Intro to XML with Steph Beene and Doug Rice

PROGRAM SYNOPSIS:

This short course will introduce you to what XML is, what it does, and how it is used. We will also look at the differences between HTML and XML, DTDs and XSDs, and common editors used for validation and markup. This handout provides links to additional tutorials, reference materials and available editors.

WHAT IS XML?

- e**X**tensible **M**arkup **L**anguage: a standard for marking up data from W3C group
    - XML is made up of tags used to apply metadata to text
    - It helps describe the text and put it into categories
    - Extensible – users define own tags, document types and relationships
- XML *defines* data, but *does not display* it – in this way, it is different than HTML or CSS.
- XML was designed to *structure, store, and transport data*.
    - It's a transmission standard for data – to and from websites, digital repositories, and databases.
    - It's a container for other data standards – no matter what schema metadata was originally written in, XML contains it so that it is *interoperable* with other standards. Adobe and Microsoft also use XML to contain data about their software and communicate between programs.
    - It's a query language (especially for federated search results): In information search and retrieval, XML-encoded queries return more precise results. XML plays well with others!
- *XML is a complement to HTML*. In most web applications, XML is used to transport data, while HTML is used to format and display the data.
    - In some cases, you will see an option to attach an XSL stylesheet – this is like the CSS of the XML world, and has supplanted HMTL for the most part.
- **You** define the tags. Your tags can be anything - the only caveat is that your tags **MUST** match a set of rules you set somewhere:
    - **DTD** (**D**ocument **T**ype **D**efinition)

- **XSD** (**X**ML **S**chema **D**efinition): currently the de facto standard for describing XML documents – you'll see this with all of the LOC standards (for example, METS, MODS, MADS, even EAD now).

WHAT IS A DTD?

- **D**ocument **T**ype **D**efinition
- It is a set of rules used to define an XML document.
- An XML document is checked against a DTD in order to validate it.
- A DTD has syntax for rule definition *(see table, below)*.

Symbol in DTD	*Rule*
?	Zero or one of element is allowed
*	Zero or one of element is allowed
+	One or more of the element is allowed
A | B	Choice of "A" or "B" is required
#PCDATA	Text (literally, parsed character data)
C, D, E	"C" must be listed first (if it exists), then "D" (if it exists), then "E" (if it exists)

- **Here is an example table of what DTD rules mean for data:**

DTD Rule	*Description*
<!ELEMENT person (name, profession*, interests+)>	First element, <name> is mandatory; Zero or one of the <profession> element is allowed; and one or more of the <interests> element is allowed
<!ELEMENT name (first_name, last_name)>	Mandatory elements (<first_name><last_name>).
<!ELEMENT first_name (#PCDATA)>	The value for this field/element (<first_name>) will be Text, or Parsed Character Data

<!ELEMENT profession (data mining | cataloger)>	Choice of attribute for the <profession> element: data mining OR cataloger is required
<!ELEMENT interests (#PCDATA)>	The value for this field/element (<interests>)will be Text, or Parsed Character Data

---

WHAT IS AN XSD?

- **X**ML **S**chema **D**efinition**:**
  - Operates just like a DTD, except the syntax is different.
  - The XML document will be checked against the XSD for compliance to the rules.
  - It is more extensible and flexible than a DTD and is therefore usurping the DTD as the standard.
  - An element definition within the XSD must have a **name property**: this is the name that will appear in the XML document.
  - There is also a **type property**. This provides the description of what can be contained within the element when it appears in the XML document.
    - There are a number of predefined types, such as xs:string, xs:integer, xs:boolean or xs:date *(see example table, below).*

Sample XSD	***Sample XML***
<xs:element name="Creator_dob" type="xs:date"/>	<Creator_dob>2000-01-12T12:13:14Z</Creator_dob>
<xs:element name="Location_address" type="xs:string"/>	<Location_address>99 London Road</Location_address>
<xs:element name="OrderID" type="xs:int"/>	<OrderID>5756</OrderID>
<xs:element name="Body" type="xs:string"/>	<Body> (a type can be defined as a string

	but not have any content, this is not true of all data types however). </Body>

- Elements defined in the XSD have two rules which affect their allowed values:
  - **Default** means that if no value is specified in the XML document then the application reading the document (typically an XML parser or XML Data binding Library) should use the default specified in the XSD.
  - **Fixed** means the value in the XML document can only have the value specified in the XSD.
- Specifying how many times an element can appear is referred to as **cardinality**
  - Specified using the attributes minOccurs and maxOccurs.
- In this way an element can be mandatory, optional, or appear many times *(see example table, below)*.

