LIS 652: XML, Document Structures, and Metadata

School of Library and Information Studies
University of Wisconsin-Madison
Spring 2014

Dorothea Salo (please call me "Dorothea") — salo@wisc.edu, 608-265-4733
Office address: 4261 Helen C. White Hall — Office Hours: by appointment
Course link page: http://pinboard.in/u:dsalo/t:652 — Skype: dortheasalo

Course Objectives

Upon completion of this course, students will be able to:

- Hand-author well-formed and valid XML documents
- Parse/validate and correct non-well-formed and invalid XML
- Analyze unfamiliar documents to decompose them into XML's hierarchical and pointer structures
- Build a basic document that is valid per an unfamiliar DTD or XML Schema, based on existing documentation and sample documents
- Declare XML namespaces and use namespace prefixes correctly
- Recognize and read a few XML languages common in libraries and archives (e.g. EAD, MODS, TEI)
- Author a well-formed and valid multiple-namespace XML document by hand
- Write a basic XSLT transformation from XML to XHTML
- Read a relatively simple RDF graph
- Read and hand-author acceptable RDF triples in N-triple, Turtle, and RDF/XML formats
- Insert RDFa data correctly into an HTML document
- Recognize and use syntax for common RDF datatypes and notations (e.g. URIs, strings, dates, language)
- Recognize and read a few RDF languages common in libraries and archives (e.g. DC, SKOS)

This course is designed to assess student progress in the following SLIS program-level outcomes: 3a and 3d.

Course Policies

I wish to fully include persons with disabilities in this course. Please let me know within two weeks if you require accommodation. I will try to maintain the confidentiality of this information.

Academic Honesty: I follow the academic standards for cheating and plagiarism set forth by the University of Wisconsin.

Readings and software

I require the purchase of *XML in a Nutshell* (O'Reilly, 3rd edition 2004) by Elliotte Rusty Harold and W. Scott Means, which should be readily available used. (It is available as an ebook through the library, but trust me, you will want to make marginal notes!) Other required readings will be on print or e-reserve, or from the open Web. Several manuals and guides are on print reserve in the SLIS library; if you are stuck on something, they should be your first resort.

I recommend but do not require purchase of the oXygen XML editor, especially if you plan to do homework away from the SLIS computer lab; we will be using it during class sessions. Consult me about free alternatives for your operating system.

Contacting me

For any difficulty with the course that is not private or confidential, please use the Learn@UW help forum; *I will not answer such questions by email*. Please also do your best to assist your classmates on the forum. I am not available weekends; otherwise, I do my level best to answer forum questions and email within two business days. I do not hold regular office hours because they are underused; please feel free to wander by my office or make an appointment to talk with me. If student groups decide to hold regular study sessions, please let me know so that I can try to be available.

Should you see dead links (it does happen, usually with no notice), weird due dates, or other syllabus problems, please post them to the "Syllabus problems" forum on Learn@UW.

Course logistics

Class will meet in the SLIS Computer Lab in Helen C. White Hall on Thursdays from 9 to 11:30AM. Class attendance is mandatory. If you cannot attend class, please notify me by email beforehand, and make arrangements on your own to get notes and assignments from another student. Excessive absences or tardiness may result in grade penalties beyond the normal 5% for attendance and participation.

Please have some type of external storage available to you at class and lab sessions. Cloud storage such as Box, SpiderOak, or Dropbox is acceptable, as are USB (key, thumb, flash) drives. Emailing files to yourself is a last resort only.

LIS 652 will require you to go beyond the basic computer skills needed in other courses. I assume that each student has successfully mastered the following computer skills:

- Be able to install and teach yourself unfamiliar software.
- Be able to use or learn the Linux/OSX command line.
- Be able to save files to and retrieve them from external storage (USB drives, cloud storage, etc.)
- Be able to log in to and use Learn@UW.
- Be able to create, find, and use directories or folders on computer storage media.
- Be familiar with the Microsoft Windows operating system. Knowledge of MacOS and Unix or Linux may also be helpful, but are not necessary in most cases.
- Be able to use a Web browser, perform Internet searches, and save a file to your computer from a hyperlink.
- Be able to save files to a specific location and e-mail files to yourself.
- Understand the difference between a plain-text editor and a word processor.
- Understand filename extensions and how they affect use of a file.

