Exploring Metadata for Laser Diagnostics and Control Systems on the OMEGA EP Laser System

Chris Baldwin
Honeoye Falls-Lima High School
Advisor: Mr. Rick Kidder
Summer 2008

Abstract

For this project, an application was made that illustrates the advantages of an ontology over the current databases in use at LLE, which tend to be ad-hoc and poorly documented. An ontology is a way of organizing the data of a system based on the data's characteristics (ie: the metadata). By connecting the data via their properties, the ontology creates a model of the system, and thus derives its strength. This project focuses on the gain that an ontology provides by allowing programs to simulate "cause and effect", something that would be much harder to do under the current organization of data. The application implements an ontology of the OMEGA EP laser system and uses "causes and effect" to predict the effects on laser diagnostics timing as changes are made to the system. The program allows the user to make typical changes, such as switching between long pulse and short pulse, as well as to add/remove subsystems. The timings on the diagnostics are affected accordingly. All of this is done without any reference to the actual parts of the laser system, and thus the program functions the same regardless of how the ontology is configured. Ontologies can have wide-ranging applications at LLE, but the important thing to note is that more metadata needs to be readily available for ontologies to have much practical value.

1. Introduction

1.1 Background

Since its start, LLE has amassed an enormous amount of data. Currently, this data is stored in two locations: a PDM (Project Data Management) file system and several Oracle databases. The PDM file system indexes files like Word documents, Powerpoint presentations, and several other formats into sections based on their topics. Accessing the files is a matter of navigating through PDM sections and subsections, although if one knows the desired file's code one can enter it directly into a search bar. The Oracle database, on the other hand, is a collection of tables that contains the individual pieces of data. Many other tables and data are located on specific workstations.

Naturally, locating specific files and data can be quite challenging, as can be making use of the information. Database users must know exactly what it is they are looking for, and must be aware of where and how to look for it. One way to simplify the process would be to implement an ontology of the OMEGA and EP laser systems. By storing the files and data through an ontology, computers could perform the most tedious aspects of not only retrieving information but also using it.

1.2 Ontologies

Essentially, an ontology is a model of a system, a model that gains its strength through classifying and linking its members. Each member of an ontology, also known as an "individual", belongs to at least one "class", which is a type of member that is known to exhibit a specific set of characteristics. For example (see Figure 1), Bill, an individual, is a type of person, a class. This is the classifying aspect of ontologies. As for linking the members, every individual has "properties" that point to other individuals. The properties serve to illustrate how the individuals interact with each other. For instance, Bill's favorite color is blue. "Favorite color is" is a property Bill has, one that in this example points to the color blue.

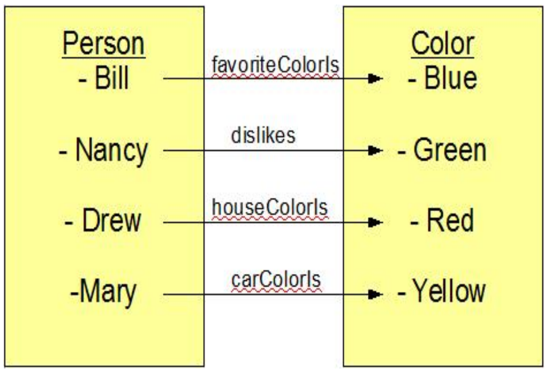


Figure 1: A sample ontology. The boxes represent classes, the examples within individuals, and the arrows properties.

Just like tables and even text files, ontologies store data. Yet ontologies don't store statistics and readings; they store characteristics. Thus the main advantage to an ontology is that it shows what subjects are related, and how they are connected. Most importantly, this is done in a manner that computers can easily understand. With ontologies, computers are able to make many of the connections and inferences that people do.

1.3 Purpose of the Project

In past years of the LLE high school program, work has been done on implementing an ontology for data retrieval.[1,2] Yet an ontology has the potential to do much more than organize LLE's collection of data. A model of the laser systems and subsystems would simplify virtually every computer-based task, and this project is intended to demonstrate that.

