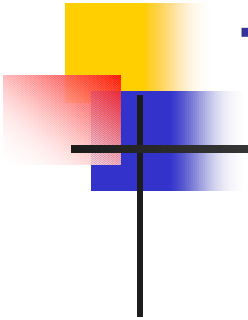
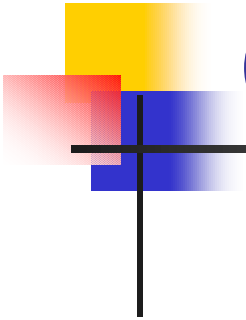
How to Use XML Parsing to Enhance Electronic Commnuication

Ted Leung
Chairman, ASF XML PMC
Principal, Sauria Associates, LLC
twl@sauria.com



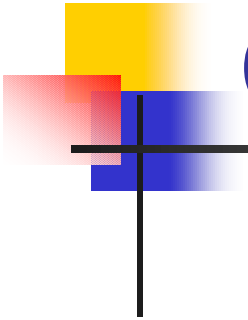
Thank You

- ASF
- xerces-j-dev
- xerces-j-user



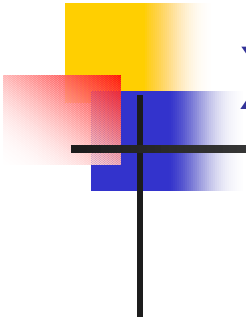
Outline

- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- JDOM/DOM4J
- Performance
- Xerces Architecture



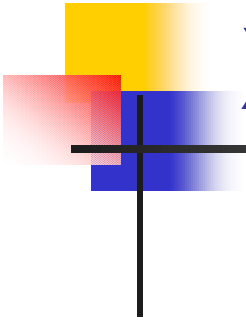
Overview

- Focus on server side XML
- Not document processing
- Benefits to servers / e-business
  - Model-View separation for *ML
  - Ubiquitous format for data exchange
    - Hope for network effects from data availability



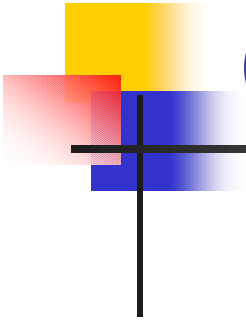
xml.apache.org

- Xerces XML parser
  - Java
  - C++
  - Perl
- Xalan XSLT processor
  - Java
  - C++
- FOP
- Cocoon
- Batik
- SOAP/Axis



Xerces-J

- Why Java?
  - Unicode
  - Code motion
  - Server side cross platform support
- C++ version has lagged Java version



Outline

- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- JDOM/DOM4J
- Performance
- Xerces Architecture

Basic XML Concepts

- Well formedness
- Validity
  - DTD's
  - Schemas
- Entities

Example: RSS

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE rss PUBLIC "-//Netscape Communications//DTD RSS .91//EN"
            "http://my.netscape.com/publish/formats/rss-0.91.dtd">

<rss version="0.91">

 <channel>
  <title>freshmeat.net</title>
  <link>http://freshmeat.net</link>
  <description>the one-stop-shop for all your Linux software
needs</description>
  <language>en</language>
  <rating>(PICS-1.1 "http://www.classify.org/safesurf/" 1 r (SS~~000
1))</rating>
  <copyright>Copyright 1999, Freshmeat.net</copyright>
  <pubDate>Thu, 23 Aug 1999 07:00:00 GMT</pubDate>
  <lastBuildDate>Thu, 23 Aug 1999 16:20:26 GMT</lastBuildDate>
  <docs>http://www.blahblah.org/fm.cdf</docs>
```

RSS 2

```
<image>
 <title>freshmeat.net</title>
 <url>http://freshmeat.net/images/fm.mini.jpg</url>
 <link>http://freshmeat.net</link>
 <width>88</width>
 <height>31</height>
 <description>This is the Freshmeat image stupid</description>
</image>

<item>
 <title>kdbg 1.0beta2</title>
 <link>http://www.freshmeat.net/news/1999/08/23/935449823.html</link>
 <description>This is the Freshmeat image stupid</description>
</item>

<item>
 <title>HTML-Tree 1.7</title>
 <link>http://www.freshmeat.net/news/1999/08/23/935449856.html</link>
 <description>This is the Freshmeat image stupid</description>
</item>
```

RSS 3

```
  <textinput>
   <title>quick finder</title>
   <description>Use the text input below to search freshmeat</description>
   <name>query</name>
   <link>http://core.freshmeat.net/search.php3</link>
  </textinput>

  <skipHours>
   <hour>2</hour>
  </skipHours>

  <skipDays>
   <day>1</day>
  </skipDays>

 </channel>
</rss>
```

RSS DTD

```
<!ELEMENT rss (channel)>
<!ATTLIST rss
          version     CDATA #REQUIRED> <!-- must be "0.91"> -->

<!ELEMENT channel (title | description | link | language | item+ |
                   rating?| image? | textinput? | copyright? | pubDate? |
                   lastBuildDate? | docs? | managingEditor? |
                   webMaster? | skipHours? | skipDays?)*>
<!ELEMENT title (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT link (#PCDATA)>
<!ELEMENT language (#PCDATA)>
<!ELEMENT rating (#PCDATA)>
<!ELEMENT copyright (#PCDATA)>
<!ELEMENT pubDate (#PCDATA)>
<!ELEMENT lastBuildDate (#PCDATA)>
<!ELEMENT docs (#PCDATA)>
<!ELEMENT managingEditor (#PCDATA)>
<!ELEMENT webMaster (#PCDATA)>
<!ELEMENT hour (#PCDATA)>
<!ELEMENT day (#PCDATA)>
<!ELEMENT skipHours (hour+)>
<!ELEMENT skipDays (day+)>
```

RSS DTD 2

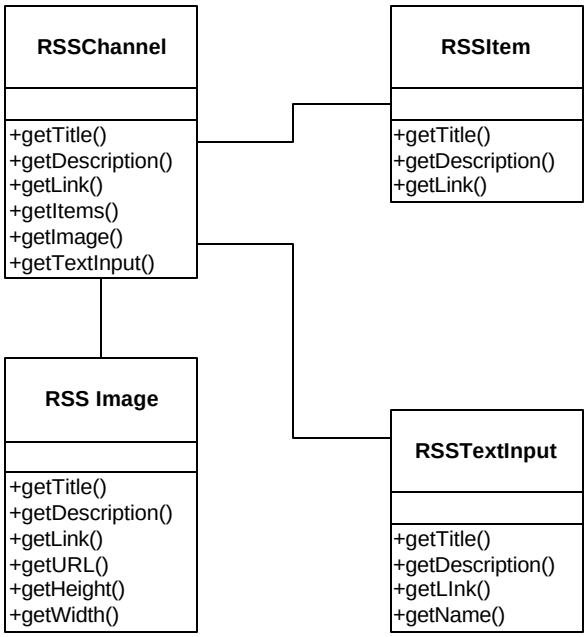
```
<!ELEMENT item (title | link | description)*>
<!ELEMENT textinput (title | description | name | link)*>
<!ELEMENT name (#PCDATA)>

<!ELEMENT image (title | url | link | width? | height? |
          description?)*>
<!ELEMENT url (#PCDATA)>
<!ELEMENT width (#PCDATA)>
<!ELEMENT height (#PCDATA)>
```

XML Application Tasks

- Convert XML into application object
- Convert application object to XML
- Read or write XML into a database
- XSLT conversion to XHTML or HTML

RSS Objects

Logical View

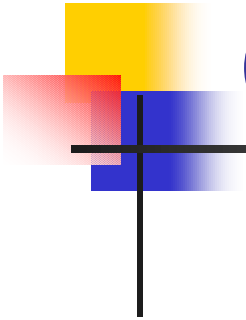


RSSItem 1

```
public class RSSItem implements java.io.Serializable {

  private String title;
  private String description;
  private String link;

  public String getTitle () {
    return title;
  }

  public void setTitle (String value) {
    String oldValue = title;
    title = value;
  }

  public String getLink () {
    return link;
  }

  public void setLink (String value) {
    String oldValue = link;
    link = value;
  }
```

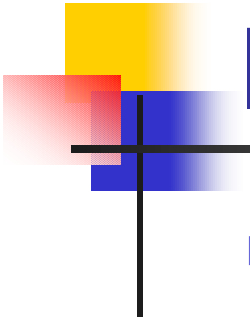
RSSItem 2