These are the *minimum* skills required to start the course. I will not stop class or lab sessions to explain any of these skills.

Class schedule

This is not set in stone! If particular topics give difficulty, I may contribute extra time to them; if they prove easier than I anticipate, I may move on early. Please complete readings for a given week by the beginning of class that week. (Obviously you have grace for Week 1.)

Unit 1: XML, XML schema languages, and XSLT

Week 1: Course overview. Document analysis. Basic XML syntax.

Learning objectives: Goals and processes of document analysis. Content, structure, and presentation. Hierarchy and OHCO. IDs and pointers. Rules of well-formed XML. Where whitespace does and doesn't matter.

N.b. Don't be thrown off by "SGML" in some of the readings below. XML is SGML! (Though the converse may not be true.)

DeRose, Durand, Mylonas, & Renear. "What is text, really?" `http://dx.doi.org/10.1145/264842.264843`
Harold & Means. Section 1.4, sections 2.1-2.5, 2.7, 2.9. Chapter 6 introduction, sections 6.1, 6.2
Lubas, Jackson & Schneider. *The Metadata Manual.* (Chapter on e-reserve.)
Maler and El Andaloussi. *Developing SGML DTDs: From Text to Model to Markup.* Sections 1.2, 1.3, 4.1.2, 4.2.

Week 2: XML namespaces. XML validation.

*Learning objectives: How XML validation and XML parsing differ. Basic DTD syntax. ?, *, +, |. Parameter entities. DTD limitations. Basic XML Schema syntax. Datatyping in XML Schema. Finding and using DTD and schema documentation and sample XML documents for XML-based markup languages.*

N.b. We will not learn to write DTDs and schemas in this course! Vanishingly few people who work daily with XML ever write a DTD or schema. Your goals: validate your own XML and make a stab at reading DTDs and schemas by others.

Harold & Means. Sections 3.1-3.4, 3.7.
Ray. *Learning XML.* Section 4.3. (Available as an ebook through the library.)

Week 3: More on DTDs, schemas, and namespaces

Learning objectives: Practice with reading DTDs and schemas, working with novel XML-based markup languages. XML namespaces, namespace URIs, namespace prefixes, binding URIs to prefixes.

Harold & Means. Chapter 4 introduction, sections 4.1, 4.2.

Week 4: XSLT 1. XHTML.

Learning objectives: What XHTML is. Why XHTML did not take over the web; analogous failures of XML implementation. What XSLT does. Setting up an XML to XHTML transformation in oXygen. xsl:stylesheet. xsl:output. xsl:template and its match attribute. xsl:text. xsl:apply-templates. xsl:value-of. xsl:copy. xsl:copy-of. Using target markup in an XSLT stylesheet.

Harold & Means. Section 7.1.

Wagner. *XSLT for Dummies*. pp. 14-16, Chapter 2, Chapter 3, Chapter 4. (Available as an ebook through the library.)

Week 5: XSLT 2

Learning objectives: Basic XPath; XPath predicates. Attribute value templates. Dealing with common transformation situations.

N.B. I will be out of town at the Library Technology Conference this week. Class will not meet in person, though the lab is reserved for you at regular class time if you want it. Class lecture and assignments will be available on Learn@UW. Week 4 assignments will have a Learn@UW dropbox open for them, and are due at 9AM class day as usual.

Wagner. *XSLT for Dummies*. Chapter 5, Chapter 6.

Milowski, "There are monsters in my closet; or, how not to use XSLT." `http://www2.sims.berkeley.edu/academics/courses/is290-8/s04/lectures/5/dragons/allslides.html`

Unit 2: XML for documents and data

Week 6: Using established document standards

Learning objectives: Is a metadata record a document? What metadata do/should XML documents contain? JATS. TEI; TEI header. EAD. DocBook. Document standards in libraries and archives. Presenting XML documents on the web. Pretty-printing XML documents; XSL:FO and CSS-based techniques.

