Bootstrapping to a Semantic Grid

Jens Schwidder[1], Tara Talbott[2], Jim Myers[2]
[1]*Oak Ridge National Laboratory,* [2]*Pacific Northwest National Laboratory*
schwidderj@ornl.gov, Tara.Talbott@pnl.gov, Jim.Myers@pnl.gov

Abstract

The Scientific Annotation Middleware (SAM) is a set of components and services that enable researchers, applications, problem solving environments (PSE) and software agents to create metadata and annotations about data objects and document the semantic relationships between them. Developed starting in 2001, SAM allows applications to encode metadata within files or to manage metadata at the level of individual relationships as desired. SAM then provides mechanisms to expose metadata and relationships encoded either way as WebDAV properties. In this paper, we report on work to further map this metadata into RDF and discuss the role of middleware such as SAM in bridging between traditional and semantic grid applications.

1. Introduction

Scientific progress depends increasingly on effective collaboration between widely distributed communities of researchers at various institutions around the world. The amount of data produced and shared is enormous and more effective ways to organize the information and keep track of dependencies are becoming very important. The semantic data grid (SDG), an anticipated merger of semantic web and data grid concepts, is envisioned as the solution to this problem – a scalable means of sharing data, and its context of descriptive information and relationship to other data, through standard protocols and description languages.

However, many obstacles remain before SDGs can fulfill their promise. SDG concepts and software are still evolving and, while the potential uses of data with explicit semantics are compelling, the mechanics of how semantic information will be captured, as well as the economics of metadata production and consumption are very unclear. In particular, while SDGs enable a new class of applications that will become critical to information intensive science efforts, it is not so clear that they provide enough direct benefit to traditional science applications to justify upgrading them to use semantic technologies. Further, since traditional applications are the producers of primary data and metadata, the SDG may have a bootstrapping problem.

The Scientific Annotation Middleware (SAM), being developed by researchers at Pacific Northwest National Laboratory and Oak Ridge National Laboratory, has been created in part, to serve as a research platform for understanding these issues. SAM provides general data/metadata storage capabilities that can be accessed via a number of interfaces with varying levels of metadata awareness. Further, SAM provides configurable datatype-specific mechanisms to map information submitted via a simple interface into information with explicit semantics exposed via other interfaces. For example, as described below, this capability can be used, to expose information within binary files as RDF-encoded relationships. This type-specific mechanism provides an alternative to more generic methods of extracting metadata from text, web pages, and XML [1]. Further, as middleware, SAM allows the metadata extraction process to be defined independently of the data format and the producing application and therefore, for the costs of metadata generation to potentially be transferred to those who can benefit from it.

In the following sections, we provide additional information about SAM in general and describe the mechanisms we have developed to support metadata extraction and bridging between multiple metadata management interfaces, focusing in particular on work to expose metadata via RDF. These are then discussed in terms of their potential to support semantic applications such as semantic data discovery, annotation, and provenance services over data provided by more traditional applications.

2. Background

As shown in Figure 1, SAM is a layered set of middleware components and services for managing data annotations and the semantic relationships among data objects [2]. Conceptually, SAM presents applications with a schema-less store that can manage arbitrary metadata and relationships that are defined by namespace qualified names. As such, it is well suited to a written-by-one-read-by-many usage model in which multiple semantic applications contribute unique information about different

aspects of federated data generated by independent scientific applications, all of which (data and metadata) must be presented to the user and further analysis tools as an integrated data context.