```
public String getDescription () {
    return description;
  }

  public void setDescription (String value) {
    String oldValue = description;
    description = value;
  }
}
```



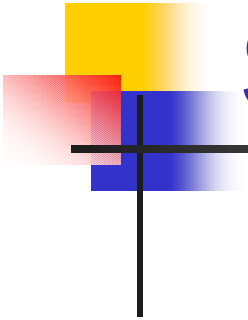
Outline

- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- JDOM/DOM4J
- Performance
- Xerces Architecture



Parser API's

- What is the job of a parser API?
  - Make the structure and contents of a document available
  - Report errors



SAX API

- Event callback style
- Representation of elements & atts
  - must build own stack
- SAX Processing Pipeline model
- Development model
  - Reasonably open
  - xml-dev
  - http://www.megginson.com/SAX/SAX2

ContentHandler

- The basic SAX2 handler for XML
- Callbacks for
  - Start of document
  - End of document
  - Start of element
  - End of element
  - Character data
  - Ignorable whitespace

Handler Strategies

- Single Monolithic
  - Requires the handler to know entirely about the grammar
- One per application object and multiplexer
  - Each application object has its own DocumentHandler
  - The Multiplexer handler is registered with the parser
  - Multiplexer is responsible for swapping SAX handlers as the context changes.

SAX 2 RSS Handler

```
public void startElement(String namespaceURI, String localName,
                         String qName, Attributes atts)
                         throws SAXException {
      currentText = new StringBuffer();
      textStack.push(currentText);
      if (localName.equals("rss")) {
          elementStack.push(localName);
      } else if (localName.equals("channel")) {
          elementStack.push(localName);
          currentChannel = new RSSChannel();
          currentItems = new Vector();
          currentChannel.setItems(currentItems);
          currentImage = new RSSImage();
          currentChannel.setImage(currentImage);
          currentTextInput = new RSSTextInput();
          currentChannel.setTextInput(currentTextInput);
          currentChannel.setSkipHours(new Vector());
          currentChannel.setSkipDays(new Vector());
      } else if (localName.equals("title")) {
      } else if (localName.equals("description")) {
      } else if (localName.equals("link")) {
      } else if (localName.equals("language")) {
```

SAX 2 RSS Handler 2

```
        } else if (localName.equals("item")) {
            elementStack.push(localName);
            currentItem = new RSSItem();
        } else if (localName.equals("rating")) {
        } else if (localName.equals("image")) {
            elementStack.push(localName);
        } else if (localName.equals("textinput")) {
            elementStack.push(localName);
        } else if (localName.equals("copyright")) {
        } else if (localName.equals("pubDate")) {
        } else if (localName.equals("lastBuildDate")) {
        } else if (localName.equals("docs")) {
        } else if (localName.equals("managingEditor")) {
        } else if (localName.equals("webMaster")) {
        } else if (localName.equals("hour")) {
        } else if (localName.equals("day")) {
        } else if (localName.equals("skipHours")) {
            elementStack.push(localName);
        } else if (localName.equals("skipDays")) {
            elementStack.push(localName);
        } else {}
    }
```

SAX 2 RSS Handler 3

```
public void endElement(String namespaceURI, String localName,
                       String qName) throws SAXException {
       try {
           String stackValue = (String) elementStack.peek();
           String text = ((StringBuffer)textStack.pop()).toString();
           if (localName.equals("rss")) {
               elementStack.pop();
           } else if (localName.equals("channel")) {
               elementStack.pop();
           } else if (localName.equals("title")) {
               if (stackValue.equals("channel")) {
                   currentChannel.setTitle(text);
               } else if (stackValue.equals("image")) {
                   currentImage.setTitle(text);
               } else if (stackValue.equals("item")) {
                   currentItem.setTitle(text);
               } else if (stackValue.equals("textinput")) {
                   currentTextInput.setTitle(text);
               } else {}
```

SAX 2 RSS Handler 4

```
} else if (localName.equals("description")) {
    if (stackValue.equals("channel")) {
        currentChannel.setDescription(text);
    } else if (stackValue.equals("image")) {
        currentImage.setDescription(text);
    } else if (stackValue.equals("item")) {
        currentItem.setDescription(text);
    } else if (stackValue.equals("textinput")) {
        currentTextInput.setDescription(text);
    } else {}
} else if (localName.equals("link")) {
    if (stackValue.equals("channel")) {
        currentChannel.setLink(text);
    } else if (stackValue.equals("image")) {
        currentImage.setLink(text);
    } else if (stackValue.equals("item")) {
        currentItem.setLink(text);
    } else if (stackValue.equals("textinput")) {
        currentTextInput.setLink(text);
    } else {}
```

SAX 2 RSS Handler 5

```
} else if (localName.equals("language")) {
    currentChannel.setLanguage(text);
} else if (localName.equals("item")) {
    currentItems.add(currentItem);
    elementStack.pop();
} else if (localName.equals("rating")) {
    currentChannel.setRating(text);
} else if (localName.equals("image")) {
    currentChannel.setImage(currentImage);
    elementStack.pop());
} else if (localName.equals("height")) {
    currentImage.setHeight(text);
} else if (localName.equals("width")) {
    currentImage.setWidth(text);
} else if (localName.equals("url")) {
    currentImage.setURL(text);
} else if (localName.equals("textinput")) {
    currentChannel.setTextInput(currentTextInput);
    elementStack.pop());
} else if (localName.equals("name")) {
    currentTextInput.setName(text);
```

SAX 2 RSS Handler 6

```
        } else if (localName.equals("copyright")) {
            currentChannel.setCopyright(text);
        } else if (localName.equals("pubDate")) {
            currentChannel.setPubDate(text);
        } else if (localName.equals("lastBuildDate")) {
            currentChannel.setLastBuildDate(text);
        } else if (localName.equals("docs")) {
            currentChannel.setDocs(text);
        } else if (localName.equals("managingEditor")) {
            currentChannel.setManagingEditor(text);
        } else if (localName.equals("webMaster")) {
            currentChannel.setWebMaster(text);
        } else if (localName.equals("hour")) {
            currentHour = text;
        } else if (localName.equals("day")) {
            currentDay = text;
        } else if (localName.equals("skipHours")) {
            currentChannel.getSkipHours().add(currentHour);
            elementStack.pop();
        } else if (localName.equals("skipDays")) {
            currentChannel.getSkipDays().add(currentDay);
            elementStack.pop();
        } else {}
    } catch (Exception e) {
        e.printStackTrace();
```

SAX 2 RSS Handler 7

```
public void characters(char[] ch,int start,int length) throws SAXException {
   currentText.append(ch, start, length);
}
```

SAX 2 Basic Event Handling

- Basic Event Handling
  - attribute handling in startElement
  - characters() for content
  - character data handling in endElement
    - Multiple characters calls

SAX 2 RSS ErrorHandler

```
public class SAX2RSSErrorHandler implements ErrorHandler {
  public void warning(SAXParseException ex) throws SAXException
{
   System.err.println("[Warning] ”+getLocationString(ex)+":”+
     ex.getMessage());
  }

  public void error(SAXParseException ex) throws SAXException {
   System.err.println("[Error] "+getLocationString(ex)+":”+
     ex.getMessage());
  }

  public void fatalError(SAXParseException ex) throws
SAXException {
   System.err.println("[Fatal Error]"+getLocationString(ex)+":”
     +ex.getMessage());
   throw ex;
  }
```

SAX 2 RSS ErrorHandler 2

```
private String getLocationString(SAXParseException ex) {
  StringBuffer str = new StringBuffer();
  String systemId = ex.getSystemId();
  if (systemId != null) {
    int index = systemId.lastIndexOf('/');
    if (index != -1)
      systemId = systemId.substring(index + 1);
      str.append(systemId);
    }
    str.append(':');
    str.append(ex.getLineNumber());
    str.append(':');
    str.append(ex.getColumnNumber());
    return str.toString();
  }
}
```

SAX 2 Error Handling

- Error Handling
  - warning
  - error – parser may recover
  - fatalError – XML 1.0 spec errors
- Locators

SAX 2 RSS Driver

```
public static void main(String args[]) {
  XMLReader r = new SAXParser();
  ContentHandler h = new SAX2RSSHandler();
  r.setContentHandler(h);
  ErrorHandler eh = new SAX2RSSErrorHandler();
  r.setErrorHandler(eh);
  try {
    r.parse("file:///Work/fm0.91_full.rdf");
  } catch (SAXException se) {
    System.out.println("SAX Error "+se.getMessage());
    se.printStackTrace();
  } catch (IOException e) {
    System.out.println("I/O Error "+e.getMessage());
    e.printStackTrace();
  } catch (Exception e) {
    System.out.println("Error "+e.getMessage());
    e.printStackTrace();
  }
  System.out.println(((SAX2RSSHandler) h).getChannel().toString());
}
```

Entity Handling

- Entity / Catalog handling
  - Catalog handling is mostly for document processing (SGML) applications
  - Pluggable entity handling is important for server applications.
    - Provides access point to DBMS, LDAP, etc.
- predefined character entities
  - amp, lt, gt, apos, quot

Entity Handler

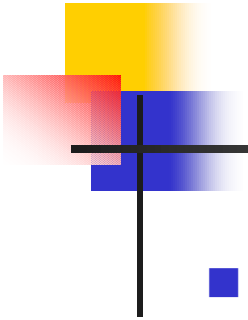
- `<!DOCTYPE rss PUBLIC "-//Netscape Communications//DTD RSS91//EN" "http://my.netscape.com/publish/formats/rss-0.91.dtd">`

- This will do a network fetch!

