A SEMANTIC ELECTRONIC LAB NOTEBOOK FOR EDUCATION

William Kudrle, Rupa Iyer
Center for Life Sciences Technology, College of Technology
University of Houston

Abstract

The electronic lab notebook (ELN) is becoming an increasingly valuable component for scientific research, and scientific research is increasingly turning to semantic web technologies to solve data growth and integration challenges. The integration of semantic web technologies with ELN capabilities is thus proving to be advantageous for scientific research and this paper describes further developments in this direction. Specifically, we have upgraded an existing online spreadsheet that is integrated with the Drupal content management system and then enhanced it with semantic web technologies for use with our Drupal-based ELN. We discuss how these capabilities relate to the recent concept of research objects and how this approach can improve data sharing and distributed analysis for scientific experiments.

Introduction

In a previous paper [1] we described how version 6 of the Drupal content management system (drupal.org) has been used as the basis for an electronic lab notebook (ELN) in an educational biotechnology lab. This platform has been given the acronym of PERC, for Platform for Education and Research Collaboration. PERC provides online spreadsheets and forms for data collection, geo-location of data points on Google maps, basic data graphing capabilities, collaboration via social networking and analysis tools via web services.

With the newest version of Drupal (version 7 was officially released in January 2011) semantic web technologies have been incorporated into the core software, making them available to all Drupal 7 installations. This is significant because, according to the World Wide Web consortium (W3C) and Tim Berners-Lee, the inventor of the web, the semantic web is the next stage in the evolution of the World Wide Web [2]. Often called "Web 3.0", it extends current web standards with technologies such as the Resource Description Format (RDF) for structured content, RDF Schema and the web ontology language (OWL) for metadata relationships and the SPARQL Protocol and RDF Query Language (SPARQL) for distributed data queries. These technologies are designed to make it easier for computer software to parse, analyze and draw inferences from digital web-enabled content. Semantic web technologies have already been shown to be advantageous for scientific data sharing and analysis [3], and so it is a natural step to extend the PERC platform with these additional semantic web capabilities.

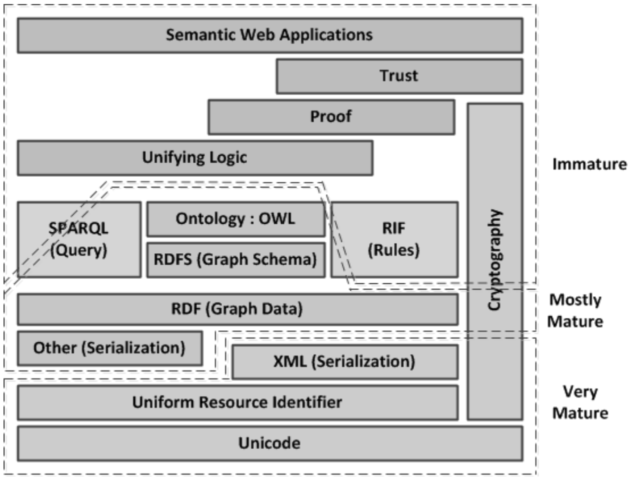


Figure 1: Semantic Web Stack (after [4]).

To describe the different components of the semantic web, a commonly used diagram is the semantic web stack, or "layer cake", as shown in Figure 1. The lower layers are mature technologies and provide the basis of the current World Wide Web. The middle layers (e.g., RDF, RDFS, OWL) are more recent in development but do have standards recommended by the W3C and are being used in an increasing number of applications. The upper layers (e.g., Trust, Proof) have yet to be standardized by the W3C, so an implementation of a recommended standard is not currently possible for those layers.

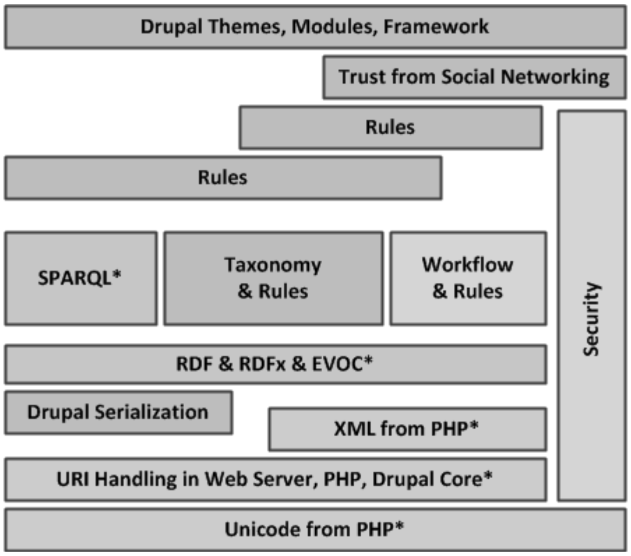


Figure 2: Drupal Semantic Web Functionality The * denotes Drupal modules that do well at matching W3C recommendations.

