Learning Linked Data Project
http://lld.ischool.uw.edu/wp/
Final Version, August 29, 2012

Final Inventory of Learning Topics

This document, developed by the IMLS-funded Learning Linked Data Project [1], presents an inventory of topics to be covered by learners who want to understand, process, and create Linked Data. (Definitions of key concepts highlighted in the Inventory may be found in a separate Glossary. [2]) The initial draft was extensively discussed and edited at a February 2012 workshop, posted on a blog for feedback [3] through June, and modified to its present form by the end of the year-long project in September.

The Inventory associates key learning topics with software tools and processing methods needed by learners, whether for coursework or self-instruction, in university or professional training contexts. The project partners intend to propose a follow-on project in 2013 for creating tutorial materials on how to use tools to perform specific analytical or data-processing tasks under each of the headings in this Inventory. The ultimate goal is to develop a body of screencast-based "microtutorials," linked to the Inventory headings, that can be easily updated as new tools become available. This documentation would help independent learners and students, as well as faculty members and instructors who need to follow a rapidly changing landscape of tools and methods for incorporation into their own teaching. The project will also aim at creating a community forum where instructors can formulate new analytical tasks and trade experience on how these tasks fit into particular instructional contexts.

Because Linked Data is based on linguistic data structures, the guiding metaphor for the project has been that of a "language lab" equipped with tools for mastering the learning topics. By design, the Inventory is neutral about the background and goals of the learners to be supported. The Inventory is not a syllabus for a particular course, nor does it reflect particular curricular or pedagogical approaches. Rather, the Inventory is seen as the basis for a structured "palette" of tools and methods from which learners and instructors can draw for a variety of courses of study. To use a cooking metaphor, the project aspires to outfit a kitchen with utensils usable for preparing a wide range of meals.

While analogies to natural language or to cooking may help to illustrate project goals and boundaries, the project uses the native terminology of RDF for the Inventory. Where that terminology is foreign to learners, the project expects that instructors will bridge any conceptual gaps in ways suited to their own pedagogical approaches.

Prerequisite: An Understanding of Linked Data

Linked Data is data that can fit into a "cloud" of interconnected data sources, either published world-readably on the Web (Linked Open Data) or behind corporate or institutional firewalls (Linked Enterprise Data). More specifically, for the purposes of this inventory, Linked Data is data published in a form compatible with the Resource Description Framework (RDF) model, a

Semantic Web standard of the World Wide Web Consortium (W3C). Inasmuch as RDF is a language designed for processing by machines, learners must acquire competence in the use of software tools for ingesting, visualizing, transforming, and interpreting its contents. Prerequisite to using any sort of tool, however, a learner must have some knowledge of the following basics (see the Glossary for definitions):

- **Technologies related to Linked Data**, such as Unix and other server environments, XML, HTML, and databases.
- **The use of Uniform Resource Identifiers (URIs)** as globally unique identifiers and as the "links" of Linked Data.
- **The RDF data model**: the Open World Assumption, the structure of statements (properties, classes, and instances in RDF triples), ontologies, RDF vocabularies, and the like.
- **The node-arc model of RDF graphs**, and how triples are linked into graphs.
- **New developments in RDF 1.1** (under development), such as Named Graphs.
- **Basic principles of inferencing** (reasoning).
- **Principles of publishing Linked Data**, such as "following one's nose" to multiple representations on the basis of content negotiation.

A growing number of Web and courseware resources cover such fundamentals, such as the EUCLID curriculum, university courses such as those at Cornell and Indiana, guidelines for publishing Linked Data and Linked Data patterns, and YouTube videos.