Harold & Means. Sections 6.3 (TEI), 6.4 (DocBook), 7.2.1 (XML in browsers).

(Optional) Ray. *Learning XML*. Section 3.2.

Ray. *Learning XML*. Section 5.1 (Stylesheets).

Week 7: Using XML for non-document (meta)data.

Learning objectives: Decomposing a spreadsheet into XML. Decomposing a database into XML. Why sometimes neither of those is a good idea; why it's done anyway. XML datatyping and its limitations. MARCXML. MODS. Dublin Core and (a few of) its many XMLish permutations; OAI-PMH. XML internationalization; the xml:lang attribute.

Harold & Means. Chapter 16.

Kennedy, "Nine questions to guide you in choosing a metadata schema." `https://journals.tdl.org/jodi/article/viewArticle/226/205`

Riley, "Seeing Standards." `http://www.dlib.indiana.edu/~jenlrile/metadatamap/` (Download the poster and read the legend and definitions carefully.)

Examine the MARCXML, HTML, and MODS versions of the record for Carl Sandburg's *Arithmetic*, available from `http://www.loc.gov/standards/marcxml/`, concentrating on the MARC fields familiar to you from LIS 551. Glance at the MARCXML to MODS transformation at `http://www.loc.gov/standards/mods/v3/MARC21slim2MODS3-4.xsl` and see how much of it you can understand (beware: it's lengthy!). Bring questions to class!

Week 8: Cleaning up metadata. Getting non-XML (meta)data into XML. OpenRefine.

Learning objectives: Atomicity, and why a lot of library/archive metadata doesn't have it. The problem with assuming context (including markup nesting). Why strings often confuse computers. Coping with Other People's Metadata. Open Refine (installing, importing data, basic data cleanup).

Dueber. "ISBN parenthetical notes: Bad MARC data #1." `http://robotlibrarian.billdueber.com/isbn-parenthetical-notes-bad-marc-data-1/`

Miller. "How will users manage without finding aids?" In "All text considered: a perspective on mass digitizing and archival processing." *American Archivist* 76:2 (2013) pp. 529-532.

Weeks. "OMG! My metadata is as fresh as the Backstreet Boys!" http://www.slideshare.net/rascalwhale/using-google-refine-long-version

Heller. "A librarian's guide to OpenRefine." http://acrl.ala.org/techconnect/?p=3276

Nguyen. "Using Google Refine to clean messy data." http://www.propublica.org/nerds/item/using-google-refine-for-data-cleaning

SPRING BREAK: enjoy!

Week 9: Using XML for software configuration. Solr.

Learning objectives: Apple .plists. Indexes, indexers, and search engines. Relevance ranking; making relevance decisions. Lucene and Solr. Solr's update schema. Solr's schema.xml. XML in server-software configuration. XML and JSON.

N.b. If we need a catch-up week, this is it!

The Apple Examiner. "PLIST files." http://appleexaminer.com/MacsAndOS/Analysis/PLIST/PLIST.html

Rochkind. "Information retrieval and relevance ranking for librarians." http://bibwild.wordpress.com/2011/03/28/information-retrieval-and-relevance-ranking-for-librarians/

Kainulainen. "Spring data Solr tutorial: introduction to Solr." http://www.petrikainulainen.net/programming/solr/spring-data-solr-tutorial-introduction-to-solr/

"UpdateXmlMessages." https://wiki.apache.org/solr/UpdateXmlMessages

Mikoluk. "JSON vs XML: How JSON is superior to XML." https://www.udemy.com/blog/json-vs-xml/

Unit 3: Linked data and RDF

Week 10: Introduction

Learning objectives: Why integrating documents and data from different sources is hard, but often useful. How computers think about data (in spreadsheets, databases, XML) and how that contributes to integration problems. Why XML does not entirely solve data-integration problems. History of the Semantic Web; transition to and context of the linked-data movement. Five stars of linked data.

