LIS 652: XML and Linked Data

School of Library and Information Studies
University of Wisconsin-Madison
Spring 2016

Dorothea Salo (please call me "Dorothea")
Office address: 4261 Helen C. White Hall
Course link page: http://pinboard.in/u:dsalo/t:652

salo@wisc.edu, 608-265-4733
Office Hours: by appointment

Course Objectives

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

- Insert schema.org microdata correctly into an HTML document or template
- Read a relatively simple RDF graph
- Read and hand-author acceptable RDF triples in N-triple, Turtle, and RDF/XML formats
- Recognize and use syntax for common RDF datatypes and notations (e.g. URIs, strings, dates, language)
- Recognize and read a few RDF vocabularies common in libraries and archives (e.g. DC, SKOS, PCDM)
- Hand-author well-formed and valid XML documents
- Parse/validate and correct non-well-formed and invalid XML
- Build a well-formed and valid multiple-namespace XML document
- Build a basic document that is valid per an unfamiliar DTD or XML Schema, based on existing documentation and sample documents
- Recognize, read, and (limitedly) author a few XML-based document and metadata languages common in libraries, archives, and scholarly publishing (e.g. EAD, MODS, TEI, JATS)
- Read and (limitedly) author SPARQL queries and XSLT stylesheets

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

Course Policies

I intend to fully include persons with disabilities in this course. Please let me know within one week how I can best meet your needs. 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 recommend 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 also available as an ebook through the library, but trust me, you will want to make marginal notes!) Other required readings will be on e-reserve or from the open Web.

I recommend installing the oXygen XML editor; license information is available on Learn@UW. Consult me about free alternatives for your operating system. You will also need to install MARCEdit (`http://marcedit.reeset.net/`) and LODRefine (`https://github.com/sparkica/LODRefine`), both of which are free and cross-platform.

Contacting me

For any difficulty with the course that is not private or confidential, please ask in class or use the Learn@UW help forum; *I will not answer such questions by email*. Please also do your best to assist your classmates. I am not available weekends; otherwise, I do my level best to answer forum questions and email within two business days. 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.

Known schedule disruptions

Expect me to be slower to respond than usual during the following dates:

- March 20-22: North Carolina Serials Conference

Prerequisite skills

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

- Hand-author basic HTML (what you learned in the Tech Gateway or LIS 551 or 644 should be sufficient)
- Be able to save files to and retrieve them from external storage (USB drives, cloud storage, etc.)
- Be able to create, find, and use directories/folders.
- Be able to make and unzip zip files.
- Be able to use a Web browser, perform Internet searches, and save a file to your computer from a hyperlink.
- Know the difference between http:// and file:// URLs, and what that means for whether someone other than you can access a given file.
- 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. If any of the above confuses you, you are expected to work it out on your own *immediately*.

Unit 1: Linked data and RDF

Week 1: Context. RDF use in information agencies. XML use in information agencies.

Learning objectives: Why integrating documents and data from different sources is hard, but often useful. How computers represent data (in spreadsheets, databases, XML) and how that contributes to integration problems. History of the Semantic Web; transition to and context of the linked-data movement. Five stars of linked data. XML as a document technology.

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/`

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/`

Week 2: Identifiers and RDF. Microdata in HTML.

Learning objectives: Identifiers, unique identifiers. Why strings are lousy identifiers, and what that means for library/archive authority control. URIs, URLs, URNs. Identifier/URI correlation; owl:sameas and similar constructs. Identifier sources. How search engines use schema.org microdata. Microdata syntaxes; using them in HTML. Using microdata in templated web applications (CMSes, ILSes, digital libraries and archives, repositories).

Linklist: http://pinboard.in/u:dsalo/t:identifiers, http://pinboard.in/u:dsalo/t:schemaorg

"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!)

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!

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.)

Ronallo. "HTML5 microdata and schema.org." *code4lib journal*. `http://journal.code4lib.org/articles/6400` (You may stop when you reach the tutorial, but keep this article in mind for examples that will help you do your homework.)

Coyle. "Schema.org: where it works." `http://kcoyle.blogspot.com/2014/10/schemaorg-where-it-works.html`

Hellman. "Spoonfeeding library data to search engines." `http://go-to-hellman.blogspot.com/2011/07/spoonfeeding-library-data-to-search-engines.html`

Arnold. "Improving search engine traffic to DIMES." `http://rockarch.org/programs/digital/bitsandbytes/?p=826`

Week 3: RDF graphs. N-triples.