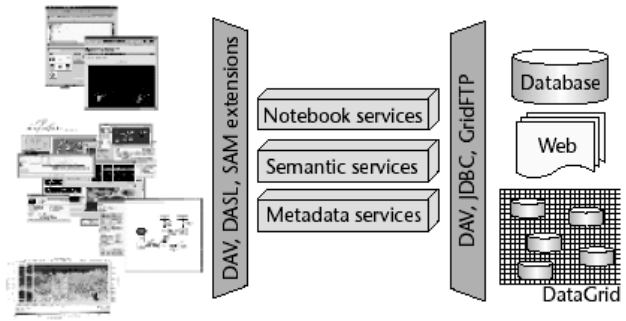

Figure 1. Scientific Annotation Middleware

SAM is built on the Jakarta Slide [3] content management system and implements the web Distributed Authoring and Versioning (webDAV) protocol [4]. WebDAV and its extensions adopt the Web's HTTP model of resources accessed via a URL, adding standard methods for creating new collections (directories) and resources, adding and querying name–value-pair properties (arbitrary strings or XML) associated with each resource, and supporting versioning, locks, and list-based access control [5,6]. WebDAV is an IETF standard and is supported by a wide range of client and server applications including open-source and commercial projects, such as Jakarta Slide, Apache Tomcat, Adobe Acrobat, Mac OS X, and Microsoft Windows [7]. Also among the clients are file system drivers that allow accessing a webDAV server like a local file system. For the purposes of this paper, the most relevant methods are PUT for uploading content, and PROPPATCH and PROPFIND for setting and retrieving properties, respectively.

Slide implements a webDAV-centric content repository as middleware that can store data and metadata in multiple independent underlying data stores, which could be remote, e.g. GridFTP servers and Grid metadata catalogs. When new resources are created using webDAV, Slide generates standard webDAV properties that describe the resource, such as its type, size, owner, and creation date.

SAM extends Slide in a number of ways that enhance its ability to function as a bridging mechanism. To make activities in SAM visible to third-party software, we have modified Slide to produce Java Messaging Service (JMS) events whenever the resources are accessed or modified via webDAV. Supplementing Slide's default internal authentication method, we've added a Java Authentication and Authorization Services (JAAS) based mechanism to allow SAM to be configured to use external authentication services, e.g. a Grid MyProxy server [8].

3. Mapping Between Embedded Metadata and Properties

Through webDAV, SAM can be accessed either as a file system using third party drivers or natively as a resource-plus-properties repository. In designing SAM, we wished to map between these two models and add support for an RDF/graph-based interaction model. Towards these ends, we have added a number of capabilities to allow SAM administrators and end users to specify correlations between metadata in files and properties, and between properties and RDF. As shown in Figure 2, this enables end-to-end scenarios where desktop, file-based applications with custom data formats can directly contribute to a shared network of semantic information.

To extract metadata from files, we developed a configurable, automated mechanism that can run a series of user-defined scripts and web services to produce properties. The mechanism invokes, in order, during a webDAV PUT call, a Binary Format Description (BFD) language script, web service, and/or an XSLT script that have been registered for the relevant content MIME-type.

