PCDATA)>
<!ELEMENT seller(username,feedback,email?)>
<!ELEMENT payment(#PCDATA)>
<!ELEMENT price(#PCDATA)>
<!ELEMENT city(#PCDATA)>
<!ELEMENT state(#PCDATA)>
<!ELEMENT username(#PCDATA)>
<!ELEMENT feedback(#PCDATA)>
<!ELEMENT email(#PCDATA)>
```

CSE@UTA Programming Languages Introduction to XML  
Fall2007 ©Weimin He 16

### Document Object Model (DOM)

- Presents an XML document as an ordered, node-labeled tree, with the elements, attributes, and text defined as nodes.
- Defines a standard way for accessing and manipulating XML documents
- Validate and determine wellformedness of XML documents
- Memory intensive
- Preferred way to actually manipulate a document

CSE@UTA Programming Languages Introduction to XML  
Fall2007 ©Weimin He 17

### Building DOM Tree By Example

```
<!-- An exmple document for DOM -->
<auction>
  <item id = "item1">
    <name>Mountain Bicycle</name>
    <seller>
      <username>bicycle seller </username>
    </seller>
  </item>
</auction>
```

CSE@UTA Programming Languages Introduction to XML  
Fall2007 ©Weimin He 18

## CSE@UTA DOM Interface

**Java packages:**

- `org.w3c.dom` provides the interfaces
- `javax.xml.parsers` provides the parser

```
public interface Node {
    public String getNodeName ();
    public String getNodeValue ();
    public NodeList getChildNodes ();
    public NamedNodeMap getAttributes ();
}
public interface Element extends Node {
    public NodeList getElementsByTagName ( String name );
}
public interface Document extends Node {
    public Element getDocumentElement ();
}
public interface NodeList {
    public int getLength ();
    public Node item ( int index );
}
```

CSE3302 Programming Languages Fall2007 ©Weimin He Introduction to XML 19

## CSE@UTA Building A Document Using DOM

```
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setValidating( true );
DocumentBuilder builder = dbf.newDocumentBuilder();
Document d = builder.newDocument();

Element root = d.createElement( "auction" );

d.appendChild( root );
Comment c = d.createComment( "This is a comment" );
root.appendChild( c );

Element item = d.createElement( "item" );
Element name = d.createElement( "name" );
Element city = d.createElement( "city" );
name.appendChild( d.createTextNode( "Mountain Bicycle" );
city.appendChild( d.createTextNode( "Arlington" );
item.appendChild( name );
item.appendChild( city );
root.appendChild( item );
...
```

CSE3302 Programming Languages Fall2007 ©Weimin He Introduction to XML 20

## CSE@UTA Reading A Document Using DOM

```
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setValidating( true );
DocumentBuilder db = dbf.newDocumentBuilder();
db.setErrorHandler( new MyErrorHandler() );
Document doc = db.parse( new File( "message.xml" ) );
// assuming the document looks as follows:
/*
<message to = "john@cse.uta.edu" from = "tom@yahoo.com">
  A message for john from tom
</message>
*/
Element root = doc.getDocumentElement();
if ( !root.getTagName().equals( "message" ) )
{ // some error handling routine; return; }
String from = root.getAttribute( "from" );
String to = root.getAttribute( "to" );
String text = root.getFirstChild().getNodeValue();
// send message to corresponding user
processMail( to, from, text );
```

CSE3302 Programming Languages Fall2007 ©Weimin He Introduction to XML 21

## CSE@UTA Traversing A Document Using DOM

