Computer Science E-259

XML with Java, Java Servlet, and JSP

Lecture 13: Conclusion

7 January 2008

David J. Malan

malan@post.harvard.edu

1

Last Time

Ajax at HBS

- JDOM
  `http://www.jdom.org/`
- JXPath
  `http://jakarta.apache.org/commons/jxpath/`
- `XMLHttpRequest`
- Yahoo! UI Library
  `http://developer.yahoo.com/yui/`

JDOM

Versus DOM

- JDOM

```
Document doc = new Document();
Element e = new Element("root");
e.setText("This is the root");
doc.addContent(e);
```

- DOM

```
DocumentBuilderFactory factory =
 DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();
Document doc = builder.newDocument();
Element root = doc.createElement("root");
Text text = doc.createText("This is the root");
root.appendChild(text);
doc.appendChild(root);
```

Excerpted from http://www.servlets.com/speaking/jdom-javaone.pdf.

JXPath

Versus Plain Ol' Java

- JXPath
  ```
  Address address = (Address)JXPathContext.newContext(vendor).
   getValue("locations[address/zipCode='90210']/address");
  ```

- Plain Ol' Java
  ```
  Address address = null;
  Collection locations = vendor.getLocations();
  Iterator it = locations.iterator();
  while (it.hasNext()){
   Location location = (Location)it.next();
   String zipCode = location.getAddress().getZipCode();
   if (zipCode.equals("90210")){
     address = location.getAddress();
     break;
   }
  }
  ```

4

Excerpted from http://jakarta.apache.org/commons/jxpath/.

XMLHttpRequest

History

- The XMLHttpRequest Object is a Working Draft since 10/07.
  - "The XMLHttpRequest Object specification defines an API that provides scripted client functionality for transferring data between a client and a server."

Excerpted from http://www.w3.org/TR/XMLHttpRequest/.

XMLHttpRequest

Methods

- **abort**
- **getAllResponseHeaders**
- **getResponseHeader**
- **open**
- **send**
- **setRequestHeader**

XMLHttpRequest

Properties

- `onreadystatechange`
- `readyState`
- `responseText`
- `responseXML`
- `status`
- `statusText`

Computer Science E-259

This Time

- XForms 1.0 (Third Edition)
- XLink 1.0
- XPointer 1.0
- XInclude 1.0 (Second Edition)
- XML Base
- XML Encryption
- XML Key Management 2.0
- XML Signature
- Data Modeling
- Computer Science E-259

8

XForms 1.0 (Third Edition)

History

- XForms 1.0 (Third Edition) is a Recommendation since 10/07.
  - "XForms is an XML application that represents the next generation of forms for the Web. By splitting traditional XHTML forms into three parts—XForms model, instance data, and user interface—it separates presentation from content, allows reuse, gives strong typing—reducing the number of round-trips to the server, as well as offering device independence and a reduced need for scripting."
  - "XForms is not a free-standing document type, but is intended to be integrated into other markup languages, such as XHTML or SVG."

Excerpted from http://www.w3.org/TR/xforms/.

XForms 1.0 (Third Edition)

What

- Successor of HTML forms
- Separate data from presentation
- Use XML to define forms
- Use XML to transport data
- Device-independent

Adapted from `http://www.w3schools.com/xforms/xforms_intro.asp`.

XForms 1.0 (Third Edition)

XHTML Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC
    "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <title/>
  </head>
  <body>
    <form action="payment.asp" method="post">
      <p><b>Select Payment Method</b></p>
      <p>
        Cash <input type="radio" name="as" value="cash" />
        Credit Card <input type="radio" name="as" value="credit" checked />
      </p>
      <p>Card Number:<br /><input type="text" id="cc" /></p>
      <p>Expiration Date:<br /><input type="text" name="exp" /></p>
      <p><input type="submit" /></p>
    </form>
  </body>
</html>
```

Adapted from http://www.w3schools.com/xforms/xforms_xhtml.asp.

XForms 1.0 (Third Edition)

XHTML Example

Select Payment Method

Cash ○ Credit Card ◉

Card Number:

[   ]

Expiration Date:

[   ]

[Submit Query]

Excerpted from http://www.w3schools.com/xforms/xforms_xhtml.asp.

XForms 1.0 (Third Edition)

XForms Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC
    "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"
    xmlns:xform="http://www.w3.org/2001/08/xforms">
 <head>
   <xform:xform id="payment">
     <xform:submitInfo action="submit.asp" method="post"/>
   </xform:xform>
   <xform:instance>.....</xform:instance>
   <xform:model>........</xform:model>
   <xform:bindings>.....</xform:bindings>
  </head>
  ...
```

