What Comes First, the OWL or the Bean?

Creating Reusable Scientific Software with OWL/RDF Vocabularies.

Eric G Stephan, Todd O Elsethagen, Kerstin Kleese van Dam, Laura Riihimaki
(eric.stephan, todd.elsethagen, kerstin.kleesevandam, laura.riihimaki)@pnnl.gov

Introduction

In the spring of 2013 the U.S. White House by executive order mandated: “Government information shall be managed as an asset throughout its life cycle to promote interoperability and openness, and, wherever possible and legally permissible, to ensure that data are released to the public in ways that make the data easy to find, accessible, and usable.” [1] Key for the reusability of any scientific data is hereby the availability of metadata describing the published data in a vocabulary that is familiar to its potential users. The objective of this paper is to help scientific application developers who want to adopt the continuous stream of new community vocabularies [2][3] to help make their data sharable, self-describable, and easily understood. To achieve this we suggest semantic vocabulary and application integration best practices and discuss the tradeoffs of encoding vocabularies through code versus deriving code from vocabularies.

Dilemma

A “chicken and egg” dilemma occurs when software developers need to decide if they should first incorporate a new vocabulary directly as code (e.g. Java Bean), or derive code from a defined vocabulary (E.g. OWL). Semantic web programmers may naturally gravitate toward producing semantic statements from code representing a vocabulary, while knowledge engineers may gravitate toward developing the model external to any code. Scientific teams wanting to adopt vocabularies might ask questions such as: “What is the most cost effective approach for my project?” “What approach is the easiest to maintain?” or “Which approach will make my software reusable in similar applications or other domains?” The semantic community provides technologies to support encoding the vocabulary in software or converting vocabularies into code. Open source Jena API [4], OWL API [5], and Alibaba API [6] all offer approaches for users to write encode vocabularies Java. At the same time Protégé and Alibaba among other tools offer the ability to turn OWL and RDF vocabularies into Java code. By encoding a vocabulary purely as code, the vocabulary schemata risks being invisible to users wanting to query and reason over data generated by the scientific application. By converting externally defined document-based vocabularies such as thesauri and schemata the code risks either being illegible because the code and vocabulary logic are intermixed or it cannot be realistically maintained as a scientific application APIs.

In the following we will highlight through a real live example the challenges and necessary tradeoffs encountered when choosing to encode vocabularies through code versus deriving code from vocabularies.

Use Case: Provenance based explanation generation for atmospheric science plots through the PNNL Provenance Environment (ProvEn)

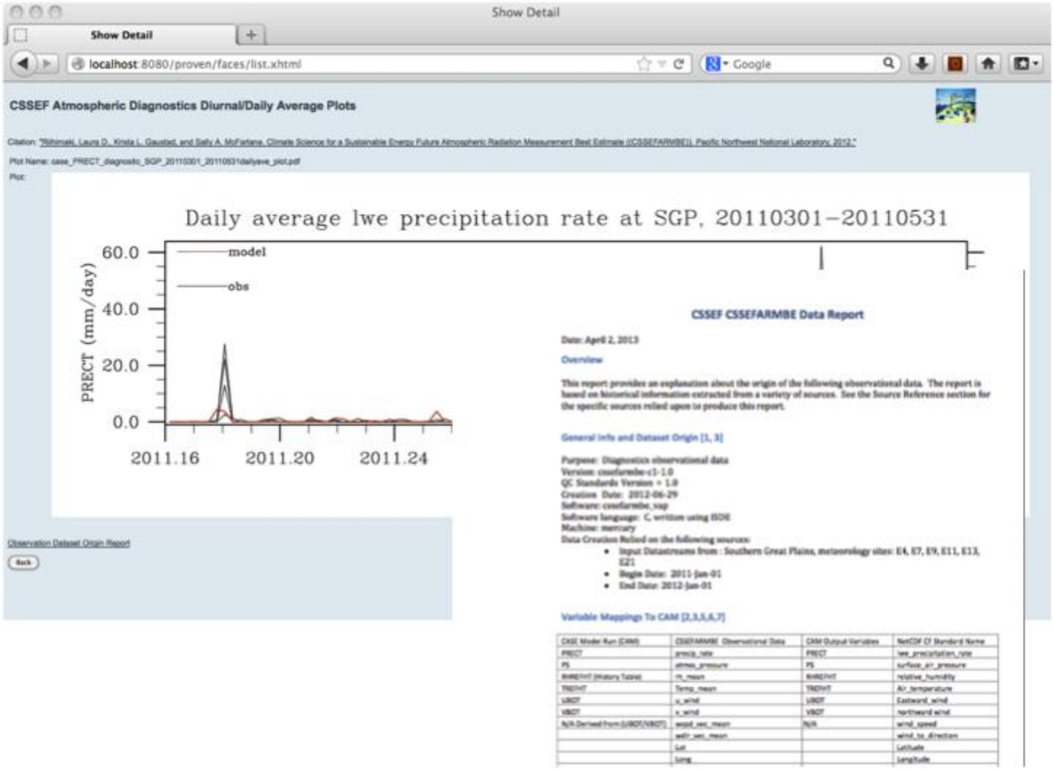
The goal of the ProvEn Services environment was to provide users of the atmospheric climate diagnostic plotting tool developed for the DOE Climate Science for a Sustainable Energy Future (CSSEF) [7] with information about the origin, credibility, and precision of the diagnostics data sets they were using. ProvEn Services organizes the plots, information sources, and semantic knowledge using its own application layer with a Java-based RESTful interface and application vocabulary. ProvEn Services relies on a Sesame triple store and maps Java objects through an Object Triple Mapping (OTM) layer provided by Alibaba. ProvEn Services work by ingesting two dimensional daily average and diurnal diagnostics plots that were generated comparing the Community Atmosphere Model (CAM) [8] against observational data. These plots are referenced as linked data IRI within ProvEn. Original sources of provenance information (e.g. NetCDF [9] headers, plotting scripts, test scripts, NetCDF Climate and