In a system as complex as LLE, making one change to the laser systems will have many effects down the line. This makes seemingly straightforward adjustments quite complicated, for many changes will need to be made as opposed to one. Furthermore, when an employee wishes to make these changes many times, the time requirement becomes much larger. However, an ontology would allow for programs to be designed that simplify the entire process. Computers could recognize the numerous changes that go hand-in-hand and perform them on a case-by-case scenario, adapting to subtle changes in the conditions.

2. The Project

2.1 Final Product

At LLE, different types of shots require different configurations of the laser systems, and these different configurations have a variety of effects. The application produced through this project focuses on one specific “cause and effect”: the effect that modifying the OMEGA EP laser's beam path will have on the separate timing of each diagnostic instrument, i.e., when the beam will reach each instrument with respect to a “default” time.

The application is divided into two sections, an ontology and a program implementing the ontology. The ontology illustrates the layout of the OMEGA EP laser system. Furthermore, it marks the diagnostic instruments in the system and gives each one a numerical property representing a delay in the time at which a portion of the beam reaches it. The program allows users to adjust the laser system in a variety of ways, such as by changing the beam's final destination from the sidelighter to the backlighter or by doubling the number of times the beam passes through the main amplifier. All these changes affect when the beam reaches each diagnostic. The program recognizes these changes in delay and adjusts the instruments' delay settings accordingly.

2.2 OMEGA EP Layout

Before an application simulating the OMEGA EP beamlines can be developed, it is necessary to have a basic understanding of the layout. The EP laser facility houses four separate beamlines, all of which are capable of firing in long-pulse mode, and two of which are capable of firing in short-pulse mode. The ontology created for this project focuses on beamlines 1 and 2, which can fire both long and short pulses.

There are four main sections to each beamline, indicated in Figure 2. The beam originates in Laser Sources. It is then injected into the amplifier, which is

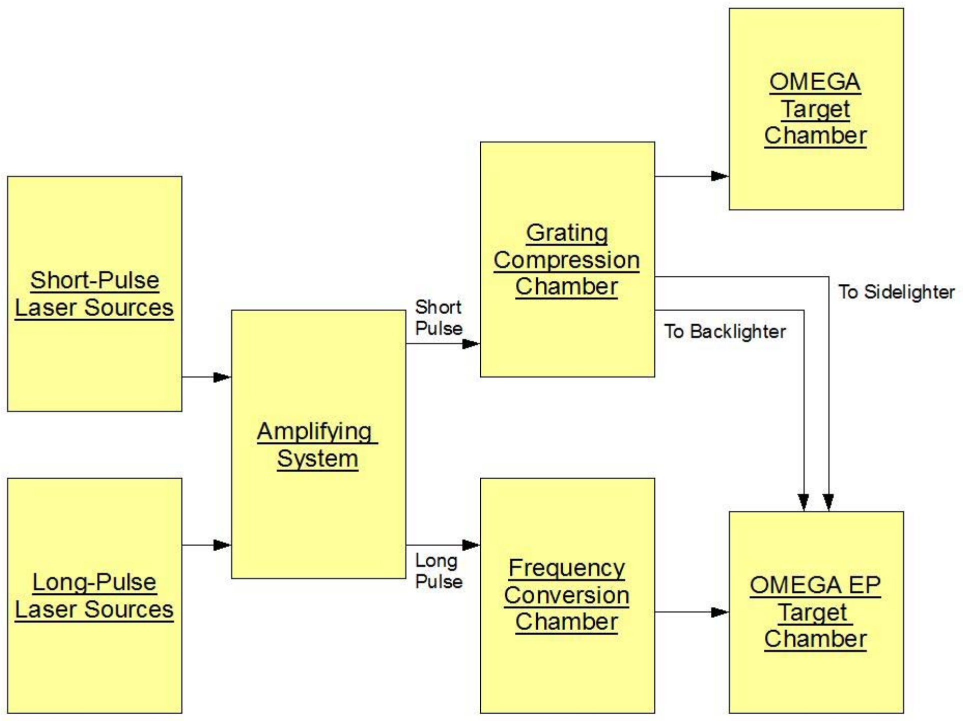


Figure 2: A basic diagram applicable to each of the first two EP beamlines.

