Designing and Implementing an Ontology for LLE Experimental Diagnostics

Robert W. Cooper, III
Allendale Columbia School
Advisor: Rick Kidder
Summer 2010

Abstract:

The OMEGA Laser experimental diagnostics operations were researched to discover how knowledge of the operation and set up of these systems is managed. The goal of the project was to see if a technology existed that provided mechanisms to capture this knowledge into an artificial intelligence environment, making it more accessible to the many users of the experimental lab. Semantic web technologies were investigated for this purpose and to determine what platform was best for designing an ontology (a computerized representation of knowledge based on relationships between data) for LLE experimental diagnostics. The research determined that open source editors like ProtégéOWL were available that simplified designing the ontology, and Web Ontology Language Descriptive Language (OWL DL), a semantic web standard, was the most useful syntax for the project. Once the basic design was complete, the ontology was developed into a prototype application using Java NetBeans Integrated Development Environment (IDE). The combination of this work provided a centralized repository for the extraction of useful information about the modeled diagnostics. The wide range of complex relationships were mapped out in the ontology allowing users to search for extremely specific result sets, eliminating the need to manually parse documents. The application also provided results windows, which were programmed to provide the user with links to other, potentially useful information. The application has also been equipped with data entry windows to allow users to increase the ontology's scope. The simplicity of the NetBeans interface was

demonstrated and makes enhancements of the code relatively easy. As a result, the ontology can be further developed to capture virtually every aspect of LLE's laser systems.

Introduction:

The current systems

LLE has accumulated an enormous amount of useful knowledge about its laser controls, imaging and diagnostic systems over the course of its existence. Unfortunately, LLE's knowledge repositories are dispersed over several disjoined systems with no common thread in between. This type of structure requires a variety of inefficient methods to access the wealth of information, which wastes users' time and often prevents the most useful information from being discovered and utilized. LLE has several repositories of information: the Product Data Management (PDM) system, the Oracle database, operation documents called Volumes and individuals who work at LLE. Individuals maintain knowledge on various networked and non-networked hard drives, notebooks and in their heads. Data retrieval is dependent on the source. Accessing data in the PDM system consists of a keyword search, in which any documents containing words or phrases that match the query are returned. The user is then required to manually parse the documents for the piece of desired information. The Oracle database contains all shot-related data from the OMEGA and OMEGA EP laser systems and must similarly be queried by keyword and parsed manually. Finally, the human-based resources are even more difficult to query, with data filed on personal computers, hidden among stacks of paperwork, and simply present in the minds of individual researchers. Accessing this information is difficult, as general users may not know the data even exist, may not understand methods to retrieve the data, and may not even know the many locations or types of data.

An overview of the Semantic Web

A semantic web is a basic form of artificial intelligence in which a machine is able to interpret data as knowledge. This process is made possible by using machine-readable data called metadata (data about data) as tags. These metadata tags describe relationships between other data, and contain important information about the tagged data. As the computer interprets these relationships, it is able to make inferences on the actual information represented by the data. The three main types of relationships, or "properties," found in ontologies can be seen in Figures 1, 2, and 3. Functional properties, illustrated in Figure 1, require that an individual can be related to only one other individual. Therefore, functional properties enable machines to determine whether two or more entities with different identifiers are, in fact, the same individual. This ability would greatly enhance the organization of a knowledge database, as confusion resulting from an individual with more than one name would be avoided. Transitive properties (Figure 2) help to illustrate pathways. By linking a linear progression of individuals, these properties enable machines to navigate through linear hierarchical relationships using "shortcuts," as transitive properties allow reasoners to jump straight to the final point instead of parsing each

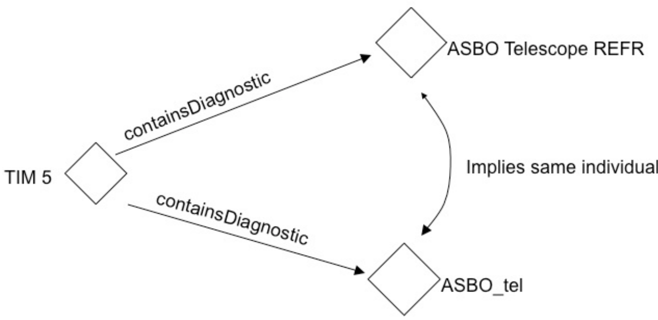


Figure 1: Graphical representation of a functional property*. Also known as a "single valued property," a functional property for a given individual stipulates that only one other individual can be related via this property. In the above case,* ASBO Telescope REFR *and* ASBO_tel *must be the same individual because* containsDiagnostic *is functional.*