XForms 1.0 (Third Edition)

XForms Example

```
...
<body>
  <xform:selectOne xform="payment" ref="as" >
    <xform:caption>Select Payment Method</xform:caption>
    <xform:choices>
      <xform:item value="cash"><xform:caption>Cash</xform:caption></xform:item>
      <xform:item value="credit"><xform:caption>Credit</xform:caption></xform:item>
    </xform:choices>
  </xform:selectOne>
  <xform:input xform="payment" ref="cc">
    <xform:caption>Credit Card Number</xform:caption>
  </xform:input>
  <xform:input xform="payment" ref="exp">
    <xform:caption>Expiration Date</xform:caption>
  </xform:input>
  <xform:submit xform="payment"><xform:caption>Submit</xform:caption></xform:submit>
</body>
</html>
```

14

Adapted from http://www.w3schools.com/xforms/xforms_xhtml.asp.

XForms 1.0 (Third Edition)

Processing

```
<envelope>
  <body>
    <as>Credit</as>
    <cc>1235467789012345</cc>
    <exp>2001-08</exp>
  </body>
</envelope>
```

15

XLink 1.0

History

- XML Linking Language (XLink) Version 1.0 is a recommendation since 6/01
  - "This specification defines the XML Linking Language (XLink), which allows elements to be inserted into XML documents in order to create and describe links between resources. It uses XML syntax to create structures that can describe links similar to the simple unidirectional hyperlinks of today's HTML, as well as more sophisticated links."
- Influenced by
  - HTML
  - HyTime (ISO/IEC 10744-1992)
 - Text Encoding Initiative (TEI) Guidelines

16

Excerpted from http://www.w3.org/TR/xlink/.

XLink 1.0

Goals

- Assert linking relationships among more than two resources
- Associate metadata with a link
- Express links that reside in a location separate from the linked resources

Excerpted from http://www.w3.org/TR/xlink/.

XLink 1.0

Simple Links

Excerpted from http://www.w3.org/TR/xlink/.
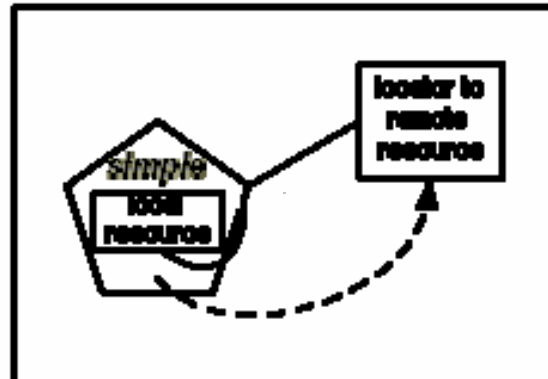
XLink 1.0

Simple Links

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
        "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg:svg height=".5in" width="5.5in" xml:space="preserve"
   xmlns:svg="http://www.w3.org/2000/svg"
   xmlns:xlink="http://www.w3.org/1999/xlink">


  <svg:a xlink:href="http://www.fas.harvard.edu/~cscie259/"
   target="_blank">
    <svg:text style="fill:red;" y="15">This is CSCI E-259.</svg:text>
  </svg:a>