divided into an upper level and a lower level. After amplification, the beam can take one of two paths, depending on whether it is intended to be long-pulse or short-pulse. If long-pulse, the beam heads into the Frequency Conversion Chamber, where its frequency is tripled into the UV spectrum. The beam then enters the EP target chamber. If short-pulse, the beam enters either the upper or lower Grating Compression Chamber (GCC), from which it can pass into the EP target chamber through the backlighter port or through the sidelighter port, or head into the OMEGA target chamber.

The EP laser system contains three main diagnostic tables, as well as a series of instruments in Laser Sources. After amplification, a portion of the beam can enter the Infrared Diagnostics Table. If long-pulse, part of the beam is directed to the Ultraviolet Diagnostics Table following frequency-conversion. If short-pulse, the beam passes by the Short Pulse Diagnostics Table after the GCC.

2.3 Ontology

The ontology has been designed through OWL (Web Ontology Language),

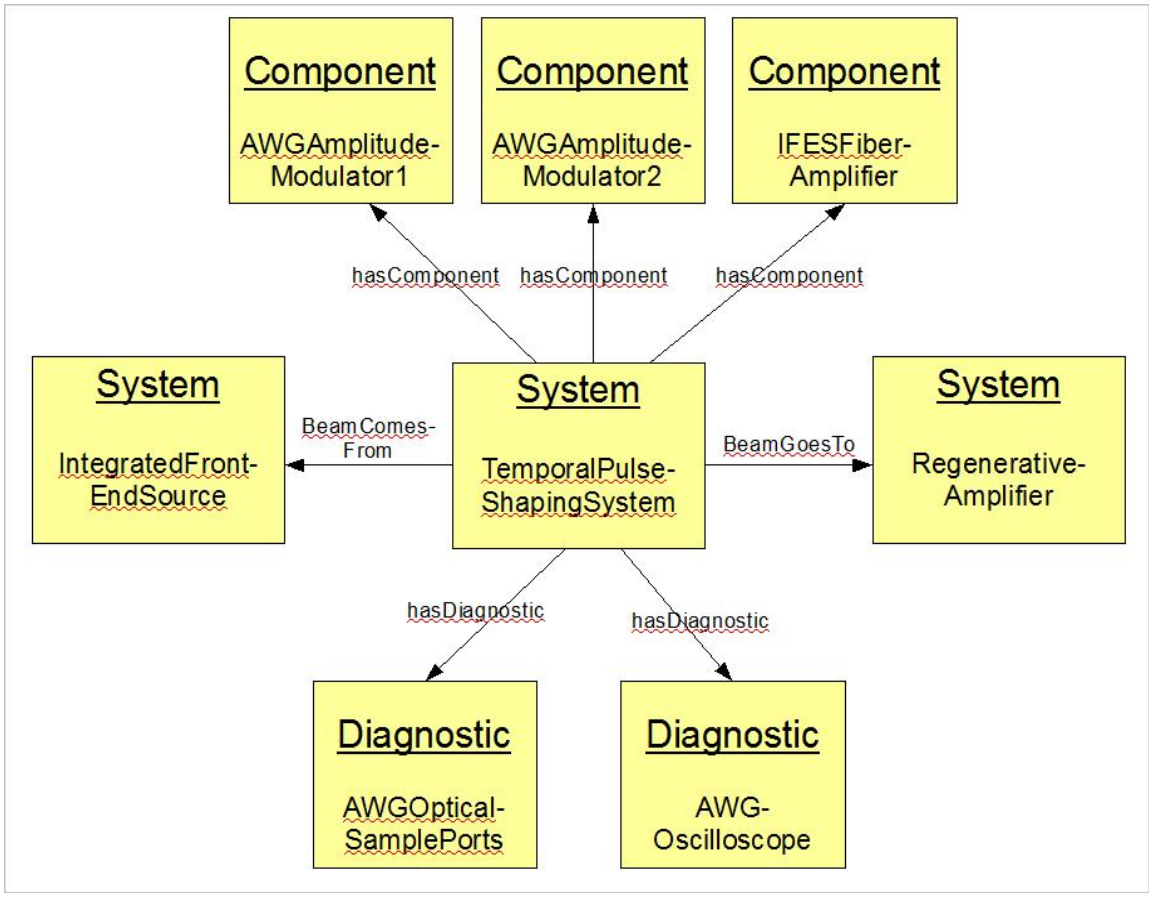


Figure 3: A portion of the ontology, showing the properties of the system, "TemporalPulseShapingSystem".

a modeling language built on XML (eXtensive Modeling Language). Thus, it follows the typical format of clearly defined classes, individuals, and properties. Its key component is the class labeled “Systems”. Each main stage of the EP laser's beam path, such as the temporal pulse shaping system, is an individual in this class. The properties of the temporal