```
public class TraverseDOM{
    private Document document;
    public TraverseDOM(String file){
        try {
            DocumentBuilderFactory dbf =
                DocumentBuilderFactory.newInstance();
            dbf.setValidating( true );
            DocumentBuilder db = dbf.newDocumentBuilder();
            db.setErrorHandler( new MyErrorHandler() );
            document = db.parse( new File( "file.xml" ) );
        } catch ( ... ) { ... }
    }
    void processNode(Node currentNode) { ... }
    void processChildNodes(NodeList children){
        if ( children.getLength() != 0 ) {
            for ( int i = 0; i < children.getLength(); i++ )
                processNode( children.item(i) );
        }
    }
    public static void main(String args[]){
        TraverseDOM traverseDOM = new
            TraverseDOM(args[0]);
    }
}
void processNode(Node currentNode){
    switch( currentNode.getNodeType() ){
        case Node.DOCUMENT_NODE:
            Document doc = (Document) currentNode;
            System.out.println( "Document node: "+
                doc.getNodeName() + " root element: "+
                doc.getDocumentElement().getNodeName() );
            processChildNodes( doc.getChildNodes() );
            break;
        case Node.ELEMENT_NODE:
            System.out.println( "Element node: "+
                currentNode.getNodeName() );
            NamedNodeMap attrs = currentNode.getAttributes();
            for ( int i = 0; i < attrs.getLength(); i++ ){
                Attr attr = (Attr) attrs.item( i );
                System.out.println( "Attribute: "+
                    attr.getNodeName() + "; Value = "+
                    attr.getNodeValue() );
            }
            processChildNodes( currentNode.getChildNodes() );
            break;
        case Node.TEXT_NODE:
            TEXT text = (TEXT) currentNode;
            if ( !text.getNodeValue().trim().equals( "" ) )
                System.out.println( "Text: "+ text.getNodeValue() );
            break;
    }
}
```

CSE3302 Programming Languages Fall2007 ©Weimin He Introduction to XML 22

## CSE@UTA Simple API for XML (SAX)

- Event-based data processing
- Operates on a byte stream
- A lightweight approach to scanning XML documents
- Efficient and fast
- Can handle documents of any size
- Hard to write non-trivial applications

CSE3302 Programming Languages Fall2007 ©Weimin He Introduction to XML 23

## CSE@UTA Invoking SAX

```
import java.io.*;
import javax.xml.parsers.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;

public class InvokeSax {
    static class MyHandler extends DefaultHandler { ... }

    public static void parseXmlFile(String filename, DefaultHandler handler, boolean validating) {
        try {
            SAXParserFactory factory = SAXParserFactory.newInstance();
            factory.setValidating( validating );
            SAXParser parser = factory.newSAXParser();
            XMLReader reader = parser.getXMLReader();
            reader.setContentHandler( handler );
            reader.parse( new InputSource( new FileReader( filename ) ) );
        } catch ( ... ) { ... }
    }

    public static void main(String[] args) {
        DefaultHandler handler = new MyHandler();
        parseXmlFile( "file.xml", handler, false );
    }
}
```

CSE3302 Programming Languages Fall2007 ©Weimin He Introduction to XML 24

## SAX Parser Events

- Receive notification of the beginning of a document  
void startDocument ()
- Receive notification of the end of a document  
void endDocument ()
- Receive notification of the beginning of an element  
void startElement ( String namespace, String localName, String qName, Attributes atts )
- Receive notification of the end of an element  
void endElement ( String namespace, String localName, String qName )
- Receive notification of character data  
void characters ( char[] ch, int start, int length )

CSE3302 Programming Languages Introduction to XML Fall2007 ©Weimin He 25

## SAX Example

```

public class MyHandler extends DefaultHandler {
    private int indentation = 0;

    public void startElement(String namespaceUri, String localName,
        String qualifiedName, Attributes attributes) {
        indent(indentation);
        System.out.println("Start tag: " + qualifiedName);
        int numAttributes = attributes.getLength();
        if (numAttributes > 0) {
            System.out.print(" ");
            for(int i=0; i<numAttributes; i++) {
                if (i>0) System.out.print(", ");
                System.out.print(attributes.getQName(i) + "=" +
                    attributes.getValue(i));
            }
            System.out.println();
        }
        System.out.println();
        indentation = indentation + 2;
    }

    public void endElement(String namespaceUri, String localName,
        String qualifiedName) {
        indentation = indentation - 2;
        indent(indentation);
        System.out.println("End tag: " + qualifiedName);
    }

    public void characters(char[] chars,
        int startIndex,
        int endIndex) {
        String data = new String(chars, startIndex,
            endIndex);
        StringTokenizer tok = new StringTokenizer(data);
        if (tok.hasMoreTokens()) {
            indent(indentation);
            System.out.print(tok.nextToken());
            if (tok.hasMoreTokens()) {
                System.out.print(" ");
            } else {
                System.out.println();
            }
        }
    }

    private void indent(int indentation) {
        for(int i=0; i<indentation; i++) {
            System.out.print(" ");
        }
    }
}
    