</svg:svg>
```

XLink 1.0

Simple Links

```
<?xml version="1.0"?>
<!DOCTYPE ARTIST SYSTEM "artist.dtd">
<ARTIST xmlns:xlink="http://www.w3.org/1999/xlink"
        xlink:type="simple"
        xlink:href="pic_dm.jpg"
        xlink:title="Dave Matthews"
        xlink:role="image"
        xlink:arcrole="Photo for Dave Matthews"
        xlink:actuate="onRequest"
        xlink:show="replace"
>Dave Matthews</ARTIST>
```

20

XLink 1.0

Attributes of Simple Links

- **xlink:type**
  - Specifies link's type.
- **xlink:href**
  - URL of target resource.
- **xlink:title**
  - Link's title.
- **xlink:role**
  - Target's role.
- **xlink:arcrole**
  - Relationship between source and target.
- **xlink:actuate**
  - Defines moment of actuation. Must be **none**, **onLoad**, **onRequest**, or **other**.
- **xlink:show**
  - Defines behavior of link upon actuation. Must be **embed**, **new**, **none**, **other**, or **replace**.

21

XLink 1.0

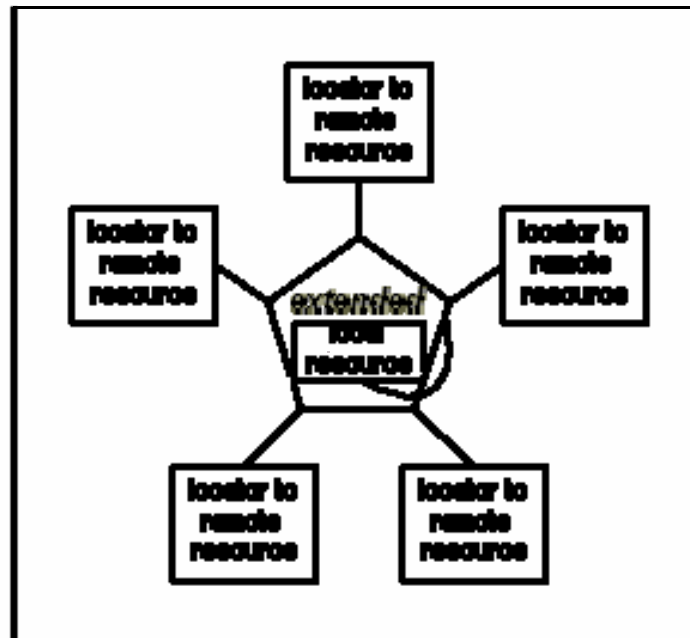## Simple Links

```
<!ELEMENT ARTIST (#PCDATA)>
  <!ATTLIST ARTIST
    xmlns:xlink    CDATA   #FIXED      "http://www.w3.org/1999/xlink"
    xlink:type     CDATA   #FIXED      "simple"
    xlink:href     CDATA   #REQUIRED
    xlink:role     CDATA   #IMPLIED
    xlink:title    CDATA   #IMPLIED
    xlink:actuate  CDATA   #IMPLIED
    xlink:arcrole  CDATA   #IMPLIED>
]>
```

22

XLink 1.0

Extended Links

XLink 1.0

Extended Links