Hilton, "Rise of the machines." http://blog.wellcomelibrary.org/2013/12/rise-of-the-machines/

Campbell & MacNeill. "The Semantic Web, Linked and Open Data." http://wiki.cetis.ac.uk/images/1/1a/The_Semantic_Web.pdf

Kelley. "How the W3C has come to love library linked data." http://lj.libraryjournal.com/2011/08/technology/how-the-w3c-has-come-to-love-library-linked-data/

"Library Linked Data Incubator Group Final Report." Sections 2, 3, 4.1, 4.4. http://www.w3.org/2005/Incubator/lld/XGR-lld-20111025/

"5 star Open Data." http://5stardata.info/

Week 11: Identifiers and RDF. Basic N-triples syntax. RDF graphs.

Learning objectives: RDF as data model with many serialization syntaxes. Identifiers, unique identifiers, why strings are lousy identifiers. URIs. Subject, predicate, object. String-literal objects; marking language on literals. N-triples syntax. Reading RDF graphs.

Gonzalez. "RDF vs. XML." http://www.cambridgesemantics.com/semantic-university/rdf-vs-xml

"Falsehoods programmers believe about names." http://www.kalzumeus.com/2010/06/17/falsehoods-programmers-believe-about-names/ (pay special attention to 7, 12-13, 21-22, 37-40!)

Custer & Joyner. "Approaching authority." http://www.slideshare.net/steganogram/approaching-authority-a-preliminary-implementation-of-encoded-archival-context-eaccpf-at-east-carolina-university (Skim this for what they're trying to accomplish and how they did it.)

Gonzalez. "RDF 101." http://www.cambridgesemantics.com/semantic-university/rdf-101

Week 12: More RDF syntaxes: Turtle, RDF/XML, RDFa.

Learning objectives: Turtle abbreviations. RDF/XML syntax. RDFa; embedding RDFa in web pages.

Gutteridge. "What you need to know about RDF+XML." http://blog.soton.ac.uk/webteam/2010/11/08/what-you-need-to-know-about-rdfxml/

Gonzalez. "RDF Nuts & Bolts." http://www.cambridgesemantics.com/semantic-university/rdf-nuts-and-bolts

W3C. "RDFa 1.1 primer." http://www.w3.org/TR/rdfa-primer/

Week 13: Commonly-seen linked-data languages and vocabularies

Learning objectives: SKOS. Dublin Core. FOAF. VIAF. id.loc.gov.

W3C. "SKOS primer." `http://www.w3.org/TR/skos-primer/` (through section 3; ignore sections 4 and 5)

FOAF. "FOAF vocabulary specification." `http://xmlns.com/foaf/spec/#sec-standards` (through "FOAF Auto-Discovery")

Dempsey. "Names and identities: looking at Flann O'Brien." `http://orweblog.oclc.org/archives/002212.html`

Look me (and/or your favorite author) up in `http://viaf.org/` and the name authority search at `http://id.loc.gov`. Check out the available RDF!

Week 14: RDFizing other metadata. SPARQL.

Learning objectives: Transforming XML to linked data. Limitations of such transformations. Reconciliation as a step toward linked data. Reconciliation using OpenRefine. SPARQL.

Coyle. "Linked data first steps & catch-21." `http://kcoyle.blogspot.com/2013/07/linked-data-first-steps-catch-21.html`

Stevenson, "Archives Hub and VIAF Name Matching." `http://archiveshub.ac.uk/blog/2013/08/hub-viaf-namematching/`

Page. "Using Google Refine and taxonomic databases…" `http://iphylo.blogspot.co.uk/2012/02/using-google-refine-and-taxonomic.html`

(Optional; for those who have taken or are taking LIS 751) Prud'hommeaux. "SPARQL vs. SQL: Intro." `http://www.cambridgesemantics.com/semantic-university/sparql-vs-sql-intro`

Week 15: The future's so bright…

Learning objectives: Current library and archive initiatives based on markup and linked data.