Figure 2 shows our assessment of how various modules or groups of modules from the Drupal content management system (CMS) create functionality that parallels at least some of the corresponding features in the semantic web stack. The modules with an asterisk have implementations based primarily on the recommended W3C web standards. Modules or module groups with no asterisk have some similarities to the corresponding layer in the semantic web stack in functionality, but either do not adhere specifically to semantic web standards or the W3C standards for these layers do not exist at this time.

While only components in the lower part and some of the middle portion of the Drupal semantic web stack currently utilize the semantic web standards, this is enough to realize many of the benefits of the semantic web approach. In the Drupal 7 version of PERC (denoted here as PERC7), we are able to utilize the RDF modules (RDF, RDFx, EVOC) and the SPARQL module (SPARQL) to provide semantic web capabilities in relation to scientific data, as described below. Note that the lowest levels (XML, URI Handling) are basic to the Drupal CMS and are by default available for all sites.

Spreadsheets and Research Objects

In the implementation of semantic web capabilities for an ELN, we have concentrated initially on the spreadsheet module of Drupal. The spreadsheet is widely used in many contexts, including the collection, analysis and sharing of experimental data [5, 6]. Other projects have also adapted spreadsheets for semantic web technologies, including the ability to export data from spreadsheets to RDF stores [7, 8]. The widespread use of spreadsheets in current research environments has been a primary motivator for extending spreadsheets with semantic web technologies while at the same time trying to minimize the learning curve required by researchers who use them [9]. In the context of an ELN, the flexibility of a spreadsheet presents several attractive possibilities. For scenarios that are primarily educational, the spreadsheet can be structured prior to the experiment according to the specific desired learning objectives of the project. For more free-form research, a spreadsheet gives researchers a structure for storing the parameters and results from their investigations while not a priori requiring a certain format or workflow.

In creating an ELN that is relevant to the future of scientific research and education, it is valuable to also consider the recent developments in Research Objects (RO) [10] and scientific social objects [11]. The goal of an RO is to “create a class of artifacts that can encapsulate our digital knowledge and provide a mechanism for sharing and discovering assets of *reusable* research and scientific knowledge” [10]. Technical standards for the RO are still evolving based on best practices from several projects. Just as in the case of the semantic web layer cake, however, certain principles have been articulated in a more abstract sense and provide some guidelines concerning the nature of an RO. For example, an RO should be designed with the dimensions of being *reusable, repurposeable, repeatable, reproducible, replayable, referenceable, revealable* and *respectful* [11]. The RO should, in effect, encapsulate all of the artifacts relevant to a given experiment so that it can be reproduced, extended, re-investigated and referenced by other researchers. We might note that the type of experiment will also bear on which dimensions can be applicable. For example, the dimension of *replayable* would only be applicable for *in-silico* experiments or other experiments carried out fully by automated means such as robots.

An appropriate goal for a robust and forward-looking semantic ELN like PERC7 would be to produce an RO for any given experiment to the extent that the RO dimensions are applicable. The use of spreadsheets in PERC7 takes a step towards that goal. Similar to the way in which Drupal implements a portion of the semantic web stack and provides significant benefits from that implementation, we can also use spreadsheets to implement dimensions of an RO and provide significant value to an ELN.