```
public class SAX1RSSEntityHandler implements EntityResolver {
  public SAX1RSSEntityHandler() {
  }
  public InputSource resolveEntity(String publicId,String systemId)
   throws SAXException, IOException {
    if (publicId.equals("-//Netscape Communications//DTD RSS
   91//EN")) {
       FileReader r = new FileReader("/usr/share/xml/rss91.dtd");
       return new InputSource(r);
    }
    return null;
  }
}
```

EntityDriver

```
public static void main(String args[]) {
  XMLReader r = new SAXParser();
  ContentHandler h = new SAX2RSSHandler();
  r.setContentHandler(h);
  ErrorHandler eh = new SAX2RSSErrorHandler();
  r.setErrorHandler(eh);
  EntityResolver er = new SAX2RSSEntityResolver();
  r.setEntityResolver(er);
  try {
    r.parse("file:///Work/fm0.91_full.rdf");
  } catch (SAXException se) {
    System.out.println("SAX Error "+se.getMessage());
    se.printStackTrace();
  } catch (IOException e) {
    System.out.println("I/O Error "+e.getMessage());
    e.printStackTrace();
  } catch (Exception e) {
    System.out.println("Error "+e.getMessage());
    e.printStackTrace();
  }
  System.out.println(((SAX2RSSHandler) h).getChannel().toString());
}
```



DefaultHandler

- The kitchen sink
- Lets you do ContentHandler, ErrorHandler, EntityResolver and DTD Handler
- A good way to go if you are only overriding a small number of methods

Locators

```
public class LocatorSample extends DefaultHandler {
  Locator fLocator = null;

  public void setDocumentLocator(Locator l) {
      fLocator = l;
  }

  public void startElement(String namespaceURI,
         String localName, String qName, Attributes atts) throws
SAXException {
    System.out.println(localName+" @ "+fLocator.getLineNumber()+",
"+fLocator.getColumnNumber());
    }

  public static void main(String args[]) {
    XMLReader r = new SAXParser();
    ContentHandler h = new LocatorSample();
    h.setDocumentLocator(((SAXParser) r).getLocator());
    r.setContentHandler(h);

    try {
      r.parse("file:///Work/fm0.91_full.rdf");
    } catch (Exception e) {}
  }
}
```

SAX 2 Extension Handlers

- LexicalHandler
  - Entity boundaries
  - DTD bounaries
  - CDATA sections
  - Comments
- DeclHandler
  - ElementDecls and AttributeDecls
  - Internal and External Entity Decls
  - NO parameter entities
- Compatibility Adapters

SAX 2 Configurable

- Uses strings (URI's) as lookup keys
- Features - booleans
  - validation
  - namespaces
- Properties - Objects
  - Extensibility by returning an object that implements additional function

Configurable Example

```
 public static void main(String[] argv) throws Exception {
   // construct parser; set features
   XMLReader parser = new SAXParser();
   try {
     parser.setFeature("http://xml.org/sax/features/namespaces",
true);
     parser.setFeature("http://xml.org/sax/features/validation",
true);
    parser.setProperty(“http://xml.org/sax/properties/declaration-
handler”, dh)
   } catch (SAXException e) {
     e.printStackTrace(System.err);
   }
}
```

Xerces and Configurable

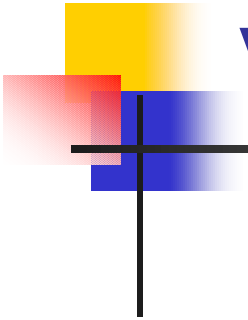
- Standard way to set all parser configuration settings.
- Applies to both SAX and DOM Parsers

Xerces Features

- Inclusion of general entities
- Inclusion of parameter entities
- Dynamic validation
- Extra warnings
- Allow use of java encoding names
- Lazy evaluating DOM
- DOM EntityReference creation
- Inclusion of igorable whitespace in DOM
- Schema validation [also full checking]
- Control of Non-Validating behavior
  - Defaulting attributes & attribute type-checking

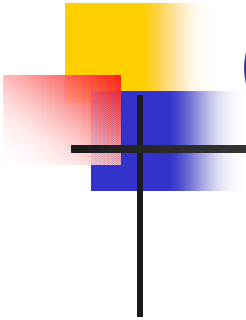
Xerces Properties

- Name of DOM implementation class
- DOM node currently being parsed
- Values for
  - noNamespaceSchemaLocation
  - SchemaLocation



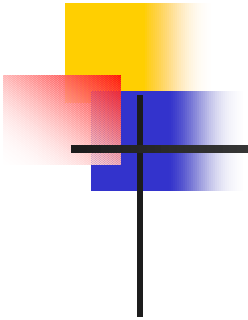
Validation

- When to use validation
  - System boundaries
  - Xerces 2 pluggable futures
- non-validating != wf checking
  - Not required to read external entities
- validation dial settings



Outline

- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- JDOM/DOM4J
- Performance
- Xerces Architecture



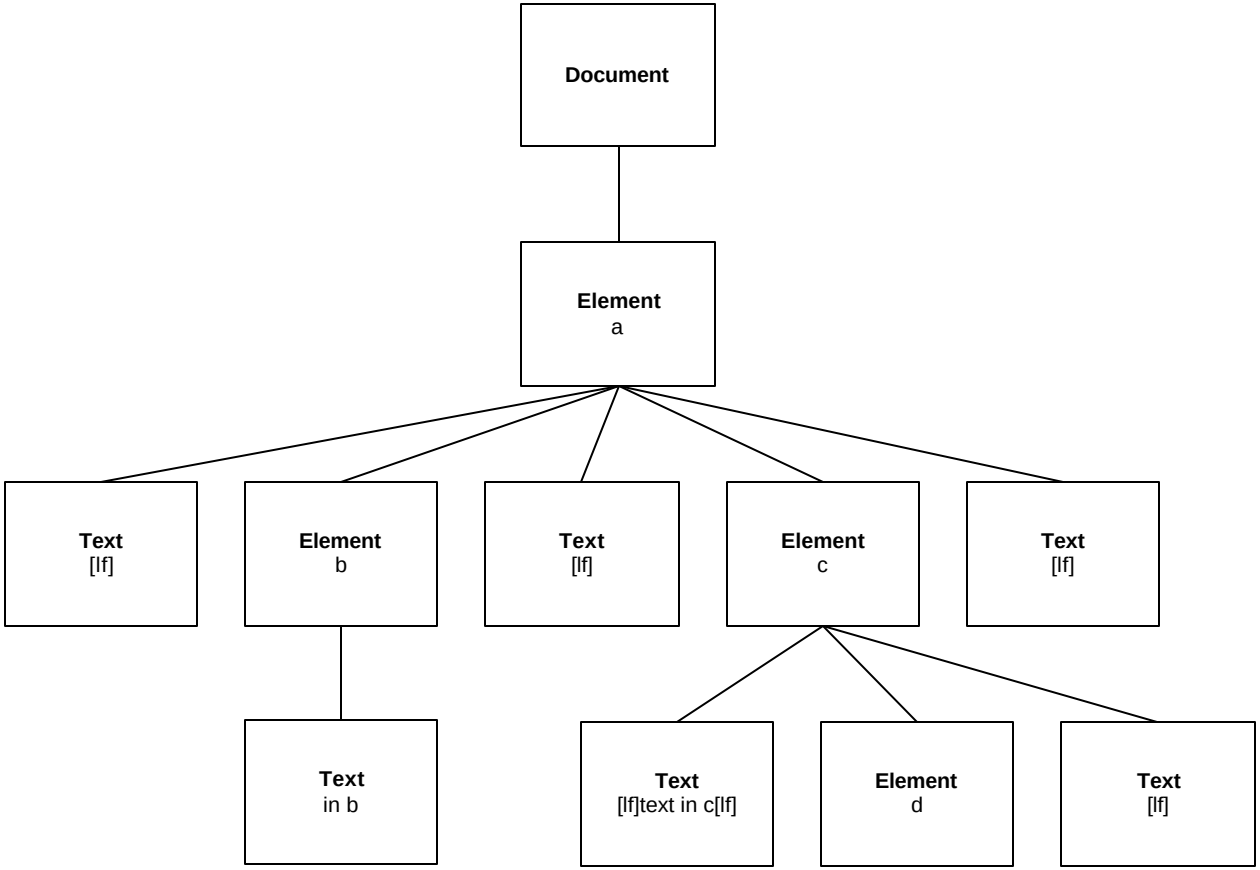
DOM API

- Object representation of elements & atts
- Development Model
  - W3C Recommendations and process
- http:///www.w3.org/DOM

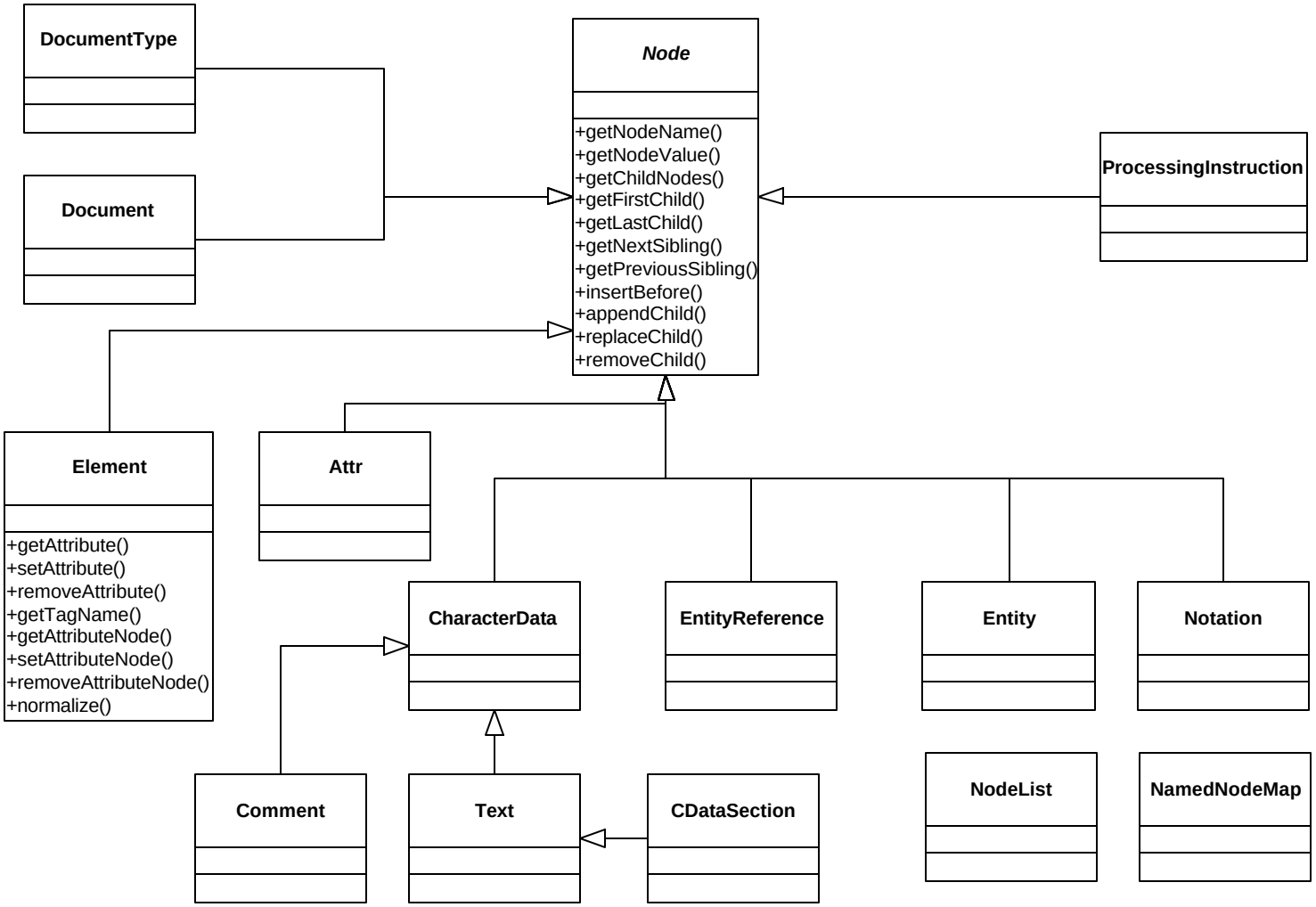
DOM Instance (XML)