```xml
<?xml version="1.0" standalone="yes"?>
<!-- from http://wps.aw.com/aw_webwizard/0,6065,184738-,00.html -->
<!DOCTYPE courses [
  <!ELEMENT  courses  (locator*, resource*, arc*, title?)>
  <!ATTLIST  courses
    xmlns:xlink  CDATA  #FIXED  "http://www.w3.org/1999/xlink"
    xlink:type   CDATA  #FIXED  "extended">
  <!ELEMENT locator EMPTY>
  <!ATTLIST locator
    xlink:href   CDATA  #REQUIRED
    xlink:title  CDATA  #IMPLIED>
]>
<courses xmlns:xlink="http://www.w3.org/1999/xlink"
         xlink:type="extended">
  <locator xlink:type="locator"
           xlink:href="courses/xml101.xml"
           xlink:title="XML 101"/>
  <locator xlink:type="locator"
           xlink:href="courses/advxml.xml"
           xlink:title="Advanced XML"/>
  <locator xlink:type="locator"
           xlink:href="courses/bw.xml"
           xlink:title="Basket Weaving"/>
</courses>
```
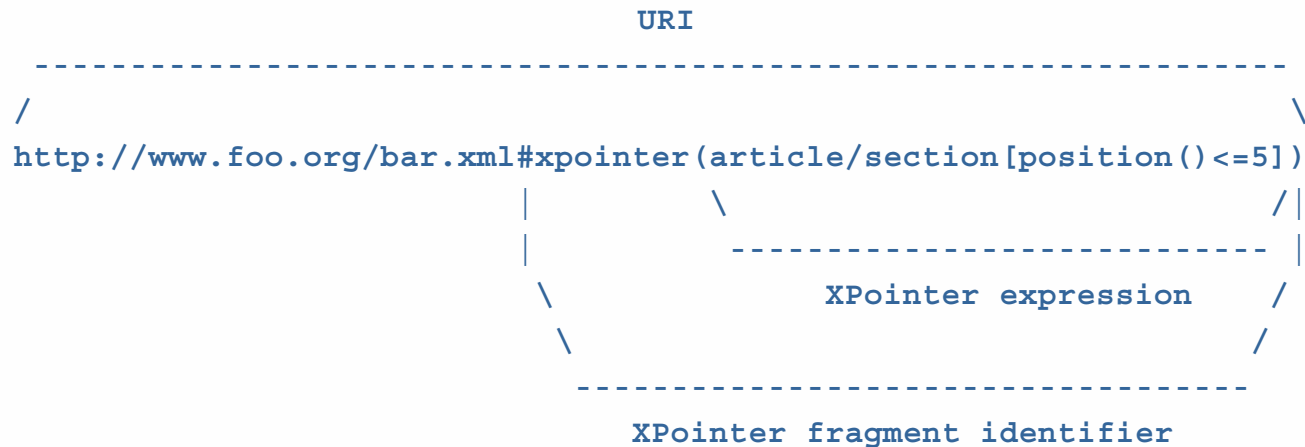
24

XPointer

History

- XPointer is specified by three recommentations as of 3/03
    - "This specification defines the XML Pointer Language (XPointer) Framework, an extensible system for XML addressing that underlies additional XPointer scheme specifications. The framework is intended to be used as a basis for fragment identifiers for any resource whose Internet media type is one of `text/xml`, `application/xml`, `text/xml-external-parsed-entity`, or `application/xml-external-parsed-entity`. Other XML-based media types are also encouraged to use this framework in defining their own fragment identifier languages."

25

XPointer

Why, what, and how?

- An extension of XPath which is used by XLink to locate remote link resources
- Relative addressing: allows links to places with no anchors
- Flexible and robust: XPointer/XPath expressions often survive changes in the target document
- Can point to substrings in character data and to whole tree fragments

