XML for Java Developers
G22.3033-002

Session 8 - Main Theme
XML Information Rendering (Part II)

Dr. Jean-Claude Franchitti

New York University
Computer Science Department
Courant Institute of Mathematical Sciences

Agenda

- Summary of Previous Session
- XML/XSL and JSP/JavaBeans Rendering Technology
- Internationalization Issues
- Web Content Accessibility Guidelines (WCAG)
- Assignment 4a+4b (due next week)
Summary of Previous Session

- Extensible Stylesheet Language Transformation (XSL-T)
- Extensible Stylesheet Language Formatting Object (XSL-FO)
- XML and Document/Content Management
- Introduction to XML Application Servers
- Working with XSLT-T and XSL-FO Processors

XML-Based Rendering Development

- XML Software Development Methodology
  - Language + Stepwise Process + Tools
  - Rational Unified Process (RUP) vs. “XML Unified Process”
- XML Application Development Infrastructure
  - Metadata Management (e.g., XMI)
  - XSLT, XPath XSL-FO APIs (JAXP, JAXB, JDOM, SAX, DOM)
  - XML Tools (e.g., XML Editors, Apache’s FOP, Antenna House’s XSL Formatter, HTML/CSS1/2/3, XHTML, XForms, WCAG)
- XML Applications Involved in the Rendering Phase:
  - Application(s) of XML
  - XML-based applications/services (markup language mediators)
    - MOM, POP, Other Services (e.g., persistence)
  - Application Infrastructure Frameworks
XML Data Rendering Patterns

- Manipulating and Rendering XML Structures Using Java
  - XSL-T
    - Transform
    - Sort
    - Output
  - XSL-T + -FO
    - Format
    - Output
- Querying will be covered separately

Part I

XML Application Services and XML Rendering Technology
MVC Review


- MVC architecture decouples the code to handle user actions (controller), the data and business logic (Model), and the presentation (View)

![Diagram of MVC architecture]

Implementing the “V” of “MVC” Using JSPs

- When the view is implemented as a JSP, the controller object (e.g., servlet) forwards processing of the request and the response to a JSP view
- Controller adds a reference to the model object to the user’s session or request object
- JSP gets a handle on the model object and constructs the HTML or other markup to be returned to the client
Implementing the “V” of “MVC” Using JSPs

(continued)

Implementing the “V” of “MVC” Using XSL

- When the view is implemented in XSL, the basic flow of the transaction remains the same
- The model is represented in an XML format
- Once the model is built, the controller asks for a stylesheet to transform the XML into the desired rendition markup language
- XSL view may be implemented on the client rather than the server, so the controller may return XML to the client
Implementing the “V” of “MVC” Using XSL

(continued)

JSP vs. XSL

- Both approaches are based on industry standards
- The separation of business logic and data from presentation is clearer in the XSL approach
- Visual tools are available to automate the generation of code for either approach
- The XSL approach uses rule-based templates that are very modular and easy to maintain, while JSPs can be modularized using includes and tag libraries
- XSL style sheets are interchangeable between J2EE and .Net
- JSPs are faster than XSL style sheets
- XSL is an easier approach when the data to render is already in XML form (no need for parsing)
- XSL facilitates parallel development
JSP vs. Hybrid Approach

- Favor JSPs
  - When you already have a large investment in beans, tag libraries, and JSP frameworks
  - When high performance is an application requirement
- Consider Hybrid Approaches
  - Use JSP to create XML from objects and render into HTML via an XSL style sheet
  - JSP uses XSL as a “helper” to create output that gets included into an HTML stream
  - JSP is used for data capture, but XML results are displayed using XSL
  - XSL is used as a code generation technology to create a JSP
  - JSP is used as code generation technology to generate XSL