```
<?xml version="1.0" encoding="US-ASCII"?>
<a>
 <b>in b</b>
 <c>
  text in c
  <d/>
 </c>
</a>
```

DOM Tree


Document
Element
a
Text
[lf]
Element
b
Text
[lf]
Element
c
Text
[lf]
Text
in b
Text
[lf]text in c[lf]
Element
d
Text
[lf]

DOM Architecture



DOM RSS

```
public static void main(String args[]) {
  DOMParser p = new DOMParser();

  try {
    p.parse("file:///Work/fm0.91_full.rdf");
    System.out.println(dom2Channel(p.getDocument()).toString());
  } catch (SAXException se) {
    System.out.println("Error during parsing "+se.getMessage());
    se.printStackTrace();
  } catch (IOException e) {
    System.out.println("I/O Error during parsing “+e.getMessage());
    e.printStackTrace();
  }
}
```

DOM2Channel

```
public static RSSChannel dom2Channel(Document d) {
   RSSChannel c = new RSSChannel();
   Element channel = null;
   NodeList nl = d.getElementsByTagName("channel");
   channel = (Element) nl.item(0);
   for (Node n = channel.getFirstChild(); n != null;
        n = n.getNextSibling()) {
     if (n.getNodeType() != Node.ELEMENT_NODE)
       continue;
     Element e = (Element) n;
     e.normalize();
     String text = e.getFirstChild().getNodeValue();
     if (e.getTagName().equals("title")) {
       c.setTitle(text);
     } else if (e.getTagName().equals("description")) {
       c.setDescription(text);
     } else if (e.getTagName().equals("link")) {
       c.setLink(text);
     } else if (e.getTagName().equals("language")) {

...
```

DOM2Channel 2

…

```
    }
    nl = channel.getElementsByTagName("item");
    c.setItems(dom2Items(nl));
    nl = channel.getElementsByTagName("image");
    c.setImage(dom2Image(nl.item(0)));
    nl = channel.getElementsByTagName("textinput");
    c.setTextInput(dom2TextInput(nl.item(0)));
    nl = channel.getElementsByTagName("skipHours");
    c.setSkipHours(dom2SkipHours(nl.item(0)));
    nl = channel.getElementsByTagName("skipDays");
    c.setSkipDays(dom2SkipDays(nl.item(0)));
    return c;
  }
```

DOM Node Traversal Methods

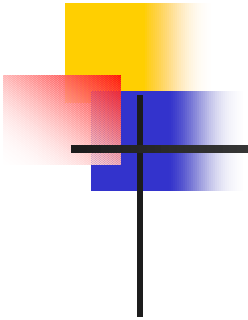
- Two paradigms
- Directly from the Node
  - `Node#getFirstChild`
  - `Node#getNextSibling`
- From the Nodelist for a Node
  - `Node#getChildren`
  - `NodeList#item(int i)`

DOM2Items

```
public static Vector dom2Items(NodeList nl) {
  Vector result = new Vector();    Node item = null;
  for (int x = 0; x < nl.getLength(); x++) {
    item = nl.item(x);
    if (item.getNodeType() != Node.ELEMENT_NODE) continue;
    RSSItem i = new RSSItem();
    for (Node n = item.getFirstChild(); n != null; n =
n.getNextSibling()) {
      if (n.getNodeType() != Node.ELEMENT_NODE) continue;
      Element e = (Element) n;
      e.normalize();
      String text = e.getFirstChild().getNodeValue();
      if (e.getTagName().equals("title")) {
        i.setTitle(text);
      } else if (e.getTagName().equals("description")) {
        i.setDescription(text);
      } else if (e.getTagName().equals("link")) {
        i.setLink(text);
      } }
    result.add(i);
  }
  return result;
```

DOM Level 1

- Attr
  - API for result as objects
  - API for result as strings
- No DTD support
- Need Import - copying between DOMs is disallowed



DOM Level 2

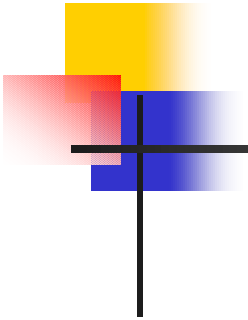
- W3C Recommendation
- Core / Namespaces
  - Import
- Traversal
  - Views on DOM Trees
- Events
- Range
- Stylesheets

DOM L2 Traversal 1

```
 public void parse() {
    DOMParser p = new DOMParser();
    try {
      p.parse("file:///Work//fm0.91_full.rdf");
      Document d = p.getDocument();

      TreeWalker treeWalker = ((DocumentTraversal)d).createTreeWalker(
                    d,
                    NodeFilter.SHOW_ALL,
                    new DOMFilter(),
                    true);
      for (Node n = treeWalker.getRoot(); n != null; n=treeWalker.nextNode(
{
          System.out.println(n.getNodeName());
      }
    } catch (IOException e) { e.printStackTrace(); }
  }


  public static void main(String args[]) {
      DOMRSSTraversal t = new DOMRSSTraversal();
      t.parse();
  }
```

DOM L2 Traversal 2