Forecast (CF) Metadata Convention (CF) vocabulary [10] and CAM variable definitions) are also retained as linked data IRI so that users can later inspect for themselves sources in their native representation. Semantic information is extracted from the original sources of provenance and added as reference to the corresponding plots. Users ingesting semantic information will need to use an existing domain vocabulary or generate their own. ProvEn Services is most useful when domain vocabularies are manually aligned to foundation vocabularies to support cross-referenced searches related to provenance, citations, collaborators, organizations etc.

Vocabulary	Type	Purpose	Description
PROV-O [13]	Foundation	Provenance	W3C provenance ontology recommendation
Dublin Core [14]	Foundation	Citation	Standardized way to cite sources
FOAF [15]	Foundation	Collaboration	Social networking vocabulary
ORG [16]	Foundation	Organization	Government Linked Data Group
CSSEF Climate	Domain	Atmosphere	CSSEF adopted or defined terms
ProvEn Services	Application	Internal Model	Used to manage the process of collecting, managing, and partitioning origin information

Table 1: Inventory of vocabularies used for CSSEF Atmosphere Diagnostics Testbed demonstration.

A browsable web interface was added to ProvEn Services that serves climate scientists diagnostics plots. By selecting citation or origin reports ProvEn provides users contextual information about the diagnostics datasets through searches which use the foundation, domain, and application vocabularies. The heart of this provenance tool is a flexible data base structure (knowledge store) that allows storage of varying sources and formats of metadata without breaking, and allows easy translation between the vocabularies of different user groups.


Show Detail
Show Detail
localhost:8080/proven/faces/list.xhtml
Google
Daily average lwe precipitation rate at SGP, 20110301–20110531
60.0
40.0
20.0
0.0
PRECT (mm/day)
model
obs
2011.16
2011.20
2011.24
Observation Dataset Origin Report
Back

CSSEF Atmospheric Diagnostics Diurnal/Daily Average Plots

Citation: "Riihimaki, Laura D., Krista L. Gaustad, and Sally A. McFarlane. Climate Science for a Sustainable Energy Future Atmospheric Radiation Measurement Best Estimate (CSSEFARMBE). Pacific Northwest National Laboratory, 2012."

Plot Name: case_PRECT_diagnostic_SGP_20110301_20110531dailyave_plot.pdf

Plot:

CSSEF CSSEFARMBE Data Report

Date: April 2, 2013

Overview

This report provides an explanation about the origin of the following observational data. The report is based on historical information extracted from a variety of sources. See the Source Reference section for the specific sources relied upon to produce this report.

General Info and Dataset Origin [1, 3]

Purpose: Diagnostics observational data
Version: cssefarmbe-c1-1.0
QC Standards Version = 1.0
Creation Date: 2012-06-29
Software: cssefarmbe_vap
Software language: C, written using ISDE
Machine: mercury
Data Creation Relied on the following sources:

- Input Datastreams from : Southern Great Plains, meteorology sites: E4, E7, E9, E11, E13, E21
- Begin Date: 2011-Jan-01
- End Date: 2012-Jan-01

Variable Mappings To CAM [2,3,5,6,7]

CASE Model Run (CAM)	CSSEFARMBE Observational Data	CAM Output Variables	NetCDF CF Standard Name
PRECT	precip_rate	PRECT	lwe_precipitation_rate
PS	atmos_pressure	PS	surface_air_pressure
RHREFHT (History Tables)	rh_mean	RHREFHT	relative_humidity
TREFHT	Temp_mean	TREFHT	Air_temperature
UBOT	u_wind	UBOT	Eastward_wind
VBOT	v_wind	VBOT	northward wind
N/A Derived from (UBOT/VBOT)	wspd_vec_mean	N/A	wind_speed
	wdir_vec_mean		wind_to_direction
	Lat		Latitude
	Long		Longitude

Figure 2: Snapshot of Atmosphere Diagnostics Testbed plots being shown in ProvEn Services along with an origin report generated by ProvEn.