Towards XML Application Services

- Processing
  - DOM Extensions
  - Binding Extensions
  - Component Frameworks (reusable component models)
  - Model-Based Automation (MDA)
- Rendering
  - DOM 2.1.0, SAX 2.0, JAXP 1.2 & TraX, XSL-FO 1.0
  - Component Frameworks
- Querying
  - XQuery 1.0, XSLT 1.1/2.0, XPath 1.0/2.0
- Messaging
- Security (signatures encryption/decryption, etc.)
- etc.
Rendering Software Development

- Languages (XSL)
- Process (“XUP”)
- Frameworks (POP/MOM) - See XMLJ2EE
  - Cocoon 2.0
  - Xang
  - Batik
  - Etc.
- XSL Infrastructure
  - XSL-T Processors: Saxon 6.3, Xalan-J 2.5.1
  - XSL-FO Processors: fop 0.20.5

W3C’s UI Domain Slant

- Latest W3C UI Domain Structure
  - Document Formats (http://www.w3.org/DF/)
    - Amaya
    - Graphics
    - Internationalization
    - Math
    - Style
  - Interoperation (http://www.w3.org/Interaction/)
    - Device independence
    - SMIL
    - VoiceXML
    - XForms
Part II

XML Internationalization Issues

W3C Internationalization

- Internationalization Activity Statement
  - [http://www.w3.org/International/Activity.html](http://www.w3.org/International/Activity.html)
  - I18N and L10N features incorporated in (X)HTML, CSS, XML, RDF, SMIL, DOM, MathML, SVG, XPath, XSL(T), and XML Schema, HTTP 1.0
  - W3C’s Jigsaw, Amaya
- Problem is to document encodings being used
  - W3C Character Model
- Character Model for the World Wide Web
  - Normalization to Unicode Standard and ISO/IEC 10646
- Ruby Annotation
- Unicode in XML and other Markup Languages
Part III

Web Content Accessibility Guidelines
(WCAG 1.0)

Current State of XML Standards

- WCAG 1.0 (5/5/99) - W3C Recommendation
- WCAG 2.0 (3/28/01) - W3C Working Draft
  - Support wide range of languages
  - Easier to use by authoring tools developers
  - Easier to check conformance
- Techniques for WCAG 1.0 (3/20/00)
Guidelines Summary

- Provide Alternatives to Auditory and Visual content
- Avoid sole reliance on colors
- Use markup and style sheets
- Clarify natural language usage
- Create tables that transform well
- Make Sure that Pages that Feature New Technologies Transform Well
- Ensure User Control of Time-Sensitive Content Changes

Guidelines Summary (continued)

- Ensure Direct Accessibility of Embedded UIs
- Design for Device Independence
- Use Interim Solutions
- Use W3C Technologies and Guidelines
- Provide Context and Orientation Information
- Provide Clear Navigation Mechanisms
- Provide Clear and Simple Documents
Part IV

Web Services

Also see Session 4 Presentation: “Towards P2P Computing”
and
Session 8 Web Services Build and Test / Deploy and Publish Tutorials located under Demo Programs

Web Services Infrastructure

- Requirements:
  - Standard ways to format the messages to the services
    - XML
  - Send and receive service requests and results
    - SOAP
  - Represent detailed technical descriptions of how to use each service
    - WSDL
  - Search for services available on the Web
    - UDDI
Web Services and XML

- XML provides a standard way to format the messages to the services
- XML-formatted data is the “payload” of an XML message
- E.g., Purchase order:
  
  ```xml
  <Order>
    <Customer>
      <CustomerID>443</CustomerID>
      <Name>Discountpillows.com</Name>
    </Customer>
    <Item>
      <ProductID>12</ProductID>
      <Description>Fluffy white feathers</Description>
      <Quantity>1000 kg</Quantity>
    </Item>
  </Order>
  ```