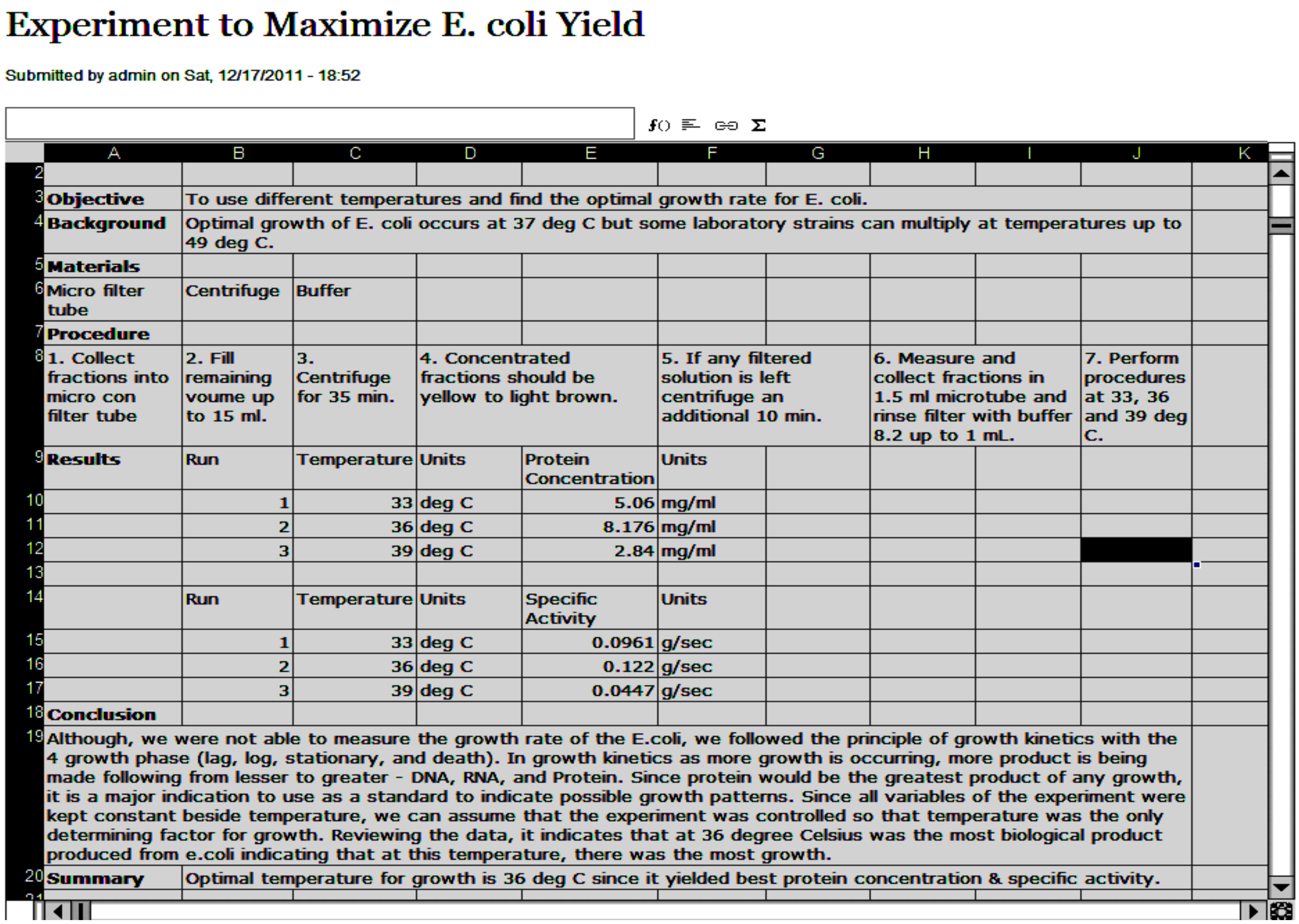
In an abstract sense, it is conceivable that a spreadsheet could fulfill all of the dimensions for an RO if structured correctly with the appropriate information and within the right environment. In a practical sense, though, the most straightforward dimensions to achieve for the experimental data in a spreadsheet would be those of being *reusable*, *repurposeable*, *repeatable*, and *reproducible* (reproducible in effect being a subset of repeatable). These should be achievable with some care and proper preparation of the data in the spreadsheet, and will be explored further in the example shown later in this paper.

It can be noted that the additional dimensions of being *replayable*, *revealable* and *respectful* should be achievable as well in many cases with the use of the Drupal CMS and additional appropriate modules (e.g., the Maestro module for workflow would help enable the *replayable* dimension). The dimension of *referenceable* will be dependent on what becomes accepted practice for referencing digital publication objects, which is still in the process of being defined in a way that is acceptable by interested parties [12]. Since the RO concept holds such promise for the future of science, it is useful to examine how the implementation of an embedded semantic spreadsheet in PERC7 can help to provide a foundation for more extensive RO capabilities.