There are two primary benefits to incorporating ProvEn Services into the diagnostics package. First, ProvEn Services makes it easier for new users of the tool to accurately interpret the figures in the diagnostics package if they are unfamiliar with the underlying observational data sets or are not an expert CAM modeler. Users are provided linked data products where the plots are linked back to semantic descriptions of the diagnostics citations, model run, and plot building process. Second, ProvEn Services has the potential to make the analysis of a large number of UQ model runs more convenient by quickly associating a diagnostic plot with a given set of parameters. In the future we are planning to roll out ProvEn Services to support provenance for the land modeling community and as part of the Earth System Grid Federation (ESGF) [12].

Bean First, OWL Second Approach

This approach is usually motivated by the need to evolve an application specific vocabulary. In the case of ProvEn Services prototype the application's vocabulary, was evolving as quickly as the application, development of the vocabulary from the code base synchronized the vocabulary maintenance with the services being developed. For long-term maintenance ProvEn Services may stick with this approach because the object oriented model is driving the vocabulary. For our use case, Sesame provided the means through the Alibaba interface to directly relate Java interfaces, classes, methods, and variables to OWL classes and properties, and RDF triples. At runtime, Java code interacting with a Sesame server is capable of persisting semantics that can be accessible to other applications. Associations between Java and the semantic vocabulary are made by annotating the java code with @IRI, @SubClassOf, and @SubPropertyOf. This can provide an extremely intuitive approach for the Java developer who requires an internal model for application-based ontology. The Atmosphere Diagnostics Testbed initially used the ProvEn Services internal application vocabulary encoded as Java beans and Property files. The advantage of this approach was that from a Java developer perspective it was possible to write an application vocabulary for ProvEn Services and have objects persisted in the triple store. The disadvantage of this approach is that any code embedded vocabulary cannot be easily inspected by collaborating knowledge engineers, and makes it clumsy for users of data if they stumbled across subtle changes in the resulting from code changes, it also makes the prospect of search and discovery more difficult because we had to manually derive a application vocabulary as an OWL document to determine how community and domain vocabularies would be aligned to the ProvEn Services vocabulary.

OWL First, Bean Second Approach

This approach is motivated by either the need to incorporate an existing RDF/OWL vocabulary into an existing application, or the need to develop a domain specific vocabulary prior to coding. Attempting to code an existing vocabulary is potentially error prone and difficult to maintain. Protégé [11] and Sesame among other off the shelf products provide the means to create Java interfaces for RDFS and OWL classes. For our use case, Alibaba provided an OWL compiler to automatically generate annotated Java interfaces; we extended this capability to automatically implement interfaces as well. There are several advantages to this approach: 1) by predefining the vocabulary software development teams can develop CRUD and search interfaces in parallel, 2) if results are shared in the future the vocabulary can be used as documentation for future users, and 3) as mentioned earlier, many vocabularies are being created and shared, by adopting an existing standard, which can mean that a vocabulary has been vetted and might have more of a stable track record if already being used by a community. Converting an existing vocabulary into code reduces the burden of API maintenance. The disadvantage of this approach is that the development team locks themselves into a particular semantic API. Jena, OWL API, and Alibaba all offer different approaches and do not provide the ubiquitous interfaces that the RDBMS community provides through ODBC and JDBC.

Lessons Learned and Conclusions

Developing scientific applications that rely on semantics can be potentially difficult to maintain without proper planning or design. We recommend where possible for scientific applications to adopt existing vocabularies developed and vetted by other communities. Active vocabulary communities also provide suggested practices to support search and discovery. The most cost effective strategy for software development will depend on the expertise of the teams developing scientific applications. Because of the availability of tools provide a means to either produce vocabularies from code or convert existing vocabularies to code, some nice options are available.

Given our team's software developer background encoding the ProvEn Services application vocabulary seemed to be the most efficient and intuitive way to prototype a ProvEn Services vocabulary along with the Java RESTful services. In retrospect, we found some initial mistakes mapping Java classes to OWL classes, and we could have overcome such errors by prototyping parts of the vocabulary in OWL and then letting Alibaba show us the way the APIs want our application to link its object model to OWL.