Web Services and SOAP

- SOAP provides a consistent way to route an XML message to an appropriate service handler
- SOAP lightweight software “envelopes” are annotated to:
  - Indicate what service to invoke on the contents
  - Express what the payload contains
    - May not need to parse each payload completely
    - Provide context information (e.g., session IDs)
- SOAP is a standard for XML messaging formats
- SOAP supports platform-independent, extensible service requests
  - SOAP doesn’t specify a transport mechanism
Sample SOAP Envelope

```xml
<SOAP-ENV:Envelope>
  <SOAP-ENV:Header>
    <Transaction>123</Transaction>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <Order> <!-- etc. --></Order>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- SOAP header is optional and may indicate which service (aka., “Actor”) should process the message

Web Services and WSDL

- WSDL is an XML format for describing Web Services that supports
  - Services: Information or data processing available on the Web
  - Ports: Addresses where a service is available
  - Messages: Describe the structure of the data passed to and returned from services via ports
  - Operations: Tie messages and method calls together
  - PortTypes: Group related operations
  - Data Types: Used to define messages described in XSD
  - Bindings: “Ground” the ADTs and structures defined in WSDL to actual data representations

- WSDL was specified by Microsoft, IBM, and Ariba
Web Services and UDDI

- Universal Description, Discovery, and Integration (UDDI)
  - Find a service
  - Contact the business that provides that service
  - Figure out how to use the service

- UDDI registry’s service descriptions
  - White pages
    - Searchable, human-readable information
  - Yellow pages
    - Link business descriptions to standard business taxonomies and geographical descriptions
      - E.g., NAICS (US Government’s codes for industries)
  - Green pages
    - Describe online business processes and services
    - Describe how to access the services over the Web using standard protocols
    - Optionally categorize the services

Resource Description Framework (RDF)

- Language to describe resources
- Use metadata (data about data) to describe Web resources
- Provides interoperability between applications that exchange machine-understandable information on the Web
- Use XML as a syntax
RDF Application Areas

http://www.xml.com/pub/a/2001/01/24/rdf.html

- Resource discovery - better search engine capabilities
- Cataloging - describe content and content relationship (web pages) – (RSS, CDF)
- Intelligent software agents - knowledge sharing
- Content Rating (PICS)
- Collections of pages
- Intellectual property rights
- Privacy preferences and policies (P3P)
- Digital signatures - build the "Web of Trust"

Part V

Conclusions
Summary

- XML Application Services support a stepwise approach towards the development of XML-based system architectures
- W3C Internationalization is moving towards normalization based on a web character model
- I18N/L10N support needs to keep being added into XSL, and CSS3. XML Query, XML Protocols, XForms, and newer XML technologies
- Content Accessibility Guidelines are targeted to Web and Authoring Tools developers to ensure that Web content is accessible to people with disabilities
- Web Services provide a standard for B2B interactions over the Web

Readings

- Readings
  - XML and Java: Chapter 13
  - Processing XML with Java: Chapter 2, Appendix B
  - Developing Java Web Services: Chapters 3-5, 7, and 10
  - Handouts posted on the course web site
  - Review XQuery 1.0 and XPath 2.0 status on W3C web site (http://www.w3.org/TR/query-datamodel/)
- Project Frameworks Setup (ongoing)
  - Apache’s Web Server, TomCat/JRun, and Cocoon
  - Apache’s Xerces, Xalan, Saxon
  - Antenna House XML Formatter, Apache’s FOP, X-smiles
  - Publishing Systems at http://www.xmlsoftware.com
  - Visibroker 4.5 (or BES 5.2), WebLogic 6.1-8.1, WAS 5.0
  - POSE & KVM (See Session 3 handout)
Assignment

- Assignment #4:
  - This part of the project focuses on the application content model design/development using XML information rendering technology. The design/development process should adhere to the following steps: (a) Identifying rendering/transformation targets, (b) Defining the optimal rendering approach for each target, (c) Considering data rendering issues when designing an overall application data model
  - More specific project related information, and extra credit assignments will be provided during the session

Next Session:
XML Information Retrieval (Part I)

- Applications of XML to Database Technology
- XML Queries
- XML Query Languages
- XML Registries API
- .Net Web Services