Implementation

In the previous version of PERC, the sheetnode module was used to encapsulate a robust Javascript spreadsheet for experimental data. For PERC7 we have upgraded the sheetnode module for Drupal 7 and have incorporated RDF export capabilities. An example of an experimental write-up using the data from one of our student groups is shown in Figure 3. This example shows a relatively simple experiment that determines the optimal temperature in the growth of E. coli [13]. The spreadsheet contains all of the relevant information for the experiment and so in the sense of being *reusable*, *repurposeable*, *repeatable*, and *reproducible* it could be considered a basic RO.

The power of the RO increases significantly when the information from the spreadsheet is encapsulated using semantic web technologies. By exporting the information from the spreadsheet in RDF format, it can then be explored as a knowledge-base, which is similar to a traditional relational database but has improved capabilities for defining richer relationships. The information includes any tagged elements from the experiment, such as the objective, materials, procedures and summary as well as the data. Using the W3C standard of SPARQL for a query language [14]



Experiment to Maximize E. coli Yield

Submitted by admin on Sat, 12/17/2011 - 18:52

	A	B	C	D	E	F	G	H	I	J	K
2											
3	**Objective**	To use different temperatures and find the optimal growth rate for E. coli.									
4	**Background**	Optimal growth of E. coli occurs at 37 deg C but some laboratory strains can multiply at temperatures up to 49 deg C.									
5	**Materials**										
6	Micro filter tube	Centrifuge	Buffer								
7	**Procedure**										
8	1. Collect fractions into micro con filter tube	2. Fill remaining voume up to 15 ml.	3. Centrifuge for 35 min.	4. Concentrated fractions should be yellow to light brown.		5. If any filtered solution is left centrifuge an additional 10 min.		6. Measure and collect fractions in 1.5 ml microtube and rinse filter with buffer 8.2 up to 1 mL.		7. Perform procedures at 33, 36 and 39 deg C.	
9	**Results**	Run	Temperature	Units	Protein Concentration	Units					
10		1	33	deg C	5.06	mg/ml					
11		2	36	deg C	8.176	mg/ml					
12		3	39	deg C	2.84	mg/ml					
13											
14		Run	Temperature	Units	Specific Activity	Units					
15		1	33	deg C	0.0961	g/sec					
16		2	36	deg C	0.122	g/sec					
17		3	39	deg C	0.0447	g/sec					
18	**Conclusion**										
19	Although, we were not able to measure the growth rate of the E.coli, we followed the principle of growth kinetics with the 4 growth phase (lag, log, stationary, and death). In growth kinetics as more growth is occurring, more product is being made following from lesser to greater - DNA, RNA, and Protein. Since protein would be the greatest product of any growth, it is a major indication to use as a standard to indicate possible growth patterns. Since all variables of the experiment were kept constant beside temperature, we can assume that the experiment was controlled so that temperature was the only determining factor for growth. Reviewing the data, it indicates that at 36 degree Celsius was the most biological product produced from e.coli indicating that at this temperature, there was the most growth.										
20	**Summary**	Optimal temperature for growth is 36 deg C since it yielded best protein concentration & specific activity.									

Figure 3: Spreadsheet that summarizes the experiment.

along with the SPARQL endpoint that is included with Drupal's SPARQL module, the information from any experiment on a given ELN website can be queried. The SPARQL query language is similar in concept to the Structured Query Language (SQL) that has been in use for over 30 years to query relational databases, but has been optimized for use on the World Wide Web and produces output in the form of RDF triples. SPARQL commands also provide more features for anonymous discovery of the content and structure of the information in the SPARQL endpoint, which opens up significant additional possibilities.

When the SPARQL endpoint is queried for the experiment shown in Figure 3, a short snippet of the expected RDF output is shown in Figure 4, using the RDF N Triples format. This output corresponds to the content of cells C15 and D15 in the spreadsheet. The second entity in the RDF triple is the predicate, and references a vocabulary or ontology. In Figure 4, the predicates use the namespace prefix of "mdex".

```
 …
<http://perc7.dev/node3#C15>
<mdex:ExperimentalValue>
"33"
<http://perc7.dev/node3#D15>
<mdex:ExperimentalUnits>
"deg C"
 …
```

Figure 4: Fragment of the RDF output from the spreadsheet in Figure 3.