Learning objectives: RDF as data model with many serialization syntaxes. Microdata as RDF. Subject, predicate, object. Reading RDF graphs. String-literal objects; marking language on literals. N-triples syntax. Basic RDF via the Yarn Game.

"Introducing Graph Data." `http://www.linkeddatatools.com/introducing-rdf`

Gonzalez. "RDF 101." `http://web.archive.org/web/20140327205008/http://www.cambridgesemantics.com/semantic-university/rdf-101` (Don't try to use the version on the Cambridge Semantics website; all the images have dropped out of it, and it's useless without them.)

Gonzalez. "RDF Nuts & Bolts." `http://web.archive.org/web/20131030053957/http://www.cambridgesemantics.com/semantic-university/rdf-nuts-and-bolts`

Week 4: Turtle. Inference engines; classes; domain/range. Common RDF vocabularies. Library-specific RDF vocabularies.

Learning objectives: Turtle abbreviations. RDF classes. Domain and range in RDF. Inference. SKOS. Dublin Core. FOAF. BIBFRAME. BL model. Portland Common Data Model.

Coyle. "Classes in RDF." `http://kcoyle.blogspot.com/2014/11/classes-in-rdf.html`

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")

"Portland Common Data Model." `https://github.com/duraspace/pcdm/wiki` (just the front page)

Week 5: RDFizing other metadata.

Learning objectives: Atomicity, and why a lot of library/archive metadata doesn't have it. Transforming metadata in other formats to linked data. Limitations of such transformations (especially with respect to string values). Reconciliation as a step toward linked data. The open-world assumption, and other potential RDF pitfalls.

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

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/`

Schilling. "Transforming library metadata into linked library data." `http://www.ala.org/alcts/resources/org/cat/research/linked-data`

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`

Weeks 6 and 7: SPARQL.

(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`

Lincoln. "SPARQL for humanists." `http://matthewlincoln.net/2014/07/10/sparql-for-humanists.html`

"Querying semantic data." `http://www.linkeddatatools.com/querying-semantic-data`

Unit 2: XML

Week 8: Basic XML syntax. XML namespaces. RDF/XML.

Learning objectives: Rules of well-formed XML. Troubleshooting well-formedness errors. RDF/XML syntax.

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 2 on e-reserve.)

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

"Introducing RDF/XML." `http://www.linkeddatatools.com/introducing-rdf-part-2`

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/`

Week 9: XML validation, DTDs, and XML Schema

Learning objectives: How XML validation and XML parsing differ. What DTDs and XML Schemas are. How to find and use DTD/schema documentation. Data typing in XML Schema. Troubleshooting validation errors.

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.

Harold & Means. Sections 3.1-3.4, 3.7, Chapter 4 introduction, sections 4.1, 4.2.

Ray. *Learning XML*. Section 4.3. (Available as an ebook through the library.)

Unit 3: XML for documents and metadata

Week 10: 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.

Harold & Means. Sections 6.3 (TEI), 6.4 (DocBook).

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

Ghetu. "EAD Instruction Manual." `http://www.dlib.indiana.edu/services/metadata/activities/EADManual.pdf` (pp. 4-6)

Beck. "NISO Z39.96 The Journal Article Tag Suite (JATS): What happened to the NLM DTDs?" `http://dx.doi.org/10.3998/3336451.0014.106`

Week 11: Using XML for non-document (meta)data.

Learning objectives: XML datatyping and its limitations. MARCXML. MODS. Dublin Core and (a few of) its many XMLish permutations; OAI-PMH.

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.

Week 12: Cleaning up metadata. Getting non-XML (meta)data into XML.

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. 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 13-14: XSLT transformations

Lapeyre and Usdin. "Introduction to XSLT Concepts." `http://www.mulberrytech.com/papers/Intro2XSLT/Intro2XSLT.pdf` (through slide 28 on page 16)

Tizag.com. "XSLT - introduction" `http://www.tizag.com/xmlTutorial/xslttutorial.php` (go through all seven pages)

Week 15: Safety week

(I am assuming I've put too much work in at least one existing class week. I reserve the right to fix this by pushing deadlines back! This week intentionally left blank to permit that to happen.)