```
                              URI
       ------------------------------------------------------------
      /                                                            \
   http://www.foo.org/bar.xml#xpointer(article/section[position()<=5])
                            |         \                        /|
                            |          -------------------------- |
                             \             XPointer expression   /
                              \                                  /
                               -----------------------------------
                                  XPointer fragment identifier
```

Excerpted from http://www.w3.org/TR/xquery/.

XPointer

Points

- **point(1.0)** is just inside the beginning of the **p** element
- **point(1.2)** is between the end of the **em** element and the following text node (which contains "**world.**")
- **point(.0)** immediately precedes the root node
- **point(1/2/1.1)** immediately following the "**b**" in the middle text node
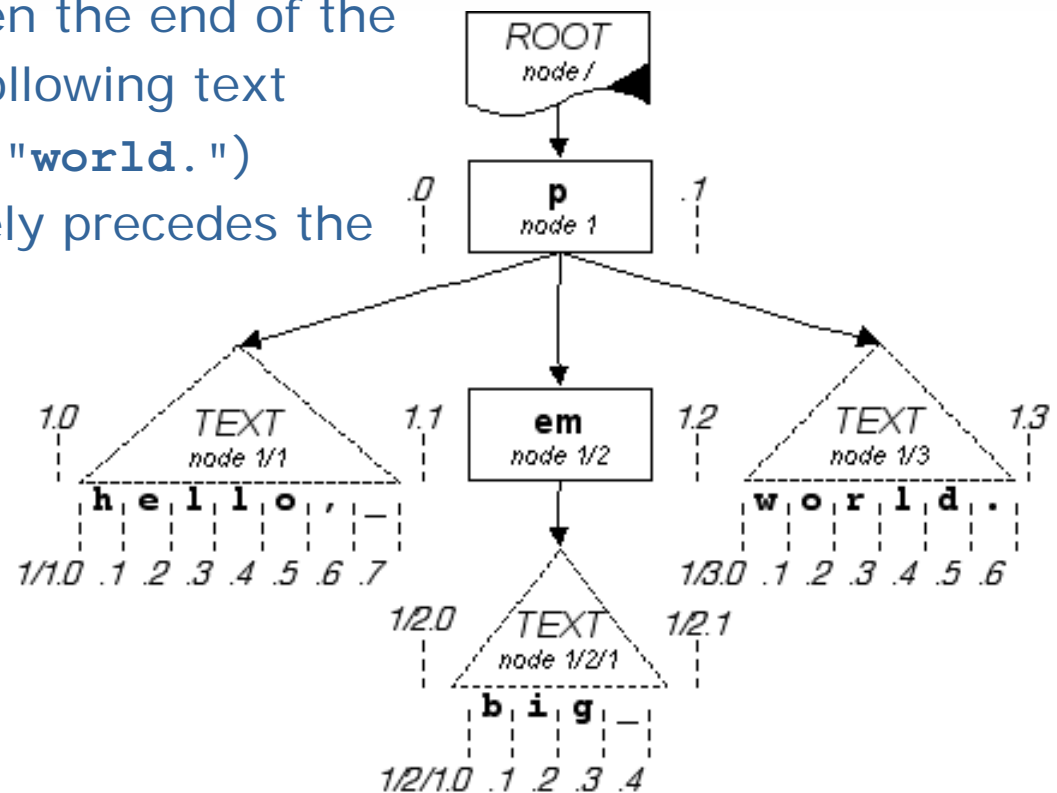
XInclude 1.0 (Second Edition)

History

- XML Inclusions (XInclude) Version 1.0 (Second Edition) is a recommendation since 11/06
  - "Many programming languages provide an inclusion mechanism to facilitate modularity. Markup languages also often have need of such a mechanism. This specification introduces a generic mechanism for merging XML documents (as represented by their information sets) for use by applications that need such a facility."

Excerpted from http://www.w3.org/TR/xinclude/.

XInclude 1.0 (Second Edition)

Example

```
<xi:include xmlns:xi="http://www.w3.org/2001/XInclude"
                xi:href="rowling_bio.xml"
                xi:parse="xml"
                xi:encoding="iso-8859-1"
/>
```

XML Base

History

- XML Base is a recommendation since 6/01
  - "This document proposes a facility, similar to that of HTML **BASE**, for defining base URIs for parts of XML documents."

Excerpted from http://www.w3.org/TR/xmlbase/.

XML Base

With XLink 1.0 and XInclude 1.0

- XLink 1.0
  - ```
    <AUTHOR xmlns:xlink="http://www.w3.org/1999/xlink"
            xlink:type="simple"
            xml:base="http://harrypotter.warnerbros.com/"
            xlink:href="rowling_bio.xml"
            xlink:actuate="onLoad"
            xlink:show="embed"
    >J.K. Rowling</AUTHOR>
    ```
- XInclude 1.0
  - ```
    <xi:include xmlns:xi="http://www.w3.org/2001/XInclude"
                xml:base="http://harrypotter.warnerbros.com/"
                xi:href="rowling_bio.xml"
                xi:parse="xml"
                xi:encoding="iso-8859-1"
    />
    ```

31

XML Encryption

History

- XML Encryption (Syntax and Processing) is a Recommendation since 12/02.
    - "This document specifies a process for encrypting data and representing the result in XML. The data may be arbitrary data (including an XML document), an XML element, or XML element content. The result of encrypting data is an XML Encryption element which contains or references the cipher data."

Excerpted from http://www.w3.org/TR/xmlenc-core/.

XML Encryption

Example

```
<PaymentInfo xmlns='http://example.org/paymentv2'>
  <Name>John Smith<Name/>
  <CreditCard Limit='5,000' Currency='USD'>
    <Number>4019 2445 0277 5567</Number>
    <Issuer>Bank of the Internet</Issuer>
    <Expiration>04/02</Expiration>
  </CreditCard>
</PaymentInfo>