```
public class DOMFilter implements NodeFilter {
     public short acceptNode(Node n) {
         if (n.getNodeName().length() > 3)
             return FILTER_ACCEPT;
         else
             return FILTER_REJECT;
     }
 }
```



DOM L3

- W3C Working Draft
- Improve round trip fidelity
- Abstract Schemas / Grammar Access
- Load & Save
- XPath access

Xerces DOM extensions

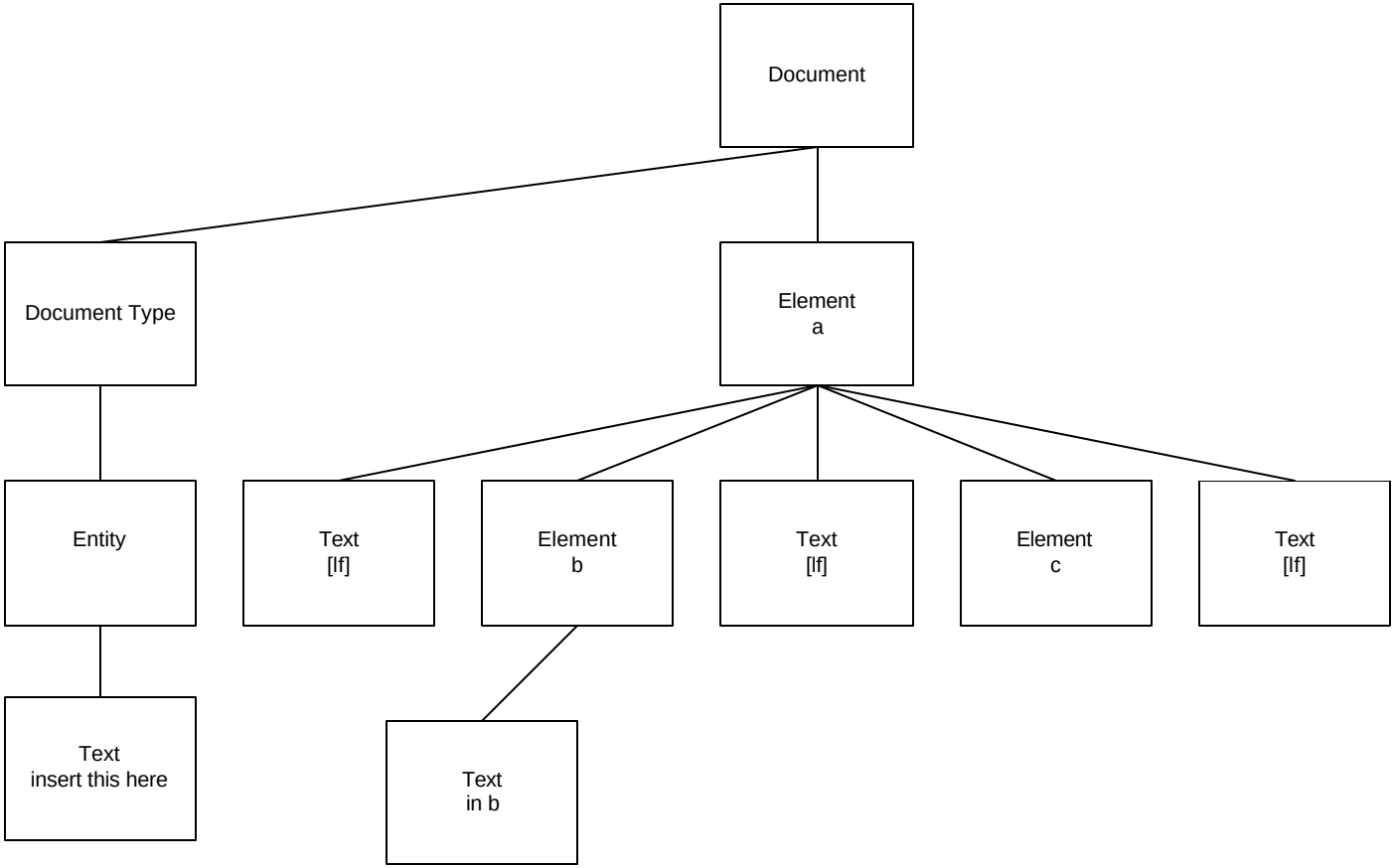
- Feature controls insertion of ignorable whitespace
- Feature controls creation of EntityReference Nodes
- User data object provided on implementation class org.apache.xml.dom.NodeImpl

DOM Entity Reference Nodes 1

```
<?xml version="1.0" encoding="US-ASCII"?>
<!DOCTYPE a [
<!ENTITY boilerplate "insert this here">
]>
<a>
 <b>in b</b>
 <c>
  text in c but &boilerplate;
  <d/>
 </c>
</a>
```

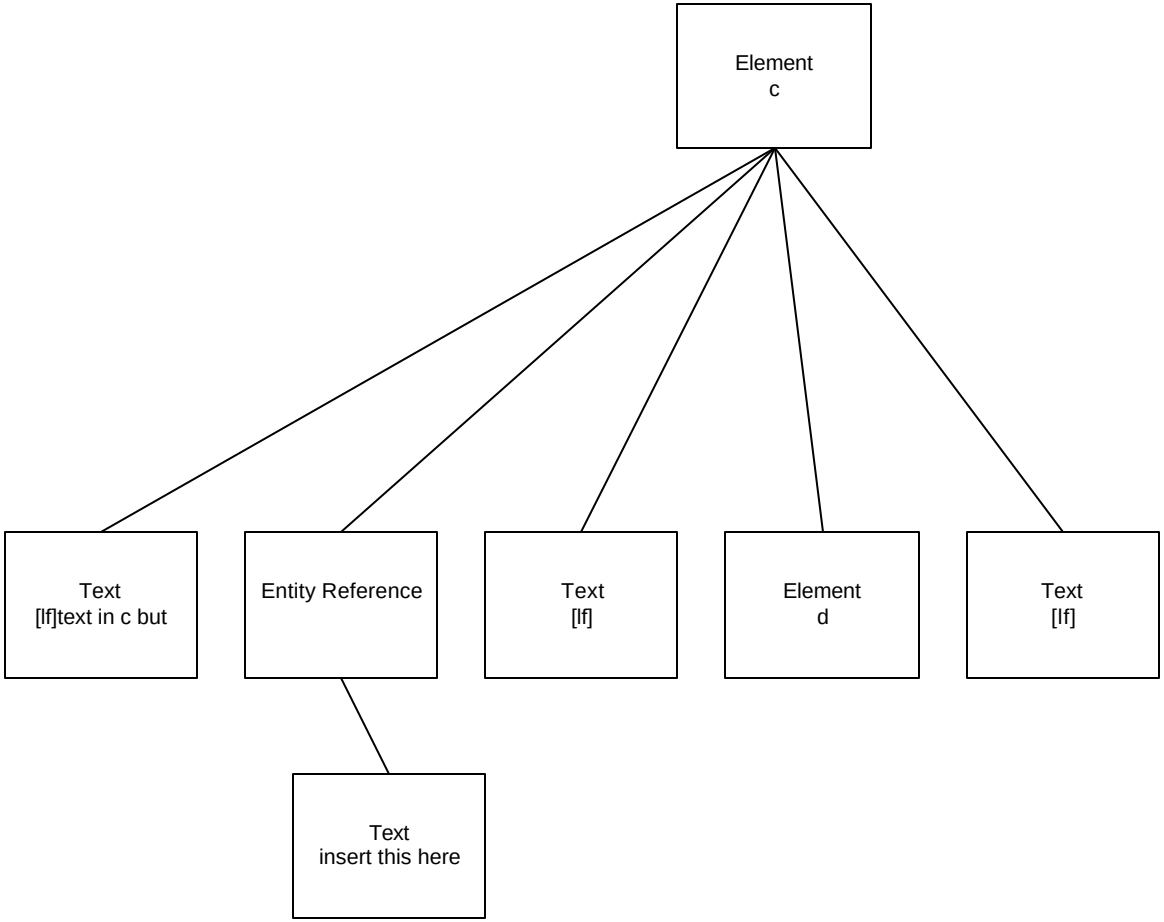
DOM Entity Reference Nodes 2

- Document Type Node



DOM Entity Reference Nodes 3

- Entity Reference Node

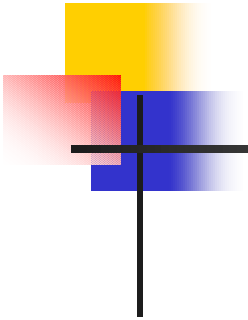


Xerces Lazy DOM

- Delays object creation
  - a node is accessed - its object is created
  - all siblings are also created.
- Reduces time to complete parsing
- Reduces memory usage
- Good if you only access part of the DOM

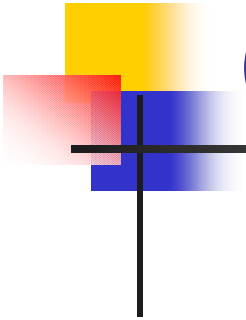
Tricky DOM stuff

- Entity and EntityReference nodes – depends on parser settings
- Ignorable whitespace
- Subclassing the DOM
  - Use the Document factory class



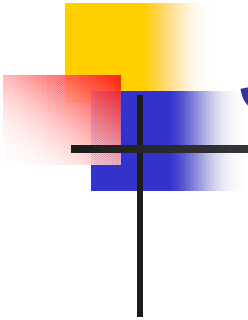
DOM vs SAX

- DOM
  - creates a document object tree
  - tree must the be reparsed & converted to BO's
  - Could subclass BO's from DOM classes
- SAX
  - event handler based API
  - stack management
  - no intermediate data structures generated



Outline

- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- JDOM/DOM4J
- Performance
- Xerces Architecture



JAXP

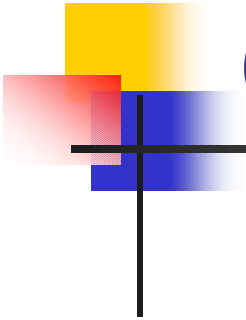
- Java API for XML Parsing
- Sun JSR-061
- Version 1.1
- Abstractions for SAX, DOM, XSLT
- Support in Xerces

SAX in JAXP