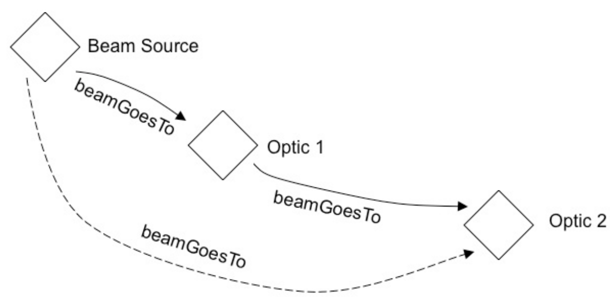


Figure 2: A graphical representation of a transitive property. If a transitive property relates individual 1 to 2 and 2 to 3, then individual 1 is also related to individual 3 by the same property.

intermediate step individually. Finally, symmetric properties (Figure 3) will make data entry a much more efficient process. These properties will enable machines to automatically fill in information on one individual when a user makes changes to its symmetrically related partner.

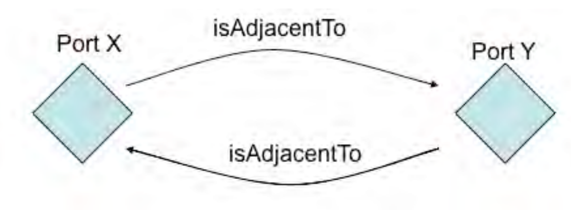


Figure 3: A graphical representation of a symmetric property. A symmetric property implies that if individual 1 is related to 2, then 2 is related to 1 by the same property.

A functioning, trusted semantic web is made up of several parts, illustrated in the "Semantic Web Stack" in Figure 4. The bottom, red layer makes up the semantic web's basic essentials – a character set and identifier. Moving up into the horizontal orange blocks are the machine-readable parts of the semantic web. These enable documents to be interpreted and interacted with by computers. The grey section, where this project was focused, is the framework of the ontology and the language and syntax used to create it. To the right and upwards of this grey area are more "optional" features of a semantic web, including the level of interaction by users, security measures, and logic reasoners.

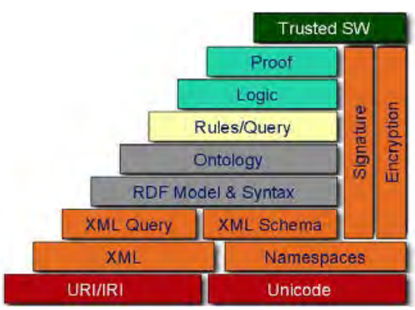


Figure 4: The "Semantic Stack."

Ontologies at LLE

Implementing a semantic web at LLE would greatly increase the efficiency of the process of searching for information in the current database in several ways. First, the data would be concentrated in one location. Second, instead of performing a generic keyword search and parsing documents, specific information would be returned along with links to other, potentially useful items. This specificity would most likely be the greatest benefit of an ontology, as poring through hundreds of documents for a particular line or figure would no longer occur.

Consequently, over the past few years developing semantic web technologies have been researched at LLE [1-3]. As these technologies are still relatively new, much of the preliminary research revolved around exploring how a semantic web works and is constructed. An ontology editor, Protégé, was discovered [2], and a prototype model of a basic semantic web was constructed [3]. Although these projects were based more on researching a developing technology than creating and implementing an ontology itself, they made for a very solid foundation for this project.

Research and Development:

Understand the Diagnostics

In order to virtually model an experimental diagnostic, one must fully understand the physical object first. As LLE possesses a wide range of complex systems, finding a method to categorize them is similarly complex. Due to the enormous quantity of diagnostics systems at LLE, fixed diagnostics for the OMEGA laser system were focused on. Nine characterizations were created to describe a sampling of 33 of the fixed OMEGA diagnostics. Each of these diagnostics could fall into any three of the nine characterizations. These categories were titled "optical," "x-rays," "neutrons," "charged particles," "electromagnetic radiation," "time integrated," "time resolved," "spectrally resolved," and "spatially resolved." Each of these characterizations describes an aspect of each diagnostic, and is intended to direct the user's attention to possible uses for the diagnostic. As such, each of these categories can be used to model metadata tags in an ontology. Further down the application's lifecycle, they will be refined into quicksearch categories and utilized to make the relationships between virtual models of diagnostics far more detailed.

Pick a technology

Returning to Figure 4, this project resided mostly in the grey section of the chart: the Resource Description Framework (RDF) data model and the ontology itself. Protégé, the editor chosen to construct LLE's experimental diagnostics ontology, is compatible with a variety of syntaxes, schema, and metadata sets. Protégé allows the user to construct an ontology in "straight RDF," RDF Schema (RDFS), and the Web Ontology Language (OWL). Both RDFS and OWL use RDF as a base standard, and elements of RDFS are used in OWL. As a result, OWL has a less limited vocabulary than RDFS, which in turn is less limited than straight RDF. OWL was therefore determined to be the ideal language with which to construct the ontology.