<PaymentInfo xmlns='http://example.org/paymentv2'>
  <Name>John Smith<Name/>
  <CreditCard Limit='5,000' Currency='USD'>
    <Number>
      <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
              Type='http://www.w3.org/2001/04/xmlenc#Content'>
        <CipherData><CipherValue>A23B45C56</CipherValue></CipherData>
      </EncryptedData>
    </Number>
    <Issuer>Bank of the Internet</Issuer>
    <Expiration>04/02</Expiration>
  </CreditCard>
</PaymentInfo>
```

Excerpted from http://www-106.ibm.com/developerworks/xml/library/s-xmlsec.html/.

33

XML Key Management 2.0

History

- XML Key Management 2.0 is a Recommendation in two parts since 6/05
    - "This document specifies protocols for distributing and registering public keys, suitable for use in conjunction with the W3C Recommendations for XML Signature and XML Encryption. "

Excerpted from http://www.w3.org/TR/xkms2/.

XML Key Management 2.0

Request

```xml
<?xml version="1.0" encoding="utf-8"?>
<CompoundRequest Id="I264f5da49b1ff367d4e7aef1f7a1df1a"
     Service="http://test.xmltrustcenter.org/XKMS"
     xmlns="http://www.w3.org/2002/03/xkms#">
  <LocateRequest Id="I8c26be5f1b4dd228b43fb6eaee285faa"
       Service="http://test.xmltrustcenter.org/XKMS">
    <RespondWith>KeyValue</RespondWith>
    <QueryKeyBinding>
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <X509Data>
          <X509Certificate>MIICEDCCAX2gAwIBAgIQimXeUAxYJbJMady9vV1bLjAJBgUrDg
              MCHQUAMBIxEDAOBgNVBAMTB1Rl
c3QgGQ0EwHhcNMDMwODE1MDcwMDAwWhcNMDUwODE1MDY1OTU5WjArMSkwJwYDVQQDEyBBbGljZSBB
YXJkdmFyayBPPUFsaWNlIENvcnAgQz1VUzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA0nIs
mR+aVW2egl5MIfOKy4HuMKkk9AZ/IQuDLVPlhzOfgngjVQCjr8uvmnqtNu8HBupui8LgGthO6U9D
0CNT5mbmhIAErRADUMIAFsi7LzBarUvNWTqYNEJmcHsAUZdrdcDrkNnG7SzbuJx+GDNiHKVDQggP
BLc1XagW20RMvokCAwEAAaNWMFQwDQYDVR0KBAYwBAMCBkAwQwYDVR0BBDwwOoAQAaVOkaVLLKoF
mLN37pC8uqEUMBIxEDAOBgNVBAMTB1Rlc3QgQ0GCEC4MndUXjPG1TZxVKg+HutAwCQYFKw4DAh0F
AAOBgQABU91ka7IlkXCfv4Zh2Ohwgg2yObtY3+6C/BTFGrOEBJDy+DoxJ/NuBF18w3rrrR18xE6j
NKYLCQb8zUGk4QOG5Y+HT/QTTFvWkiOLXcpTuhnOhXatr42FoYpDkjx2QWK+J5Q2l/Rgjgc/0ZV8
U/kD8UuRkXp4AZh7QsiX8AcO0w==</X509Certificate>
        </X509Data>
      </KeyInfo>
      <KeyUsage>Signature</KeyUsage>
    </QueryKeyBinding>
  </LocateRequest>
  <LocateRequest Id="If8e63d729384ad35498e7b65b3dc785e"
       Service="http://test.xmltrustcenter.org/XKMS">
    <RespondWith>KeyName</RespondWith>
    <RespondWith>KeyValue</RespondWith>
    <RespondWith>X509Cert</RespondWith>
    <RespondWith>X509Chain</RespondWith>
    <RespondWith>PGPWeb</RespondWith>
    <RespondWith>PGP</RespondWith>
    <QueryKeyBinding>
      <KeyUsage>Encryption</KeyUsage>
      <UseKeyWith Application="urn:ietf:rfc:2440"
          Identifier="bob@bobcorp.test" />
      <UseKeyWith Application="urn:ietf:rfc:2633"
          Identifier="bob@bobcorp.test" />
    </QueryKeyBinding>
  </LocateRequest>
</CompoundRequest>
```

Excerpted from http://www.w3.org/TR/xkms2/.

XML Signature

History

- XML Signature is a Recommendation since 2/02.
  - "This document specifies XML digital signature processing rules and syntax. XML Signatures provide integrity, message authentication, and/or signer authentication services for data of any type, whether located within the XML that includes the signature or elsewhere."

Excerpted from http://www.w3.org/TR/xmldsig-core/.

XML Signature

Example