While there have been several sophisticated vocabularies that have been developed with experimental data in mind [15,16], for simplicity we have built and utilized a vocabulary called Minimal Description for Experiments with the namespace of mdex. The vocabulary file is available from http://mylabbook.org/mdex/0.1/mdex#. This provides a vocabulary for experiments that is easily understood and utilized for basic experiments like this one.

There is an important intermediate step between creating the experimental information in the spreadsheet and making it available as RDF triples via the SPARQL endpoint. There needs to be a way to designate how one or more given spreadsheet cells will correspond to, or be mapped to, a given RDF output triple. There have been several sophisticated prior approaches to carry out this mapping [7,8,9].

Tag	Meaning
<subject>	Encloses tags pertaining to the subject and the overall triple
<predicate>	Encloses tags pertaining to the predicate and object
<object>	Encloses tags pertaining to the object
<uri>	URI of the given cell in the spreadsheet
<literal>	Use the value in the designated cell in the spreadsheet
<literal_multi>	In the spreadsheet cell, there are multiple literal values separated by commas

Table 1: Tags used in the XML mapping file.

Our approach uses an XML mapping file to describe the correspondence between the spreadsheet cells and the desired RDF output. The element tags in the XML mapping file are based on the basic subject – predicate – object structure of the RDF triple. This simple approach should be relatively easy to understand by those familiar with RDF triples and it provides flexibility in how spreadsheet cells can be mapped. The tags used in the XML mapping file are shown in Table 1.

Using this approach, the fragment of an XML mapping file is shown in Figure 5. This fragment provides the information needed by the Drupal module concerning how to construct the fragment of the RDF output file displayed in Figure 4.

For any given experiment, the researcher needs to utilize an XML mapping file if they want to produce RDF output. There can be only one XML mapping file associated with any given spreadsheet. The XML mapping file can be re-used, for example when the same experiment is performed and only the data changes. This simplifies the setup for many educational scenarios, where students are performing the same experiment, but with perhaps different parameters. It is also very suitable for automated experiments.

```
…
<subject>
 <uri>C15</uri>
 <predicate>
 <uri>mdex:ExperimentalValue</uri>
 <object>
   <literal>C15</literal>
 </object>
 </predicate>
</subject>
<subject>
 <uri>C16</uri>
 <predicate>
 <uri>mdex:ExperimentalUnits</uri>
 <object>
   <literal>C16</literal>
 </object>
 </predicate>
</subject>
…
```

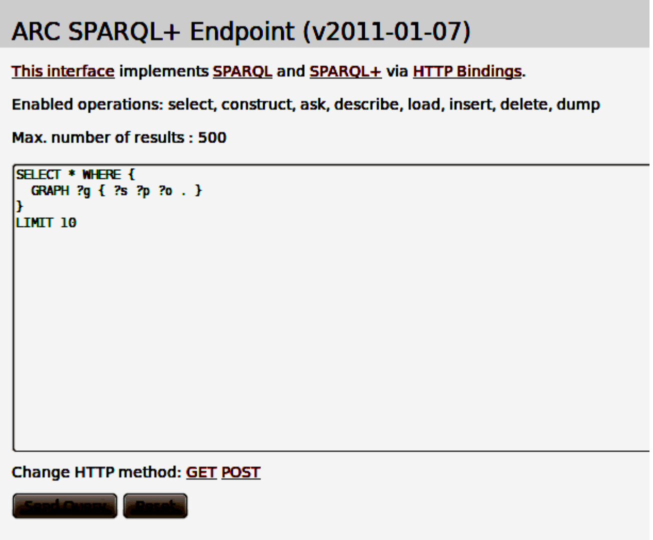
Figure 5: XML Mapping fragment corresponding to Figures 3 and 4.

SPARQL Endpoint

The SPARQL endpoint provides a URL against which information about the experiments can be queried, often in an anonymous manner. In the case of the Drupal SPARQL endpoint, a textbox is provided in which to enter the SPARQL query, as shown in Figure 6. The RDF output from the SPARQL query can be in one of several formats, including RDF/XML, Turtle, JSON or an HTML table and should be similar to the RDF output shown in Figure 4.

Using this approach provides several significant advantages when compared to the usual practice of using relational databases to store data.