OWL has three distinct sublanguages: OWL Lite, OWL DL (named after its correspondence with descriptive logic), and OWL Full. The main difference between these three sublanguages is the level of expressiveness available to the user. OWL Lite is the least expressive of the three, and only supports primitive constraints. OWL DL supports reasoning software, and is as expressive as possible while still allowing the software to properly function. OWL Full is the most expressive of the three, but it is based on a slightly different framework from the other two and is too expressive to support complete reasoning. OWL DL was consequently determined to be the most useful sublanguage to design the ontology, as its logic reasoner would enable quick and efficient sorting and searching and its expressiveness would support the relatively complex classes required for the experimental diagnostic database.

Finally, a metadata set was needed to tag the ontology's data. ProtégéOWL provided a relatively basic RDFS metadata vocabulary, but its scope was rather limited. Consequently, a variety of other metadata sets were explored, including Dublin Core, Friend of a Friend (FOAF), Semantically-Interlinked Online Communities (SIOC), and Simple Knowledge Organization

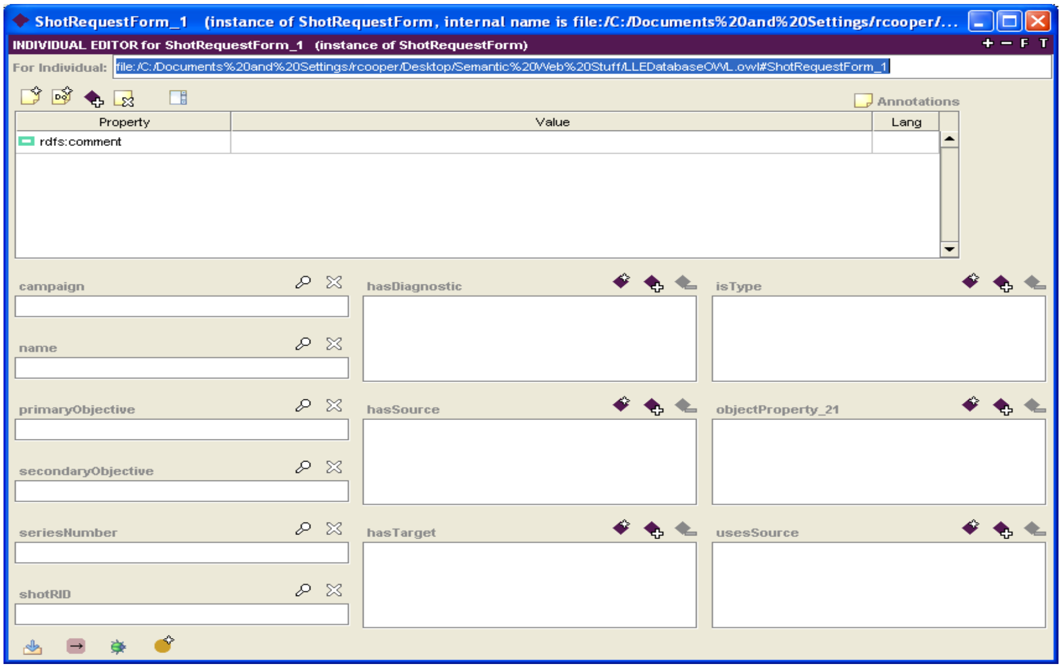
System (SKOS). After researching the structures of each vocabulary, it was determined that Dublin Core would be the most suitable metadata set. FOAF and SIOC are designed to describe relationships between people and social networking, respectively, and SKOS is generally used for dictionary or thesaurus-type structures. Dublin Core, on the other hand, is perfect for describing and annotating objects, which would enable the experimental diagnostic ontology to have an increased degree of specificity.

The ontology designed with ProtégéOWL would also have to be programmed into an application. There are two methods of creating such an application: exporting the ontology from ProtégéOWL directly to a Java format, or programming from scratch using the ontology as a model. Exporting directly from ProtégéOWL returns an interface for each object in the ontology, which was not desirable for the application's needs. Therefore, NetBeans Integrated Development Environment (IDE) was used to design and program several user interfaces to implement the ontology.