Searching and Querying Datasets

Learners of natural language must first learn "how to learn" by asking questions of native speakers. By analogy, learners of Linked Data must learn how to query datasets, explore their characteristics, assess their validity, and explore their underlying vocabularies. The project group sees three sub-topics under this general heading:

- **Formulating structured queries** involves new search engines, such as Sindice and SWSE, which are starting to focus specifically on Linked Data, sometimes specializing on search within a specific field, such as the Chem2Bio2RDF exploration tool for chemical biology. Search methods are often based on the SPARQL Protocol and RDF Query Language (SPARQL). Support for SPARQL is built into many software development frameworks, such as Jena. A growing ecosystem of tools and services aims at helping users learn SPARQL and use it effectively, such as the Gruff Structured Query Tool, SPARQLer query engine, and the Pubby SPARQL front end. Alternatives to SPARQL, such as the English-based Internet Business Logic from Reengineering LLC, are an area of ongoing research.
- **Assessing data and checking consistency** requires mastery of basic tools for checking the logical consistency of RDF and whether it is well-formed, starting with simple validation tools such as W3C's RDF Validator or tools for validating RDF markup in Web pages such as Structured Data Linter.

- **Discovering vocabularies** involves exploring the rich landscape of RDF vocabularies and datasets using resources like the Linked Open Data Cloud, Data Hub, Open Metadata Registry, and Linked Open Vocabularies (LOV).

Creating and Manipulating RDF Data

In the Resource Description Framework (RDF), "everything is data" from descriptions of things — both of things in the world or, more specifically, of information resources — to descriptions of the languages of description used to describe those things. Instances, attribute spaces, and value spaces are all expressed as RDF data. Therefore this category encompasses a broad range of topics which, in other fields, might be considered quite separate from each other.

- **Creating RDF vocabularies** and minting URIs for properties and classes (with RDF vocabulary editors such as Neologism), ontologies (with editors such as Protégé), or SKOS concept schemes (with tools such as PoolParty).
- **Mapping between vocabularies** is universally recognized as a key problem for which mature tools are lacking.
- **Specifying a domain model** enumerating the things to be described in a data, for example using diagramming tools based on Unified Modeling Language (UML) or mind maps.
- **Converting RDF data among alternative RDF syntaxes** such as RDF/XML, Turtle, N-Triples, and RDFa, or from Schema.org's microdata, using conversion programs such as Rapper or software libraries, such as RDF.rb (for Ruby) or RDFLib (for Python).
- **Generating RDF triples from non-RDF sources** such as Calais, which creates triples from the content analysis of unstructured text data; numerous specific tools for converting from specific application formats; GRDDL, a mechanism for triplifying XML data according to conversion rules; "distillers" for extracting RDF triples from Web pages marked up with RDFa or microdata; or data cleaners, such as Google Refine, a "power tool for working with messy data."
- **Creating datasets as Linked Data** as described in the section on "Implementing Linked Data Applications" below.

Visualizing Webs of Data

Visualization plays a unique role in understanding RDF data because RDF graphs are conceptually diagrammatic in nature. In RDF, "everything is data," so some of the tools usable for visualizing instance data may be used to visualize ontologies. Other tools may be used to explore the statistical, spatial, or temporal characteristics of datasets:

- **Visualizing RDF graphs** with software such as RDF validators for generating node-and-arc diagrams of RDF graphs or the D3 tool for generating graphs following a wide range of visual styles.
- **Generating a Linked Data cloud** with software such as the CKAN software of the Open Knowledge Foundation.

- **Analyzing statistical characteristics of large datasets** with visualization tools such as Spotfire.
- **Viewing data from different perspectives** with data browsers such as Gruff, or plotting data to timelines or maps (as with Simile Widgets).

Implementing Linked Data Applications

Simply learning how to interpret and process Linked Data could stop with the topics outlined above. In order to deploy Linked Data applications, however, a learner should become at least familiar with the implementation options:

- **For publishing RDF-compatible data on the Web:** Web frameworks (such as Ruby on Rails) and Content Management Systems (such as Drupal).
- **For storing RDF data:** RDF triple stores (such as Virtuoso) and RDF-compatible relational databases.
- **Integrated tool platforms** such as the open-source LOD2 stack of key tools for processing Linked Data and the VIVO platform for building interlinked networks of researchers and their research results.

References

[1] http://lld.ischool.uw.edu/wp/learning/about/
[2] http://lld.ischool.uw.edu/wp/glossary/
[3] http://lld.ischool.uw.edu/wp/