pulse shaping system are shown in Figure 3. The systems are linked to each other through the “beamGoesTo” and “beamComesFrom” properties. This creates a “linked list” data structure, in which one block of information points to another block, and only by traversing from one block to the next can information be accessed. Thus tasks such as displaying the path that the beam follows are simply matters of processing the information as it is reached.

The diagnostic instruments are represented in the ontology through a separate class, “Diagnostics”. Every individual in this class has a property pointing to a numerical value. The value represents the instrument's delay, and is modified every time an appropriate change is made to the ontology. As between two systems, diagnostics are linked to the systems on which they are found through the “hasDiagnostic” and “isDiagnosticOf” properties.

Furthermore, the ontology contains classes representing the different paths the beam could follow. Examples are traveling to the backlighter port by way of the upper GCC and traveling to the sidelighter port by way of the lower GCC. Each of these classes contains the individual systems that the beam passes through under that configuration.

2.4 Application

The implementation of the ontology, the program, has been coded with the Java Programming Language, and as such must be run through the Java Virtual Machine. Furthermore, the code makes use of a

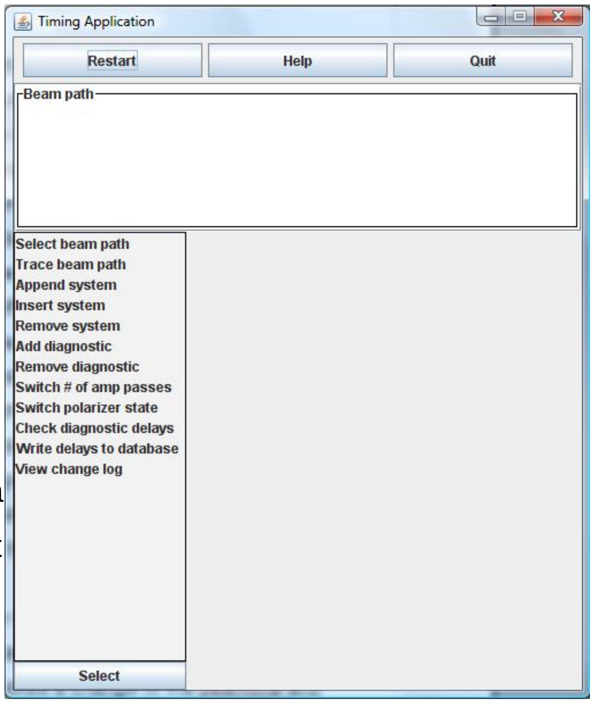


Figure 4: The program window, immediately after the program is initialized.