From a knowledge engineer perspective the rationale for intuitive prototyping might not seem as straightforward, however by only correlating the application vocabulary to our Java code, the ProvEn Services interface is light and extensible. Alignments between the application vocabulary, domain vocabulary, and foundation vocabularies are made prior to loading them into the ProvEn Services, and it is viewed as more of an administrator function. When possible, aligning vocabularies by making assertions in OWL/RDF is far cheaper than attempting maintain this through code. In the future we want to extend the ProvEn Services interface by adding existing vocabulary APIs such as the foundation vocabularies mentioned earlier. Our approach of using Alibaba to automate the creation of Java interfaces for each OWL class and property, as well as automating the process of implementing interfaces has made code creation and maintenance very cost effective. This point is significant when considering the cost of maintaining synchronized version control between code and vocabularies.

As the example use case of the ProvEn Services demonstrated, the adoption of existing vocabularies reduces the cost of maintaining data origin information for projects such as CSSEF, because other research communities provide most of the vocabularies required. By utilizing community vocabularies ProvEn Services can leverage available concepts to represent origin, citation, collaboration, and organization. Using community vocabularies also means that knowledge is more easily transferrable and exchanged, thus making it more cost effective to support the Whitehouse Executive Order by making the research results more accessible to future users by providing critical provenance information about the results in community accepted vocabularies.

Acknowledgements

This research was supported by the Office of Science of the U.S. Department of Energy as part of the Climate Science for a Sustainable Energy Future (CSSEF) program. The Pacific Northwest National Laboratory is operated for DOE by Battelle Memorial Institute under contract DE-AC05-76RL01830.

Licensing

References

[1] Obama, Barack. “Executive Order -- Making Open and Machine Readable the New Default for Government Information”. www.whitehouse.gov . 09 May 2013. 04 September 2013. http://www.whitehouse.gov/the-press-office/2013/05/09/executive-order-making-open-and-machine-readable-new-default-government-

[2] Berrueta, Diego, and Jon Phipps. "Best Practice Recipes for Publishing RDF Vocabularies-W3C Working Group Note." *World Wide Web Consortium (WC), Aug.,.: http://www. w3. org/TR/2008/NOTE-swbp-vocab-pub-20080828/.(Cit. on pp.,)* (2008).

[3] W3C. “RDF Vocabularies Current Status”. http://www.w3.org/standards/techs/rdfvocabs#w3c_all . 01-August-2013. 04 September 2013.

[4] McBride, Brian. "Jena: A semantic web toolkit." *Internet Computing, IEEE* 6.6 (2002): 55-59.

[5] Horridge, Matthew, Sean Bechhofer, and Olaf Noppens. "Igniting the OWL 1.1 Touch Paper: The OWL API." *OWLED*. Vol. 258. 2007.

[6] Aduna. “Alibaba”. www.openrdf.org. 2012. 04 September 2013. http://www.openrdf.org/alibaba.jsp

[7] U.S. Department of Energy. 1012. CSSEF: Climate Science for a Sustainable Energy Future. http://climatemodeling.science.energy.gov/projects/cssef-climate-science-sustainable-energy-future.

[8] Collins, William D., et al. "The formulation and atmospheric simulation of the Community Atmosphere Model version 3 (CAM3)." *Journal of Climate* 19.11 (2006): 2144-2161.

[9] Rew, Russ, and Glenn Davis. "NetCDF: an interface for scientific data access." *Computer Graphics and Applications, IEEE* 10.4 (1990): 76-82.

[10] Lawrence, B. N., et al. "Maintaining and advancing the CF standard for earth system science community data." (2003).

[11] Noy, Natalya F., et al. "Creating semantic web contents with protege-2000."*Intelligent Systems, IEEE* 16.2 (2001): 60-71.

[12] Williams, Dean N., et al. "The Earth System Grid: Enabling access to multimodel climate simulation data." *Bulletin of the American Meteorological Society* 90.2 (2009): 195-205.

[13] Lebo, Timothy, et al. "Prov-o: The prov ontology." *W3C Recommendation, http://www. w3. org/TR/prov-o/(accessed 30 Apr 2013)* (2013).

[14] Weibel, Stuart, et al. "Dublin core metadata for resource discovery." *Internet Engineering Task Force RFC* 2413 (1998): 222.

[15] Brickley, Dan, and Libby Miller. "The Friend of a Friend (FOAF) project." (2000).

[16] Reynolds, Dave. The Organization Ontology. 25 June 2013. 04 September 2013. http://www.w3.org/TR/vocab-org/