Salo. "Linked data in the creases." `http://lj.libraryjournal.com/2013/12/opinion/peer-to-peer-review/linked-data-in-the-creases-peer-to-peer-review/`

Raimond & Ferne. "The BBC World Service archive prototype." `http://challenge.semanticweb.org/2013/submissions/swc2013_submission_5.pdf`

Assignments

Assignments	Percentage	Due Date
Weekly assignments	50%	At class time
In-class practicals	20%	Weeks 3, 4, 8, 13
Well-formed XML résumé	10%	Class time, Week 5
XSLT résumé transformation to XHTML	10%	Class time, Week 8
RDF graph and RDF from résumé	10%	Class time, Week 15

Grading scale: 100-93.5 A; 93.4-89.5 AB; 89.4-83.5 B; 83.4-79.5 BC; 79.4-73.5 C, 69.5-73.4 D, below 69.5 F

Grading policies

I expect and encourage collaboration among students in the course, in class and on homework. Students who find partners to work and study with generally find the assignments easier. However, all assignments will be submitted and graded individually unless otherwise stated.

Weekly assignments

Weekly homework assignments will be announced in class. If no other due date is given, these assignments are due at the beginning of the next class; late assignments will be penalized 10% of available points per day or fraction thereof late. I will allow revision and resubmission at my sole discretion and on my schedule only; any student resistance will remove the opportunity.

In-class practicals

These will take place at the beginning of class (after Q&A). They will consist of problems similar to the homework, to demonstrate that you can do the technical tasks taught in this course on your own. I will cue you in advance what kinds of problems to expect. There will always be bonus points available on a practical, to leave room for the occasional error.

If you cannot attend class on a practical day, you MUST inform me at least 24 hours in advance to arrange for an online re-take. If you do not do so, you lose all points on that practical.

Well-formed XML résumé

Reformat your résumé into well-formed XML. Grading criteria: Does it parse? Do the document analysis and resulting tag structure make sense? Is everything tagged that should be? Are tags structural rather than presentational? Tag abuse will lower your grade. Turning in word-processing-derived XML or XHTML will receive no credit whatever.

XSLT résumé transformation to XHTML

Write an XSLT stylesheet that transforms your well-formed XML résumé into correct and useful XHTML 1.1. (It does not have to be beautiful; you are not required to write CSS, though you may if you like.) Turn in the stylesheet and the original XML résumé (in case you changed it after the prior assignment). Do NOT turn in the XHTML. Grading criteria: Does the stylesheet run? Does the resulting XHTML validate? Is the XHTML correctly namespaced? Is the XHTML tagging sensible, and structurally-oriented within XHTML's limits (nothing but <p>s will lower your grade)?

RDF graph and RDF from résumé

Pull out all the people (references, supervisors, instructors, etc; minimum of 5, add people if you need to), organizations (educational institutions, workplaces; minimum of 3, add to yours if need be), and work products/projects (articles, presentations, e-portfolio, projects, etc; minimum 1, fake it if you need to) mentioned in your résumé. Using at minimum the FOAF and DC vocabularies, write a minimum of 30 triples relating them to one another. Look up and use appropriate URIs for people and organizations wherever possible. You may use whichever RDF serialization you prefer. Draw a graph of the resulting triples, either by hand or with the RDF Distiller (`http://rdf.greggkellogg.net/distiller`) or similar RDF-graph-generation tool.

SLIS Program-level Learning Outcomes	652 Objectives	652 Measurable Outcomes
3a. Students organize and describe print and digital information resources.	Recognize and read a few XML languages common in libraries and archives (e.g. EAD, MODS, TEI) Recognize and read a few RDF languages common in libraries and archives (e.g. RDFS, DC, SKOS)	Weekly assignments will test student ability to recognize and use information in these description languages.
3d. Students understand and use appropriate information technologies.	All objectives.	Weekly assignments designed to familiarize students with XML and linked data. Graded on syntactic correctness, understanding of XML and RDF serializations. In-class practicals measure individual mastery of techniques learned in class. Projects measure student ability to be generative (rather than solely reactive) with the tools and techniques learned in class.