Protégé Ontology

Classes for Shot Request Form (SRF), Diagnostic, Source, Target, and Experiment Type were created. Each class was given properties that corresponded with the data requested on each SRF in the PDM database. Properties such as hasTarget, hasDiagnostic, and hasSource in the SRF class were all made as symmetric properties with isTargetForShot, isDiagnosticForShot, and isSourceForShot, respectively. The editor window for an SRF individual is shown in Figure 5. The six text-entry boxes in the lower left portion of the screenshot each represent a functional property of the SRF, and serve as data labels for the individual. The six remaining boxes represent properties that link the SRF to other individuals. Individuals listed in these boxes are clickable links to similar windows for their respective classes. The Diagnostic class was given symmetric properties for past experiments and experiments each diagnostic is designed for. Similar symmetric properties were given to the Source and Target classes, in order to link them

Figure 5: Screenshot of Individual Editor in ProtégéOWL.



to the other three. These specified relationships allowed ProtégéOWL's Pellet reasoner to automatically link individuals. Consequently, when a user adds an individual to the Protégé hierarchy, the reasoning software instantly places it in its correct place, and enables the user to quickly navigate between the individual and other related ones.

<u>NetBeans Application</u>

Windows for Shot Request Forms (SRF), Diagnostic, and Experiment Type were designed with both user-input and readonly options. The user-input windows are designed to allow database managers to easily add to and edit the existing items in the database. Code was written to mimic the behavior of Protégé's Pellet reasoner, such that relationships between objects could be quickly determined and new individuals could be linked to existing ones in the database immediately upon entry. The readonly windows corresponded directly to the layouts of the user-entry windows, without the ability to edit the contents. An example of a results window for the Diagnostic class is shown in Figure 6. Elements in each window that were based off any of the symmetric properties outlined in Protégé were made clickable to enable the user to quickly navigate between related elements.

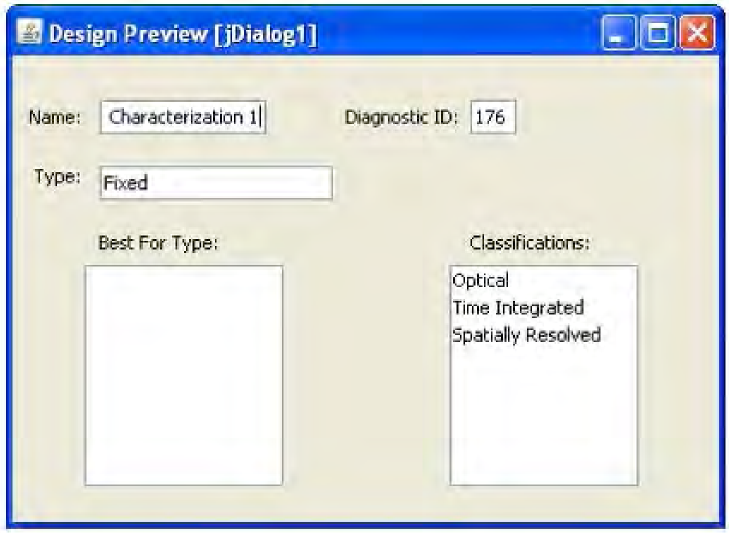


Figure 6: A search result window. In this case, the user either searched by diagnostic name or ID.

<u>The Project's Future</u>

The application is still in its prototype form, so there are still some minor problems to be worked out. Once the final version is completed, it can be exported and distributed to as many users as necessary. Developers should be able to easily link the distributed applications to one, central database, so that the ontology may be accessed from a variety of locations. Also, NetBeans and Protégé are user-friendly applications, so editing the code and the model for the ontology itself can be easily done by an administrator. Consequently, the ontology can be expanded to include any and all documents pertaining to LLE. In fact, the future ontology could conceivably form a virtual representation of all of LLE's experimental systems, making research and shot-planning a considerably more efficient and useful process.

Acknowledgements:

I would like to thank Rick Kidder and Stephen Craxton for giving me the opportunity to spend my summer at LLE. I would especially like to thank Mr. Kidder for his advice, support, and guidance throughout my project. I would like to thank Andrew Zeller for his frequent assistance and accessibility. I would also like to thank Eugene Kowaluk and Dino Mastrosimone for their suggestions on the structure and purpose of the ontology. Finally, I would like to thank Ricky Marron, Dan Gresh, and Chris Baldwin for the excellent foundation that they created in semantic web research.

References:

1. D. Gresh, “Implementing a Knowledge Database for Scientific Control Systems,” 2006 Summer High School Research Program at the University of Rochester's Laboratory for Laser Energetics, LLE Report No. 348.
2. R. Marron, “Development of an Ontology for the OMEGA EP Laser System,” 2007 Summer High School Research Program at the University of Rochester's Laboratory for Laser Energetics, LLE Report No. 353.
3. C. Baldwin, “Exploring Metadata for Laser Diagnostics and Control Systems on the OMEGA EP Laser System,” 2008 Summer High School Research Program at the University of Rochester’s Laboratory for Laser Energetics, LLE Report No. 357.