- The SPARQL endpoint can be made publically available to anyone who wishes to query the data.
- The researchers do not have to make any extra effort to provide public access to the spreadsheet, format it in any special way or manage user lists or authentication.
- The data in the SPARQL endpoint is fully discoverable through the appropriate SPARQL queries using statements such as DESCRIBE and SELECT. Since RDF triple-stores do not use schemas as in relational databases, then the user does not need to know table names or schemas beforehand. For example, the query shown in Figure 6 will show all of the data that is stored as RDF triples.


ARC SPARQL+ Endpoint (v2011-01-07)

This interface implements SPARQL and SPARQL+ via HTTP Bindings.

Enabled operations: select, construct, ask, describe, load, insert, delete, dump

Max. number of results : 500

```
SELECT * WHERE {
  GRAPH ?g { ?s ?p ?o . }
}
LIMIT 10
```

Change HTTP method: GET POST

Figure 6: Interface to the SPARQL endpoint in Drupal, based on the ARC2 libraries.

As accepted protocols are developed for research objects in RDF repositories, then the discovery process should become even easier. These capabilities open many possibilities for mixing and matching data results between experiments, as long as the experimental results are both stored in an RDF store with a SPARQL interface.

Extending the Experiment

To demonstrate some of the advantages of having the basis of research objects available via the SPARQL endpoint, we can make a simple modification to the previous experiment and its associated spreadsheet. In addition to measuring the temperature in the above experiment, we could also measure the pH of the growth medium to determine optimal conditions for E. coli yields.

To carry out this new set of measurements, we would have 2 possibilities. First of all, we could add these new measurements for pH to the existing spreadsheet, modify the XML mapping file accordingly, and then export the RDF from the spreadsheet. The new measurements would then be available from the SPARQL endpoint queries. The flexibility of this approach arises from the fact that the RDF format does not require changes to a database schema when the type of data for the experiment changes (as would usually be needed if we used a standard relational database). The only changes that would be required would be the addition of columns in the spreadsheet and the appropriate mapping XML elements to the XML mapping file.

The other approach would be to create a new Drupal node with a copy of the spreadsheet from the first experiment. Then we could run the entire experiment again to obtain fresh values for all the data points, including those for pH. This would use a new base URI for the revised experiment and its data values. The only changes would be again to add the column in the spreadsheet and modify the XML RDF mapping file appropriately. When the RDF output is created for this revised experiment, it would contain another set of data for temperature and would also include the data for pH variations. These two datasets could co-exist in the RDF store. Data could be drawn from both of the datasets if desired without any conflict.

To extend these possibilities further, data from a query from another SPARQL endpoint on another website that housed data for similar experiments could be obtained from a simple SPARQL query. Currently there are no capabilities for federated queries from multiple SPARQL endpoints in Drupal. However, it is possible to use other SPARQL mash-up tools like Semantic Pipes [17] or MashQL [18] to combine data results from several SPARQL endpoints to obtain more complete data sets. A further possibility would be to combine these data export features with a form of automated data acquisition in the laboratory [19].

We should note that instead of using a spreadsheet in these experiments, we could use Drupal forms and their associated RDF mapping capabilities that are now built into Drupal core to produce similar results for some types of experiments. However, the use of forms requires more preparation. And the use of the XML mapping file in the case of spreadsheets gives the additional capability of mappings between spreadsheet cells, for example designating a spreadsheet cell URI for the subject and a spreadsheet cell URI or literal for an object of the triple. This would not be easy to reproduce with the standard core RDF mapping approach.

Conclusion

We have shown how upgrading the spreadsheet (known as the sheetnode) module for Drupal 7 and extending it with RDF export capabilities can prove to be quite useful within the context a Drupal based ELN. It can provide the basic features of a research object. This combination also provides capabilities for easily sharing data in an anonymous manner to other interested researchers, and provides the basis for being able to “mash up” scientific data with other experiments.

Although we have emphasized the use of spreadsheets for documentation and data collection for scientific experiments, several other tools are emerging that hold promise for using the semantic web capabilities of Drupal. For example, the SPARQL Views module allows RDF data to be queried and then reformatted in one of several numerous formats. As an extension of the popular Drupal Views module, there are capabilities for plotting and filtering data with tables, lists, graphical plots, Google maps, and other formats. As this SPARQL Views module and other modules become more mature, the Drupal 7 platform should provide even more analysis and display capabilities integrated with semantic web technologies for use with an ELN. Furthermore, as research objects become more clearly defined, increased synergistic possibilities could be realized between PERC7 and research objects.

