Chapter 6 Supplement

XML-Encoded Metadata

Questions for Review, Study, or Discussion

1. What does it mean for metadata to be *encoded*, and why is it necessary and important?
2. What is *MARC*?
3. What is *XML*? How is it related to *SGML* and *HTML*?
4. What is a *DTD* and an *XML schema*?
5. What *are tags, start tags, end tags, elements, attributes, subelements, XML declarations,* and *root elements* in XML? How must they be formatted in order to be "well-formed?" See pages 150-152 as well as the "Anatomy of an XML Record" on page 160 for answering these questions.
6. What is needed in order for XML documents to be usable and interchanged among different institutions and communities? How is this relevant to metadata for cultural heritage digital collections? What is the difference between *well-formed* and *valid* XML? What is the difference between generic XML and MODS XML?
7. What is a *namespace*? How can namespaces be used to include elements from more than one element set in a single XML file and to disambiguate them?
8. What is the file name extension for an XML schema file and how is it referenced within an XML document?
9. How do metadata creators commonly create metadata in XML?
10. What elements does the *OAI DC XML schema* allow?
11. In what way can Qualified Dublin Core be expressed in a *flat*, rather than a *hierarchical*, XML syntax?
12. How does the Qualified Dublin Core XML example differ from the MODS XML example given on pages 156-159?

Recommended Readings and Resources for Reference or Further Study

- Johnston, Pete. 2006. "Metadata Sharing and XML." In *Good Practice Guide for Developers of Cultural Heritage Web Services*. Bath, UK: UKOLN. Last modified January. http://www.ukoln.ac.uk/interop-focus/gpg/Metadata/.
- Rhyno, Art. 2005. "Introduction to XML." In *Technology for the Rest of Us: A Primer on Computer Technologies for the Low-Tech Librarian*, edited by Nancy Courtney, 71–84.Westport, CT: Libraries Unlimited.
- Yott, Patrick. 2005. "Introduction to XML." In *Metadata: A Cataloger's Primer*, edited by Richard P. Smiraglia, 213–235. Binghamton, New York: Haworth Press.
- Duval, Erik, Wayne Hodgins, Stuart Sutton, and Stuart L. Weibel. 2002. "Metadata Principles and Practicalities." *D-Lib Magazine* 8, no. 4 (April). http://www.dlib.org/dlib/april02/weibel/04weibel.html
- MODS XML Schemas. http://www.loc.gov/standards/mods/mods-schemas.html.
- W3C Extensible Markup Language (XML) website. http://www.w3.org/XML/.

Exercises

Recommended Exercises

1. Take an XML record and identify XML elements, start tags, end tags, attribute names and values, nested subelements, XML declarations, root elements, namespace references, references to an XML Schema (.xsd) file, and the correct structure for each. You might print the XML record and circle and label these aspects.
2. Experiment with using the free Dublin Core Generator at http://www.dublincoregenerator.com. Note both the Simple Generator and the Advanced Generator, and use either or both depending on whether you are creating simple or qualified DC. Enter your metadata in the form, and then select your XML output options. The following are suggested:
   - Display output as: XML.
   - Include standard XML version/encoding declaration.
   - Include root element and namespace.
   - Desired root element: metadata
   - Include namespace reference for standard Dublin Core (DC Elements).
   - Include namespace reference for qualified Dublin Core (DC Terms).
3. Take a set of Dublin Core metadata created in a table format (from the Chapters 3 and 4 exercises if you did them) and put it into an OAI DC XML format, following the example given in Table 5.11 in the book. Remember that only the fifteen simple Dublin Core elements are valid in OA DC XML.
4. Find a MODS XML record like the one in Table 5.12 from the MODS User Guide Full Record Examples available online (http://www.loc.gov/standards/mods/userguide/examples.html or https://wiki.dlib.indiana.edu/download/attachments/24288/DLFMODS_ImplementationGuidelines.pdf).
   a. Copy and paste the complete record into a plain text editor such as Notepad.
   b. Select "Save as" and give it a name with a .xml file extension instead of .txt, and select UTF-8 as the character encoding before saving.
   c. Close the file.
   d. Right click on the file and select "Open With" and select your preferred Internet browser (Firefox, Internet Explorer, or other).
   e. See how the browser is able to display a well-formed XML record using color coding and collapsible/expandable hierarchically nested subelements.
   f. If the file will not open, it probably contains an inadvertent coding error. Try another record example.
5. As a follow-up to the previous exercise, download a free XML editor software package, such as Microsoft's XML Notepad, or alternatively obtain a license to one of the more powerful XML editors such as oXygen or XML Spy, or obtain a free 30-day trial license for one of these products.
   a. Open your MODS XML record from the previous exercise by right clicking on the file and selecting "Open With" and XML Notepad (or oXygen, or whatever other XML editor software you have).
   b. Explore the display and editing features, clicking on expandable plus signs as needed to open element content. experiment with editing it.
   c. Note any validation errors listed at the bottom of the screen or elsewhere, depending on the software and see if you can resolve them.
   d. If the software has a separate Validate or Validation function available from one of the menus, select that and view the results. The file can validate against the MODS XML Schema online because it is formally declared in the root tag in this file.

Suggestions for Instructors

- Provide students with the records needed for the exercises above, or have them use records they previously created themselves. Test the XML records in any editing software beforehand to be sure there are no display or validation errors.