```
<Signature Id="MyFirstSignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
  <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
    20010315"/>
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
  <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
    <Transforms>
      <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
  </Reference>
</SignedInfo>
  <SignatureValue>MC0CFFrVLtRlk=...</SignatureValue>
  <KeyInfo>
    <KeyValue>
      <DSAKeyValue>
        <p>...</p><Q>...</Q><G>...</G><Y>...</Y>
      </DSAKeyValue>
    </KeyValue>
  </KeyInfo>
</Signature>
```

37

Data Modeling

Loose Guidelines

- The intricacies of effective data modeling is impossible to teach in a few slides, but…
- Here are a few loose guidelines when designing data models
- Avoid repeating data in your XML document
  - If there are relationships between objects in your data, express this with a reference
  -
```
<videos>
  ...
  <video>
    ...
    <actorRef>0005</actorRef>
    <actorRef>0122</actorRef>
    ...
  </video>
</videos>
...
<actors>
  ...
  <actor id="0005">Harrison Ford</actor>
  ...
</actors>
```

38

Data Modeling

Loose Guidelines

- Don't fall into relational "hacks": use XML as it was designed to be used
  - If you come from the relational database world, you may be accustomed to representing an address with several numbered fields
    - address1, address2, address3, city, state
  - Don't do this in XML! Instead,
    - ```
      <address>
          <street>...</street>
          <street>...</street>
          <street>...</street>
          <city>...</city>
          <state>...</state>
      </address>
      ```

Data Modeling

Loose Guidelines

- When creating a new field for an element, always ask yourself whether it should be a child element or an attribute
  - If the field is tightly bound with the parent element and is just a simple string (*e.g.*, an id or name), consider making it an attribute
  - If the field has any structure or complexity (or you can foresee extending it to have such), make it a child element
  - Think of a Java class: is the member a `String`, or a reference to another class object?

CSCI E-75: Building Dynamic Websites

http://www.computerscience75.org/

"Today's websites are increasingly dynamic. Pages are no longer static HTML files but instead generated by scripts and database calls. User interfaces are more seamless, with technologies like Ajax replacing traditional page reloads. This course teaches students how to build dynamic websites with Ajax and with Linux, Apache, MySQL, and PHP (LAMP), one of today's most popular frameworks. Students learn how to set up domain names with DNS, how to structure pages with XHTML and CSS, how to program in JavaScript and PHP, how to configure Apache and MySQL, how to design and query databases with SQL, and how to use Ajax with both XML and JSON. The course discusses issues of security, scalability, and cross-browser support."

41

Computer Science E-259

*"XML is beautiful, but as with beautiful people,
it is neither easy to get along with nor quick."*

*"XML's strength is its wide adoption and excellent tools.
XML itself is not that exciting."*

42

Computer Science E-259

Goals of the Course

- Cut through the hype and get to the value
- Focus on
  - practicality: what you need to know to do real work
  - applications: what are the tools and technologies necessary to put XML to use
  - possibilities: what are some of the most common ways XML is being used in applications
- Emphasize understanding from the bottom up

43

Computer Science E-259

Rewards of the Course

- A solid understanding of XML, the syntax, data structures, and algorithms that surround it
- Xperience
    - DTD, SVG, XForms, XInclude, XLink, XML Base, XML Encryption, XML Key Management, XML Namespaces, XML Schema, XML Signature, XPath, XPointer, XQuery, XSL-FO, and XSLT
    - SAX and DOM
    - JAXP and TrAX
    - JavaServer Pages and Java Servlet
    - HTTP, SOAP, web services, WSDL
    - FOP, Stylus Studio, Tomcat, Xalan, Xerces, XMLSpy…

44

Computer Science E-259

XML with Java, Java Servlet, and JSP

Lecture 13: Conclusion

7 January 2008

David J. Malan

malan@post.harvard.edu

45