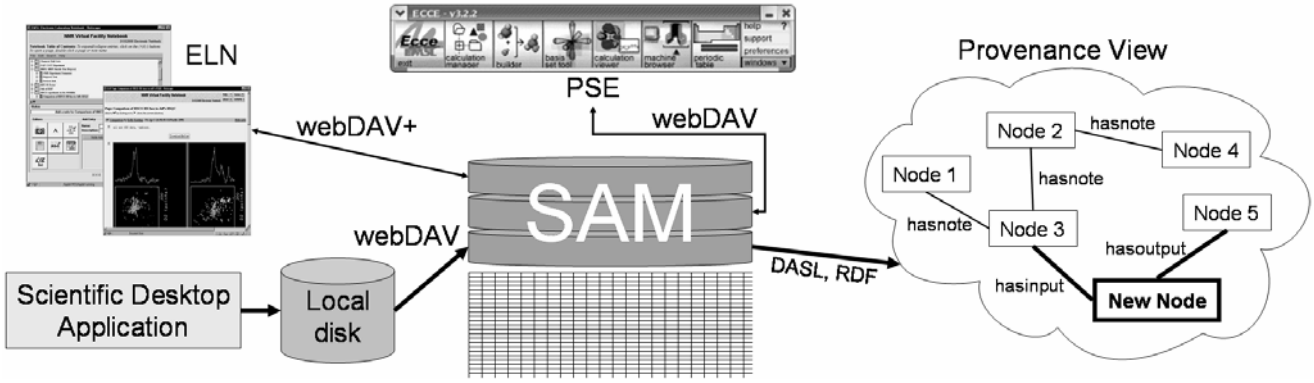

Figure 2. SAM's mechanisms for mapping metadata allows file-based applications, metadata aware applications using webDAV, and RDF-based tools to all contribute to a network of semantic information.

BFD [9] is an extension of the eXtensible Scientific Interchange Language (XSIL) [10] that can describe the layout of a binary or ASCII file format in terms of an XML data model. (BFD is one of the languages influencing the design of the Data Format Description Language (DFDL) standard being pursued through the Global Grid Forum [11].) Analogous with XSLT, a BFD parser can ingest a BFD description and a content file and produce a transformed XML output. In SAM, this output can be piped to a web service supporting a simple WSDL interface that includes a transform method. Any registered XSLT script is invoked in a final step and the resulting output is interpreted as though it were the payload of a webDAV PROPPATCH method. This mechanism is shown in the top half of Figure 3. While we in general describe this capability as a means of semantically labeling information already within the data in some form, i.e. as metadata extraction, it should be noted that it can also be used for additional metadata annotation, e.g. to document inter-file relationships implicit in the design of applications that store data sets as multifile collections, facts that cannot be inferred from the data files alone.

A similar mechanism can be invoked to generate translations and views in SAM. SAM creates a "hastranslations" property specifying 'virtual' URLs for the translated content that can be generated by BFD, web service, and XSLT sequences. Translations are then created dynamically, instantiating the translation URLs when they are requested. While this feature has primarily been used to file translations and web pages showing file content (static HTML pages or pages invoking Java applets), we have recently added a means of specifying that the URL for the data and/or the set of webDAV properties be included in the stream being transformed, allowing the translator to include information from properties in an output file and thereby providing a mechanism to map backwards from properties to content.

4. Mapping Between Properties and RDF

Enabling metadata in SAM to be accessed via RDF requires adding two related pieces of functionality; a mapping between the syntax of webDAV properties and RDF, and new access methods for retrieving and adding RDF statements. Our initial work to extend SAM in these directions is described below, followed in the next section by a more general discussion of the advantages and limitations of the described approach.

At a basic level, webDAV properties map well to RDF statements. Resource URLs become subjects, property names are predicates, and the property value can be interpreted as the object. WebDAV is following the XML namespace conventions for property names, which makes it straight forward to interpret properties as predicates of RDF statements. For the simplest properties, i.e. those with string values, this mapping is fairly intuitive. For example a webpage, http://www.example.org/index.html, which has a "creator" property as defined in the Dublin Core [12] (hereafter shown as dc:creator) whose value is "John Smith" would result in the following RDF:

```
<rdf:Description rdf:about="http://www.example.org/index.html">
  <dc:creator xmlns:dc="http://purl.org/dc/elements/1.1/">
    John Smith
  </dc:creator>
</rdf:Description>
```

However, for properties containing XML values, a number of issues arise. In theory, the use of XML in webDAV property values raises all the same issues as when attempting to interpret general XML documents as RDF [13]. To date, however, the use cases we've encountered use XML within property values for a relatively limited set of reasons. We have seen this in work within the SAM project to adapt notebook and wiki applications and in collaborations with other projects adapting science applications, portals, and problem solving environments. For example, XML is being used to overcome the webDAV limitation of one property with a given name per resource, i.e. to list multiple dc:creators for a document. XML is also being used to clearly identify URIs rather than leaving them encoded as strings. Perhaps most interesting is the use of XML nesting to represent the sources of individual relationships within a property. For example, the ELN electronic notebook [14], it is possible to include a given entry in two notebooks, e.g. as a means of including content from a public notebook in a group notebook where it will be further annotated. Thus, samns:children relationships written by the ELN need to be scoped as to which notebook they belong to.