Assignments

NO ASSIGNMENTS MAY BE SUBMITTED AS WORD-PROCESSING DOCUMENTS OR PDFS unless this is specifically stated in the assignment description. Assignments submitted in this fashion will *automatically receive zeroes*. HTML, RDF, and XML are all plain-text formats; install and use a text editor (sometimes called a "programmer's editor") or a specialized editor such as oXygen/ (for XML) to work with them.

I expect and encourage collaboration among students in this course on major projects as well as weekly assignments (see below). Students who work and study with partners generally find the assignments easier. However, all homework assignments will be submitted and graded individually unless otherwise stated.

Assignments	Percentage	Due date
Valid HTML5 résumé	10%	Week 3 (February 1)
HTML résumé with added microdata	10%	Week 5 (February 15)
RDF graph and RDF from résumé	10%	Week 8 (March 7)
Well-formed XML résumé	10%	Week 11 (April 4)
Weekly assignments	60%	(throughout semester; 4% each week)

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

Weekly assignments

Weekly assignments are part of each week's Learn@UW content. If no other due date is given, these assignments are due Monday of the next class week at 5 pm Central Time; late assignments will be penalized one point per day or fraction thereof late. (You *don't want to get behind* in this class. You really, really don't.) If for some strange reason a particular week has no assignments, I will assign its points to all students automatically.

Many weekly assignments will be of the form "Do—Check—Fix." Once you DO the work, turn in your first (draft) version to the weekly dropbox. Next, CHECK your work, sometimes via a test to pass (as with XML validation or linked-data crosswalking), sometimes by running your work past a class partner (I will assign partners at the beginning of the course). FIX any errors you find, then turn in the corrected version to the weekly dropbox. I will only look at the draft version if I suspect academic-honesty issues.

N.b. Learn@UW dropboxes do weird things to HTML and XML files if you try to view them in-browser. (You'll get the correct original file if you right-click and download it.) If this bugs you, it's fine to zip the file and upload the .zip.

HTML5 résumé

Reformat your résumé into HTML5. (You may add CSS to make your résumé look more attractive, but this is *not required*; your HTML5 résumé need not visually resemble your printed one.) Minimally, the HTML markup must use heading tags and (as appropriate) list and paragraph tags, and it must use them consistently; any other tags used must be sensible in context. Do not use HTML tables just to align non-tabular material visually in the browser, please. You will lose three points on this assignment if your HTML does not validate via `https://validator.w3.org/`.

HTML5 résumé with microdata

Next, for each person (self, reference, supervisor, etc.) and organization (school/college, workplace, professional organization, etc.) in your résumé, add Person or Organization microdata from the schema.org schemas. All available person/organization information in your résumé that can be marked up with microdata should be, but you are not required to add information just to mark it up with microdata. You should use subclasses of Person and Organization and their properties as appropriate; at minimum, educational organizations should be given an appropriate subclass of Organization (there are several!). You may add HTML tags to create scope, if you need to.

RDF graph and RDF from résumé

Pull out the people (references, supervisors, instructors, etc; minimum of 3, add people if you need to), organizations (educational institutions, workplaces; minimum of 2, add to yours if you need to), and work products/projects (books, articles, presentations, e-portfolio, projects, etc; minimum 1, fake it if you need to) mentioned in your résumé OR involving people mentioned in your résumé. Using at minimum three different linked-data vocabularies (that is, you should need at least three `@prefix` declarations if using Turtle) as sources of subject/property/value URIs, create a single RDF graph (that is, containing no triples completely isolated from the larger graph) containing a minimum of 30 triples. Look up and use appropriate URIs for people and organizations wherever possible. You may use whichever RDF serialization you prefer. Draw and turn in a graph of the resulting triples, either by hand or with the W3C's graph/validator tool (`http://www.w3.org/RDF/Validator/`), RDF Distiller (`http://rdf.greggkellogg.net/distiller`) or similar RDF-graph-generation tool.

Well-formed XML résumé

Reformat your résumé into well-formed XML according to a document model you invent. 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 a zero.

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, JATS) Recognize and read a few RDF languages common in libraries and archives (e.g. RDFS, DC, SKOS)	Weekly assignments 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. Projects measure student ability to be generative (rather than solely reactive) with the tools and techniques learned in class.