```
public static void main(String args[]) {
  XMLReader r = null;
  SAXParserFactory spf = SAXParserFactory.newInstance();
  try {
    r = spf.newSAXParser().getXMLReader();
  } catch (Exception e) {
    System.exit(1);
  }
  ContentHandler h = new SAX2RSSHandler();
  r.setContentHandler(h)
  ErrorHandler eh = new SAX2RSSErrorHandler();
  r.setErrorHandler(eh);
  try {
    r.parse("file:///Work/fm0.91_full.rdf");
  } catch (Exception e) {
    System.out.println("Error during parsing "+e.getMessage());
    e.printStackTrace();
  }
  System.out.println(((SAX2RSSHandler)
h).getChannel().toString());
}
```

DOM in JAXP

```
public static void main(String args[]) {
  DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
  DocumentBuilder db = null;
  try {
    db = dbf.newDocumentBuilder();
  } catch (ParserConfigurationException pce) {
    System.exit(1);
  }
  try {
    Document doc = db.parse("file:///Work/fm0.91_full.rdf");
  } catch (SAXException se) {
  } catch (IOException ioe) {
  }
  System.out.println(dom2Channel(doc).toString());
}
```



Outline

- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- JDOM/DOM4J
- Performance
- Xerces Architecture

Namespaces

- Purpose
  - Syntactic discrimination only
  - They don’t point to anything
  - They don’t imply schema composition/combination
- Universal Names
  - URI + Local Name
    {http://www.mydomain.com}Element
- Declarations
  - xmlns “attribute”
- Prefixes
- Scoping
- Validation and DTDs

Namespaces

- Purpose
  - Syntactic discrimination only
  - They don’t point to anything
  - They don’t imply schema composition/combination
- Universal Names
  - URI + Local Name
    {http://www.mydomain.com}Element
- Declarations
  - xmlns “attribute”
- Prefixes
- Scoping
- Validation and DTDs

Namespace Example

```
<xsl:stylesheet version="1.0“
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform“
                xmlns:html="http://www.w3.org/TR/xhtml1/strict">
 <xsl:template match="/">
   <html:html>
     <html:head>
       <html:title>Title</html:title>
    </html:head>
    <html:body>
      <xsl:apply-templates/>
    </html:body>
 </xsl:template>

 ...

</xsl:stylesheet>
```

Default Namespace Example

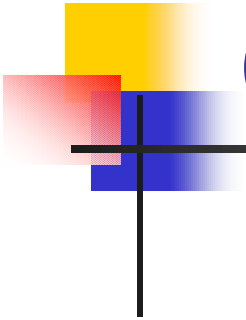
```
<xsl:stylesheet version="1.0“
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform“
                xmlns="http://www.w3.org/TR/xhtml1/strict">
 <xsl:template match="/">
   <html>
     <head>
       <title>Title</title>
    </head>
    <body>
      <xsl:apply-templates/>
    </body>
 </xsl:template>

 ...

</xsl:stylesheet>
```

SAX2 ContentHandler

```
public class NamespaceDriver extends DefaultHandler implements ContentHandler {

  public void startPrefixMapping(String prefix,String uri) throws SAXException {
    System.out.println("startMapping "+prefix+", uri = "+uri);
  }

  public void endPrefixMapping(String prefix) throws SAXException {
    System.out.println("endMapping "+prefix);
  }

  public void startElement(String namespaceURI,String localName,String
                           qName,Attributes atts) throws SAXException {
    System.out.println("Start: nsURI="+namespaceURI+" localName= "+localName+
                       " qName = "+qName);
    printAttributes(atts);
  }

  public void endElement(String namespaceURI,String localName,
                         String qName) throws SAXException {
    System.out.println("End: nsURI="+namespaceURI+" localName= "+localName+
                       " qName = "+qName);
  }
}
```

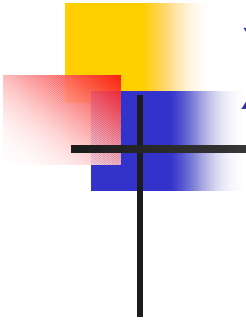
SAX2 ContentHandler 2

```
public void printAttributes(Attributes atts) {
    for (int i = 0; i < atts.getLength(); i++) {
      System.out.println("\tAttribute "+i+" URI="+atts.getURI(i)+
                         " LocalName="+atts.getLocalName(i)+
                         " Type="+atts.getType(i)+" Value="+atts.getValue(i));
    }
  }
```



Outline

- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- Performance
- JDOM/DOM4J
- Xerces Architecture



XML Schema

- Richer grammar specification
- W3C Recommendation
- Structures
  - XML Instance Syntax
  - Namespaces
  - Weird content models
- Datatypes
  - Lots
- Support in Xerces 1.4

RSS Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xs:schema
          PUBLIC "-//W3C//DTD XMLSCHEMA 200102//EN“
          "XMLSchema.dtd" >

 <xs:schema
  targetNamespace=“http://my.netscape.com/publish/formats/rss-0.91”
  xmlns="http://my.netscape.com/publish/formats/rss-0.91">
  <xs:element name="rss">
   <xs:complexType>
    <xs:sequence>
     <xs:element ref="channel"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" fixed="0.91"/>
   </xs:complexType>
  </xs:element>
```

RSS Schema 2

```
<xs:element name="channel">
   <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded“>
      <xs:element ref="title"/>
      <xs:element ref="description"/>
      <xs:element ref="link"/>
      <xs:element ref="language"/>
      <xs:element ref="item" minOccurs="1" maxOccurs="unbounded"/>
      <xs:element ref="rating" minOccurs="0"/>
      <xs:element ref="image" minOccurs="0"/>
      <xs:element ref="textinput" minOccurs="0"/>
      <xs:element ref="copyright" minOccurs="0"/>
      <xs:element ref="pubDate" minOccurs="0"/>
      <xs:element ref="lastBuildDate" minOccurs="0"/>
      <xs:element ref="docs" minOccurs="0"/>
      <xs:element ref="managingEditor" minOccurs="0"/>
      <xs:element ref="webMaster" minOccurs="0"/>
      <xs:element ref="skipHours" minOccurs="0"/>
      <xs:element ref="skipDays" minOccurs="0"/>
     </xs:choice>
   </xs:complexType>
  </xs:element>
```

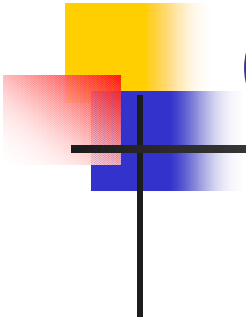
RSS Schema 3

…

```
  <xs:element name="url"  type="xs:anyURI"/>
  <xs:element name="name" type="xs:string"/>
  <xs:element name="rating" type="xs:string"/>
  <xs:element name="language" type="xs:string"/>
  <xs:element name="width" type="xs:integer"/>
  <xs:element name="height" type="xs:integer"/>
  <xs:element name="copyright" type="xs:string"/>
  <xs:element name="pubDate" type="xs:string"/>
  <xs:element name="lastBuildDate" type="xs:string"/>
  <xs:element name="docs" type="xs:string"/>
  <xs:element name="managingEditor" type="xs:string"/>
  <xs:element name="webMaster" type="xs:string"/>
  <xs:element name="hour" type="xs:integer"/>
  <xs:element name="day" type="xs:integer"/>
</xs:schema>
```