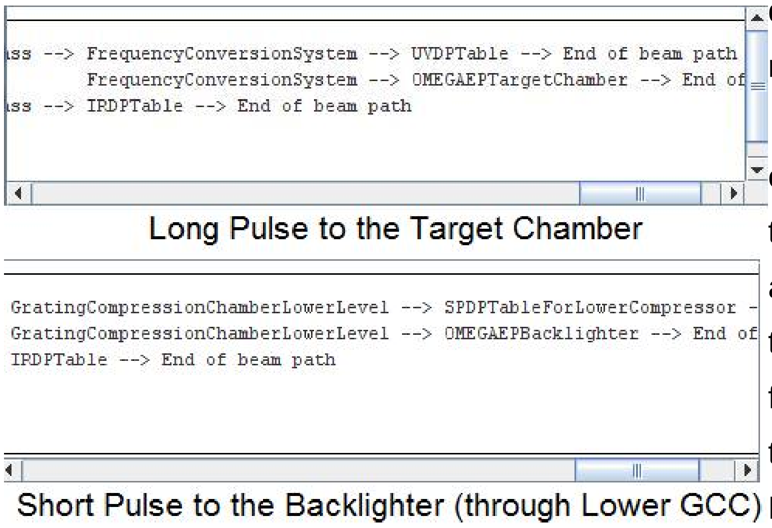
separate library, designed for programming with ontologies, known as Jena. When run, a window is opened with three main sections (see Figure 4). The first simply contains buttons allowing the user to restart, quit, and display a short set of directions on how to use the application. The second is the menu for manipulating the EP laser ontology. It allows the user to: select and display the beam's path, modify individual systems along the current path, add and remove diagnostics, switch the number of amplifier passes between two and four, switch whether the polarizer is inserted into the amplifying system or not, check the delays on the diagnostics, write the delays to a database, and view a list of what has been done in the current session. The third section is the pair of display panels. The upper panel is solely for displaying the current beam path. The large panel in the bottom-right is for everything else, from selecting where to insert a new system into the beam path to displaying the change log.

The goal of the application is to allow users to make a change to the ontology and to have the computer “understand” the change and take the appropriate actions. Specifically, the user makes a change to the beamline and the computer is to identify what diagnostic instruments' timings are affected and adjust the delays of those instruments. The organization of the ontology allows the computer's tasks to be quite manageable. When a change is made to the beamline, the laser must be fired sooner or later so that the beam arrives at its destination at the original time, “T-0”. This means that the diagnostic instruments affected by a change are those located in front of it relative to the beam's path. Since the systems in the ontology are organized sequentially, finding the appropriate diagnostics is simple. The computer starts at the first system after the change and, moving forward through the list, creates a collection of the diagnostics on the systems it comes across. The computer then removes every diagnostic in this collection from a separate collection containing every diagnostic in the ontology, removing all diagnostics not found on the current beam path as well. Thus all that remains is the diagnostics that are affected. The computer then accesses and rewrites the property of each diagnostic in this collection representing its delay (positive values for delays indicate that the diagnostic is to be fired later than normal, and negative values indicate sooner).

2.5 Timing

There are four separate actions the user can take that affect timing: switching the number of amplifier passes, inserting/removing the polarizer, inserting/removing systems, and changing the beam path. The first two are straightforward. Changing the number of amplifier passes from two to four will require diagnostics before the amplifying system to be triggered sooner, as will inserting the polarizer. Similarly, changing from four passes to two and removing the polarizer will each shorten the length of time in which the beam is passing through the beamline, and necessitate a positive delay.