```

CSE3302 Programming Languages Introduction to XML Fall2007 ©Weimin He 26

## Output for SAX Example

<pre> &lt;auction&gt; &lt;vendor&gt; &lt;company&gt;ebay&lt;/company&gt; &lt;email&gt;webmaster@ebay.com&lt;/email&gt; &lt;phone&gt;1-800-333-4444&lt;/phone&gt; &lt;item id = "item1"&gt; &lt;name&gt;Mountain Bicycle&lt;/name&gt; &lt;description&gt; Used for 2 months, but looks like new &lt;/description&gt; &lt;seller&gt; &lt;username&gt;bicycle seller&lt;/username&gt; &lt;/seller&gt; &lt;payment&gt; credit card, money order,check &lt;/payment&gt; &lt;price&gt;65.00&lt;/price&gt; &lt;/item&gt; &lt;/vendor&gt; &lt;/auction&gt;                 </pre>	<pre> Start tag: auction Start tag: vendor Start tag: company ebay End tag: company Start tag: email webmaster@ebay.com End tag: email Start tag: phone 1-800-333-4444 End tag: phone Start tag: item (id=item1) Mountain... End tag: name Start tag: description Used... End tag: description Start tag: seller Start tag: username bicycle... End tag: username End tag: seller Start tag: payment credit... End tag: payment Start tag: price 65.00 End tag: price End tag: item End tag: vendor End tag: auction                 </pre>
---	---

CSE3302 Programming Languages Introduction to XML Fall2007 ©Weimin He 27

## XPath

- A simple language for querying XML data
- Use path expressions to navigate in XML documents
- W3C standard
- Example:
  - //auction/item[name="Mountain Bicycle"][city="Dallas"]/description

CSE3302 Programming Languages Introduction to XML Fall2007 ©Weimin He 28

## XPath Axes

- Ancestor
- Ancestor-or-self
- Attribute
- Child
- Descendant
- Descendant-or-self
- Following
- Following-sibling
- Parent
- Preceding
- Self

CSE3302 Programming Languages Introduction to XML Fall2007 ©Weimin He 29

## XPath Node Tests

<ul style="list-style-type: none"> <li>• item</li> <li>• *</li> <li>• @location</li> <li>• @*</li> <li>• node()</li> <li>• text()</li> <li>• element()</li> <li>• element(item)</li> <li>• attribute()</li> <li>• attribute(location)</li> </ul>	<ul style="list-style-type: none"> <li>any element node whose name is item</li> <li>any element node regardless of its name</li> <li>any attribute whose name is location</li> <li>any attribute, regardless of its name</li> <li>any node</li> <li>any text node</li> <li>any element node</li> <li>any element node whose name is item</li> <li>any attribute node</li> <li>any attribute whose name is location</li> </ul>
--	---

CSE3302 Programming Languages Introduction to XML Fall2007 ©Weimin He 30

CSE@UTA **Abbreviated Syntax**

- Child:: can be omitted from a location step
  - `item/price` is short for `child::item/child::price`
- Attribute:: can be abbreviated to @
  - `item[@id="item1"]` is short for `child::item[attribute:id="item1"]`
- `.` is short for `self::node()`
  - `./item` is short for `self::node()/descendant-or-self::node()/child::item`
- `..` is short for `parent::node()`
  - `../location` is short for `parent::node()/child::location`
- `//` is short for `/descendant-or-self::node()/`

CSE3302 Programming Languages Introduction to XML  
Fall2007 ©Weimin He 31

CSE@UTA **Common XPath Steps**

XPath Step	Returned Context Nodes
<code>/item</code>	all the children of a context node with tagname <code>item</code>
<code>/*</code>	all the children of the context node
<code>//item</code>	the context node and all its descendants with tagname <code>item</code>
<code>//*</code>	the context node and all its descendants
<code>/@id</code>	the attribute value of the attribute name <code>id</code> of the context node
<code>//@id</code>	like <code>/@id</code> but for all descendants
<code>../</code>	parent of context node
<code>/text()</code>	the text of the context node

CSE3302 Programming Languages Introduction to XML  
Fall2007 ©Weimin He 32

CSE@UTA **XPath By Example**

- `/auction`
- `item/*`
- `*/*`
- `*[@id]`
- `@*`
- `item[name][3]`
- `auction/item[position() = 1]`
- `item[1]/seller/email[2]`

CSE3302 Programming Languages Introduction to XML  
Fall2007 ©Weimin He 33

CSE@UTA **XPath By Example (Cont'd)**

- `//item[10]`
- `//item[last()]`
- `//item[name]`
- `//item[seller/email]`
- `//item[seller/feedback="100% positive"]`
- `/auction/item[@id = "item2"]/city`
- `/auction/item[name/text()]`
- `//item[seller/feedback="100% positive"] [/name="Mountain Bicycle"] /price`
- `/auction/item[price][seller[username="bicycle seller"]][email]/state`

CSE3302 Programming Languages Introduction to XML  
Fall2007 ©Weimin He 34

CSE@UTA **Useful Links for XML**

- <http://www.w3.org/XML/>
- <http://www.w3schools.com/xpath/>
- <https://jaxp.dev.java.net/>

CSE3302 Programming Languages Introduction to XML  
Fall2007 ©Weimin He 35