RelaxNG

- Alternative to XML Schema
- Satisifies the big 3 goals
  - XML Instance Syntax
  - Datatyping
  - Namespace Support
- More simple and orthogonal than XML Schema
- OASIS TC
- Jing
  - http://www.thaiopensource.com/jing



Outline

- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- JDOM/DOM4J
- Performance
- Xerces Architecture

Grammar Access

- No DOM standard
- SAX 2
- Common Applications
  - Editors
  - Stub generators
- Also experimental API in Xerces

DTD Access Example

```
public class DTDDumper implements DeclHandler {
  public void elementDecl (String name, String model) throws SAXException {
    System.out.println("Element "+name+" has this content model: "+model);
  }

  public void attributeDecl (String eName, String aName, String type,
                             String valueDefault, String value) throws
SAXException {
    System.out.print("Attribute "+aName+" on Element "+eName+" has type
"+type);
    if (valueDefault != null)
      System.out.print(", it has a default type of "+valueDefault);
    if (value != null)
      System.out.print(", and a default value of "+value);
    System.out.println();
  }
```

DTD Access Example 2

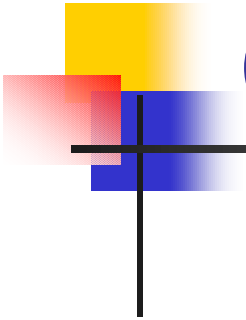
```
 public void internalEntityDecl (String name, String value) throws
SAXException {
    String peText = name.startsWith("%") ? "Parameter " : "";
    System.out.println("Internal "+peText+"Entity "+name+" has value:
"+value);
  }

  public void externalEntityDecl (String name, String publicId, String
systemId) throws SAXException {
    String peText = name.startsWith("%") ? "Parameter " : "";
    System.out.print("External "+peText+"Entity "+name);
    System.out.print("is available at "+systemId);
    if (publicId != null)
      System.out.print(" which is known as "+publicId);
    System.out.println();
  }
```

DTD Access Driver

```
public static void main(String args[]) {
  try {
    XMLReader r = new SAXParser();
    r.setProperty("http://xml.org/sax/properties/declaration-handler",
                  new DTDDumper());
      r.parse(args[0]);
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

Grammar Access in Schema

- In XML Schema or RelaxNG
- Easy, just parse an XML document

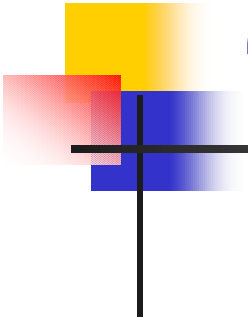


Outline

- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- JDOM/DOM4J
- Performance
- Xerces Architecture

Round Tripping

- Use SAX2 - DOM can’t do it until L3
- XML Decl
- Comments (SAX2)
- PE Decls

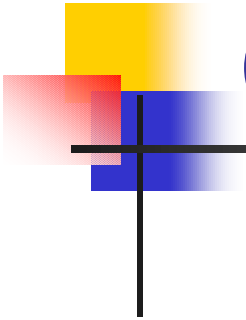


“Serializing”

- Several choices
  - XMLSerializer
  - TextSerializer
  - HTMLSerializer
    - XHTMLSerializer
- Work as either:
  - SAX DocumentHandlers
  - DOM serializers

Serializer Example

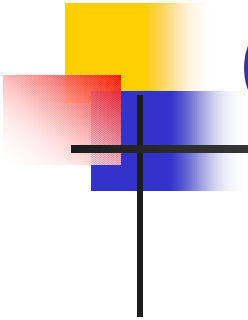
```
DOMParser p = new DOMParser();
p.parse("file:///twl/Work/ApacheCon/fm0.91_full.rdf");
Document d = p.getDocument();
// XML
OutputFormat format = new OutputFormat("xml", "UTF-8", true);
XMLSerializer serializer = new XMLSerializer(System.out, format);
serializer.serialize(d);
// XHTML
format = new OutputFormat("xhtml", "UTF-8", true);
serializer = new XHTMLSerializer(System.out, format);
serializer.serialize(d);
```



Outline

- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- JDOM/DOM4J
- Performance
- Xerces Architecture

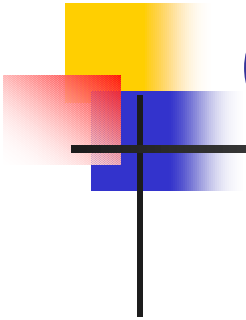
Grammar Design



- Attributes vs Elements
  - performance tradeoff vs programming tradeoff
  - At most 1 attribute, many subelements
- ID & IDREF
- NMTOKEN vs CDATA
- CDATASECTIONs
  - `<![CDATA[ My data & stuff  ]]>`
- Ignorable Whitespace

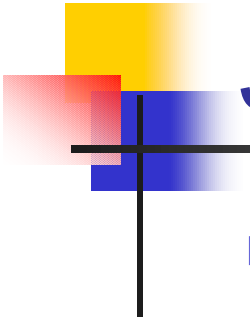
Grammar Design 2

- Everything is Strings
- Data modelling
  - Relationships as elements
  - modelling inheritance
- Plan to use schema datatypes
  - avoid weird types



Outline

- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- JDOM/DOM4J
- Performance
- Xerces Architecture



JDOM

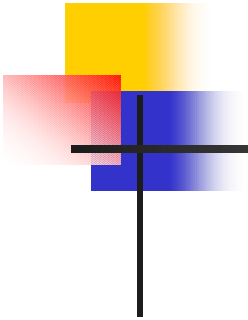
- JDOM Goals
  - Lightweight
  - Java Oriented
    - Uses Collections
  - Provides input and output (serialization)
- Class not interface based

JDOM Input

```
public static void printElements(Element e, OutputStream out)
       throws IOException, JDOMException {
  out.write(("\n===== " + e.getName() + ": \n").getBytes());
  out.flush();
  for (Iterator i=e.getChildren().iterator(); i.hasNext(); ) {
    Element child = (Element)i.next();
    printElements(child, out);
    }
  out.flush();
}

public static void main(String args[]) {
  SAXBuilder b = new
                 SAXBuilder("org.apache.xerces.parsers.SAXParser");
  try {
    Document d = b.build("file:///Work/fm0.91_full.rdf");
    printElements(d.getRootElement(),System.out);
  } catch (Exception e) {
  }
}
```

JDOM Output

```
public static void main(String args[]) {
  SAXBuilder builder = new
             SAXBuilder("org.apache.xerces.parsers.SAXParser");
  try {
    Document d = builder.build("file:///Work/fm0.91_full.rdf");
    XMLOutputter o = new XMLOutputter("\n> "); // indent
    o.output(d, System.out);
  } catch (Exception e) {
  }
}
```

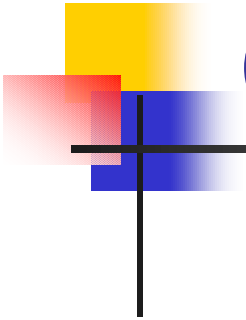


DOM4J

- DOM4J History
  - JDOM Fork
- DOM4J Goals
  - Java2 Support
    - Collections
  - Integrated XPath support
  - Interface based
  - Schema data type support
    - From schema file, not PSVI

DOM4J XPath