When a system is inserted into the ontology, it naturally adds length to the beamline. Thus all diagnostics found before the new system must be set to fire earlier. Likewise, removing a system shortens the beamline and requires that all diagnostics before the removal fire later.



```
ss --> FrequencyConversionSystem --> UVDPTable --> End of beam path
       FrequencyConversionSystem --> OMEGAEPTargetChamber --> End of
ss --> IRDPTable --> End of beam path
```

Long Pulse to the Target Chamber

```
GratingCompressionChamberLowerLevel --> SPDPTableForLowerCompressor -
GratingCompressionChamberLowerLevel --> OMEGAEPBacklighter --> End of
IRDPTable --> End of beam path
```

Short Pulse to the Backlighter (through Lower GCC)

Figure 5: A comparison of the final stages of a long pulse mode vs a short pulse mode.

As for the last action, changing the beam path: in the EP laser system, there are six main variations in the beam's path. One is firing in long-pulse mode to the EP target chamber (see Figure 5). One is in short-pulse mode directed to the sidelighter port, by way of the lower Grating Compression Chamber. Two are in short-pulse mode heading to the backlighter port, one through the upper GCC and one through the lower GCC (see Figure 5). Likewise, two are in short-pulse mode heading to the OMEGA target chamber, one through each GCC. The long-pulse mode is 106.95 nanoseconds quicker than the short-pulse mode. Traveling to the sidelighter port is 17.18 ns quicker than to the backlighter port, which is itself 92.1 ns quicker than heading to the

OMEGA target chamber. Finally, passing through the lower GCC is quicker than passing through the upper GCC by 7.24 ns. Adjusting delays for a change in beam path is a matter of separately modifying the diagnostics affected by each component (ex: short-pulse, upper compressor, backlighter).

3. Analysis

3.1 Cause and Effect

Ontologies provide two key benefits to programming, both of which originate from their form of organization. The first is that computers can behave based on the idea of “cause and effect”, that one change will require many changes down the line. The program created effectively demonstrates this concept, although simplistically. A user makes a change to part of the ontology, the beamline, and the computer uses information from the change to adjust another aspect, the timings on certain diagnostics. By grouping the members of the ontology into classes, the computer can recognize that changes in systems imply changes in diagnostics.

3.2 Adaptability

The second benefit of implementing ontologies for programming is adaptability, and this is also demonstrated in the program. When a user makes a change to the beamline, the computer receives information on the change, such as where in the beamline the change was made. The computer makes use of this information to decide what to do next. For example, when a new system is inserted in between the main amplifying system and the Frequency Conversion Chamber, the computer determines what diagnostics are found before the new system and adjusts their delays. When a new system is inserted right after the integrated front end source, the computer again locates and alters the appropriate diagnostics. In both scenarios, the computer performs the same task, yet on different diagnostics in each case. The program will do this for any change, regardless of specifics. It functions under any situation involving a change in the beamline, without excessive hard-coding of data.

3.3 Future Development

There is great potential in the implementation of an ontology at LLE. This project is only one small example of the convenience ontologies could bring. A fully designed ontology would be a great step towards the development of “smart” computers. LLE employees could enter large, general commands into a computer, such as “Switch from long-pulse mode to short-pulse mode”, and the computer could perform all the necessary steps.

Currently, however, the creation of such an ontology is limited. Ontologies are dependent on metadata, i.e., data about data. In the case of ontologies, metadata is the information that individuals' properties point to. The creation of an ontology requires that the metadata is easily accessible and understandable, and if it isn't, the ontology suffers. One major barrier in the development of this project was that it was often unclear as to what delay setting belonged to what diagnostic. The information was there, just not understandable. Thus when the ontology was attempting to create its individuals representing diagnostics, it many times couldn't give the individuals real values for their delays. Before ontologies can truly become part of LLE, the data they require must be made clear and concise.

4. Acknowledgements

I would like to thank Richard Kidder and Stephen Craxton for giving me the opportunity to work at LLE and on this project. I would also like to thank Albert Consentino, Liz Hill, and Jason Puth for their help with the timing aspect of the project; Thomas Klingenberger for his help with the LLE databases; and Dan Gresh and Ricky Marron for their work with ontologies before me.

5. 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.