We are documenting developments of this approach for a semantic ELN on our website at www.mylabbook.org. A demonstration of the capabilities of PERC7 can be viewed at the URL of www.mylabbook.org/perc7. The interested reader can also download the latest versions of the RDF extensions to the sheetnode module from that website, which has been submitted to the drupal.org maintainers for that module.

Acknowledgements

We thank Ali Givnamesh for providing helpful comments about this paper and Michael Cole for developing the Drupal Quickstart environment that made development time much more productive.

References

1. M.Elliott. The state of the ELN Market. Scientific Computing World. Dec. 2006/Jan. 2007.

2. M. Elliott. What You Should Know Before Selecting an ELN. Scientific Computing. June 2009.

3. M. Elliott. Electronic Laboratory Notebooks Enter Mainstream Informatics. Scientific Computing. Nov. 2008.

4. J. Pollock, Semantic Web for Dummies, Figure 9-2.

5. Spreadsheet Simulation by Andrew F. Sella

6. Use of Spreadsheets for Demonstrating Power and Variability, 1999

7. L. Han, T. Finin, C. Parr, J. Sachs, A. Joshi, RDF123: From Spreadsheets to RDF. The Semantic Web - ISWC 2008, pp. 451-466 .

8. A. Langegger, W. Woss, XLWrap - Querying and Integrating Arbitrary Spreadsheets with SPARQL. Proceedings of the 8th International Semantic Web Conference (ISWC2009), Washington D.C. Springer, 2009.

9. K. Wolstencroft, S. Owen, M. Horridge, O. Krebs, W. Mueller, JL Snoep, F. du Preez, C. Goble, RightField: Embedding ontology annotation in spreadsheets. Bioinformatics, 2011 Jul 15;27(14):2021-2.

10. S. Bechhofer, D. De Roure, M. Gamble, C. Goble, I. Buchan. Research Objects: Towards Exchange and Reuse of Digital Knowledge. The Future of Web for Collaborative Science. April 2010, Raleigh, NC, USA.

11. D. De Roure, S. Bechofer, C. Goble, D. Newman, Scientific Social Objects. 1st International Workshop on Social Object Networks. 2011.

12. M. Nielsen, Reinventing Discovery, the New Era of Networked Science. Princeton University Press, 2011.

13. R.S. Iyer, M. Wales. Integrating Interdisciplinary Research-Based Experiences in Biotechnology Laboratories. Accepted for publication, Journal of Advances in Engineering Education, Fall 2011 issue.

14. Bob DuCharme, Learning SPARQL, O'Reilly Media, July 2011.

15. L. Soldatova, R. King, An ontology of scientific experiments, J. R. Soc. Interface, 2006 (3)795-803.

16. P. Ciccarese, E. Wu, T. Clark, An Overview of the SWAN 1.0 Discourse Ontology, 16th International World Wide Web Conference Proceedings, 2007.

17. D. Le-Phuoc, A. Polleres, M. Hauswirth, G. Tummarello, C. Morbidoni, Rapid Prototyping of Semantic Mash-ups through Semantic Web Pipes, 18th International World Wide Web Conference Proceedings, 2009.

18. M. Jarrar, M. Dikaiakos, MashQL: A Query-by-Diagram Topping SPARQL, The 2nd International Workshop on Ontologies and Information Systems for the Semantic Web, 2008.

19. A. Givnamesh, R. Iyer, D. Benhaddou, Integrated Remote Management for Bioprocessing Experiments, International Journal of Research and Innovation, 2011, Vol. 1 (1), 75-77.

Biographical Information

William Kudrle, Ph.D., is a program manager in the department of Engineering Technology, in the College of Technology at the University of Houston. He has built websites and software packages in the health, medical, energy and travel industries as well as at several academic institutions.

Rupa Iyer, Ph.D., is an Associate Professor in the department of Engineering Technology, in the College of Technology at the University of Houston. She is the founding director of Biotechnology programs and also directs the Center for Life Sciences Technology. In this capacity she has been responsible for developing the Biotechnology degree program and the core initiatives of the center that include education, research, workforce development and outreach. Her research interests are in environmental biotechnology and interdisciplinary research-based education.