```
public static void main(String args[]) {
  try {
    SAXReader r = new SAXReader("org.apache.xerces.parsers.SAXParser");
    Document d = r.read("file:///Work/ASF/ApacheCon/fm0.91_full.rdf");
    String xpath = "//title";
    List list = d.selectNodes( xpath );
    System.out.println( "Found: " + list.size() + " node(s)" );
    System.out.println( "Results:" );
    XMLWriter writer = new XMLWriter();
    for ( Iterator iter = list.iterator(); iter.hasNext(); ) {
      Object object = iter.next();
      writer.write( object );
      writer.println();
    }
    writer.flush();
  } catch (Exception e) {
    e.printStackTrace();
  }
}
```



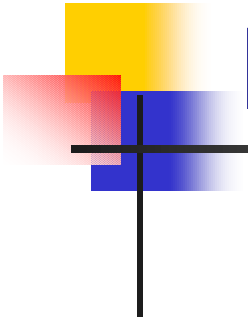
Outline

- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- JDOM/DOM4J
- Performance
- Xerces Architecture

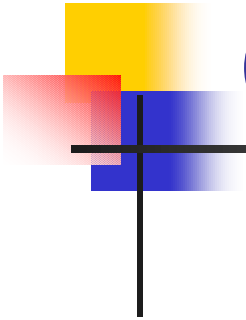
Sosnoski Benchmarks

- http://www.sosnoski.com/opensrc/xmlbench/index.html
- Results
  - Xerces-J DOM is better than JDOM/DOM4J in almost all cases – both in time and space.
  - One exception is serialization
  - Xerces-J Deferred DOM proves an excellent technique where applicable.

Performance Tips



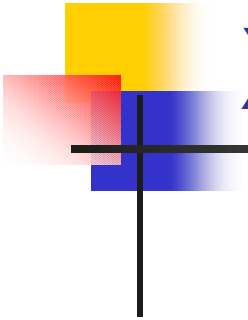
- skip the DOM
- reuse parser instances (code)
  - reset() method
- defaulting is slow
- external anythings are slower, entities are slower
- only validate if you have to, validate once for all time
- use fast encoding (US-ASCII)



Outline

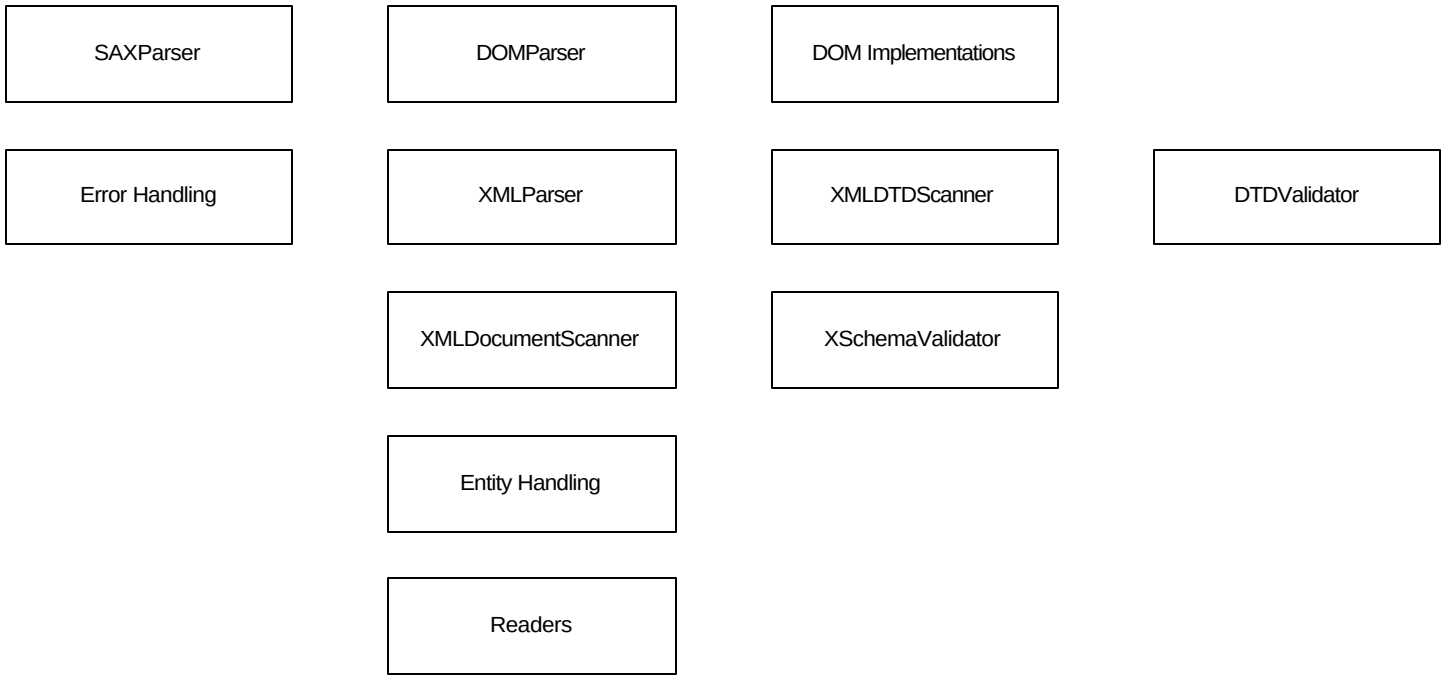
- Overview
- Basic XML Concepts
- SAX Parsing
- DOM Parsing
- JAXP
- Namespaces
- XML Schema
- Grammar Access
- Round Tripping
- Grammar Design
- Performance
- JDOM/DOM4J
- Xerces Architecture

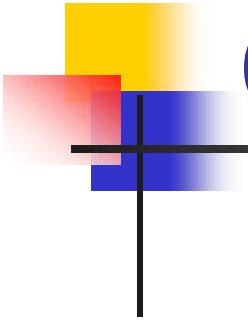
Xerces Architecture



- Use SAX based infrastructure where possible to avoid reinventing the wheel
- Modular / Framework design
- Prefer composition for system components

Xerces1 Architecture Diagram





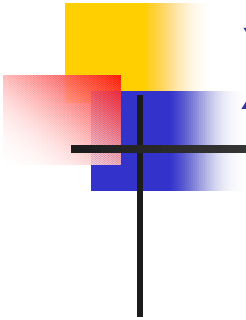
Concurrency

- “Thread safety”
  - Xerces allows multiple parser instances, one per thread
  - Xerces does not allow multiple threads to enter a single parser instance.

- `org.apache.xerces.framework.XMLParser#parseSome()`

Things we don’t do

- HTML parsing (yet)
  - java Tidy http://www3.sympatico.ca/ac.quick/jtidy.html
- Grammar caching (yet)
- Parse several documents in a stream

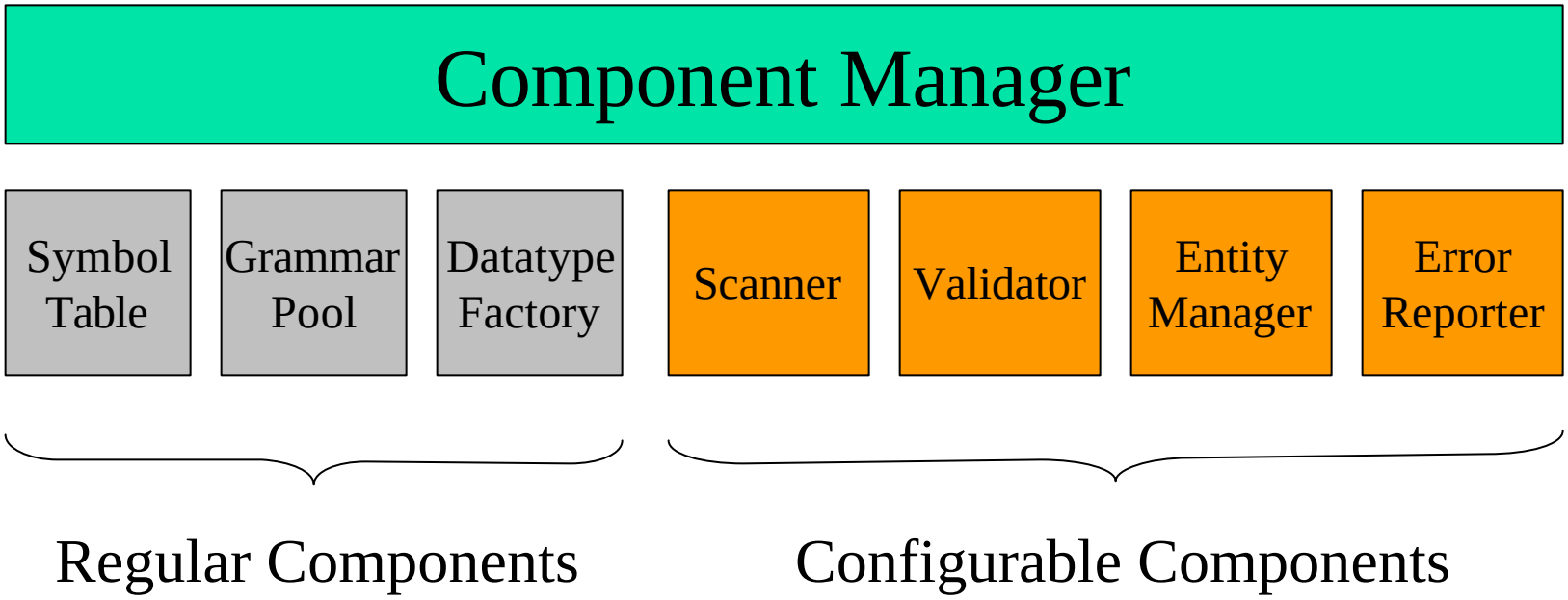


Xerces 2

- Alpha
- XNI
- Remove deferred transcoding
- Next steps
  - Conformance
  - Schema
  - Grammar Caching
  - DOM L3
- Need contributors!

Xerces 2 Architecture Diagram

- Courtesy Andy Clark, IBM TRL



Thank You!

- http://xml.apache.org
- twl@apache.org