Sample XSD	***Description***
<xs:element name="Creeator_dob" type="xs:date"/>	If we don't specify minOccurs or maxOccurs, then the default values of 1 are used, so in this case there has to be one and only one occurrence of Creator_dob
<xs:element name="Customer_order" type="xs:integer" minOccurs ="0" maxOccurs="unbounded"/>	Here, a customer can have any number of Customer_orders (even 0)
<xs:element name="Customer_hobbies" type="xs:string" minOccurs="2" maxOccurs="10"/>	In this example, the element Customer_hobbies must appear at least twice, but no more than 10 times

This example table is from http://www.liquid-technologies.com/Tutorials/XmlSchemas/XsdTutorial_01.aspx

- **Compositors**:
  - There are 3 types of compositors <xs:sequence>, <xs:choice> and <xs:all>.
  - These compositors allow us to determine how the child elements within them appear within the XML document.
  - ***Note***: The compositors <xs:sequence> and <xs:choice> can be nested inside other compositors, and be given their own minOccurs and maxOccurs properties. This allows for quite complex combinations to be formed *(see example table, below)*.

Compositor	*Description*
Sequence	The child elements in the XML document MUST appear in the order they are declared in the XSD schema.
Choice	Only one of the child elements described in the XSD schema can appear in the XML document.
All	The child elements described in the XSD schema can appear in the XML document in any order.

This example table is from http://www.liquid-technologies.com/Tutorials/XmlSchemas/XsdTutorial_01.aspx

Resources:

- O'Reilly describes the W3C XSD Standard: http://www.xml.com/pub/a/2000/11/29/schemas/part1.html
- TechTarget's definition and explanation of XSDs: http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci831325,00.html
- Liquid XML has an excellent tutorial on XSDs: http://www.liquid-technologies.com/Tutorials/XmlSchemas/XsdTutorial_01.aspx

A NOTE ON VALIDATION:

- If you are validating *without* a DTD this will just check for well-formedness of your tags (if each tag is closed): http://www.w3schools.com/xml/default.asp;
- If you are validating *with* a DTD or XSD, Oxygen (or the software you use) will compare your markup to the rules of the DTD/XSD. Validation here will check that parent-child elements are where they should be and conform to controlled fields.
    - For instance, in MODS, there are fields defined by the XSD as having to conform to MARC21 fields or LCSH vocabulary. Validation will check for this.
- Your DTD/XSD *must* be saved in the same folder as your markup. Oxygen will do this automatically if you load it upon starting a new document, or "associate" a schema or DTD with your document. With other programs, you will want to *make sure* that you have saved in the same folder.
- If you're using an online DTD/XSD put the *complete* URL into the "browse" window, starting with http://.

OXYGEN:

An XML editor available on all IT Lab computers, or available for a 30-day free trial from http://www.oxygenxml.com/download.html

- Allows you to load in a DTD or XSD in another open tab as the XML document you are marking up. This saves the step of saving to the same folder – it does it for you.
- Load an XML Schema or DTD from the desktop (or folder) or from a URL.
- You can check for well-formedness and validation with easily marked keys on your tool panel.
- A green "light" tells you your text validates.
- You can check for proper nesting in the left task pane window.
- Auto-completion of tags and auto-indentation to keep track.

LTERNATIVES TO OXYGEN:

XMLSpy: 30-day free trial at:
http://www.altova.com/products/xmlspy/xml_editor.html

XMetal: 30-day free trial at:
http://na.justsystems.com/content-xmetal

Liquid XML – Community Edition – free; also available is a 30-day trial of their Designer and Developer Editions, free for 30 days:
http://www.liquid-technologies.com/XmlStudio/Liquid-Xml-Studio-Version-Comparison.aspx

Notepad++: Open-source and free, this color-codes text and also features auto-completion and auto-indentation. http://notepad-plus.sourceforge.net/uk/download.php

---

FURTHER RESOURCES:

W3Schools Tutorial on XML, with an XML-Validator and TryItEditor:
http://www.w3schools.com/xml/default.asp; also http://validator.w3.org/

XML Tutorial, complete with up-to-date examples and open-source resources. Clear and easy to understand.
http://www.tizag.com/xmlTutorial/index.php

Nice and simple tutorial on XML Basics, plus information on browser support.
http://xmlfiles.com/xml/

The next two links are associated with an Open-Source initiative, Liquid XML (the Community version is free). The first link gives a thorough overview of XML essentials, while the second gives a clear understanding of DTDs and XSDs and how to validate with them.
http://www.liquid-technologies.com/Tutorials/XML/XmlTutorial.aspx
http://www.liquid-technologies.com/Tutorials/XmlSchemas/XsdTutorial_01.aspx

---