To interpret these types of XML properties, we have initially implemented logic hardcoding a few conventions sufficient to cover these common use cases. For example, we consider multiple top-level XML elements in a property, or a single top-level rdf:bag element containing multiple rdf:li subelements, as preferred within the Collaboratory for Multiscale for Chemical Science project [15] to imply multiple RDF relationships with a common subject and predicate. Elements including an Xlink href attribute are interpreted as identifying the href as the intended RDF object, while elements with text values are interpreted such that the text is used as the RDF object. Lastly, we have chosen to interpret the format used by the ELN, with an additional layer of XML elements representing the source of the relationships, in terms of RDF reification. The results for a simple multi-valued property and an ELN samns:children property are shown below, with the overall process of mapping from binary/ASCII files to properties and then to RDF shown in Figure 3.

Multivalued Property: dcterms:references

```
<dcterms:references xmlns="...">
  <rdf:Bag>
    <rdf:li>
      <rdf:href xlink:type="simple"
         xlink:title="Paper 1"
         xlink:href="http://collab/paper1.pdf"/>
    </rdf:li>
    <rdf:li>
      <rdf:href xlink:type="simple"
         xlink:title="Paper 2"
         xlink:href="http://collab/paper2.pdf"/>
    </rdf:li>
  </rdf:Bag>
</dcterms:references>
```

Inferred RDF

```
<rdf:RDF xmlns="...">
  <rdf:Description
      rdf:about="/sam/files/nb1/chapter_1">
    <dcterms:references
       rdf:resource="http://collab/paper2.pdf"/>
    <dcterms:references
       rdf:resource="http://collab/paper1.pdf"/>
  </rdf:Description>
</rdf:RDF>
```

Complex Property: samns:children

```
<samns:notebookroot xmlns="..."
     xlink:href="/files/nb_1/">
  <samns:child
    xlink:href="/files/nb1/chapter_1/"
    xlink:title="c1" />
  <samns:child
    xlink:href="/files/nb1/chapter_2/"
    xlink:title="c2" />
</samns:notebookroot>
```

Inferred RDF

```
<rdf:RDF xmlns="...">
  <rdf:Description rdf:about="/sam/files/nb_1">
    <samns:children
        rdf:resource="/files/nb1chapter_2/"
        rdf:ID="statement1" />
    <samns:children
        rdf:resource="/files/nb1/chapter_1/"
        rdf:ID="statement2" />
  </rdf:Description>
  <rdf:Description rdf:about="#statement1">
    <samns:notebookroot
        rdf:resource="/files/nb_1/" />
  </rdf:Description>
  <rdf:Description rdf:about="#statement2">
    <samns:notebookroot
        rdf:resource="/files/nb_1/" />
  </rdf:Description>
</rdf:RDF>
```

Since webDAV, with our conventions for interpreting XML property values, provides a basic means of reading and writing semantic relationships about a resource, our initial focus in providing RDF –based functionality was in returning provenance information, i.e. subgraphs of related resources. Towards this end, very early in the SAM project we implemented dynamically generated properties whose values include all resources linked to the current resource by a specified subset of properties, down to a specified maximum link depth. These properties rely on a common configuration resource that specifies the desired properties and the maximum traversal depth. For the pedigreerdf property, the value is in RDF. For the pedigreegxl property, the same subgraph is encoded in the Graph Exchange Language (GXL) [16] which can be consumed directly by a number of graph display toolkits. (As these properties were intended as a temporary measure primarily supporting the CMCS project (see Discussion), they are both in the CMCS http://purl.oclc.org/NET/SAM/cmcs namespace.)

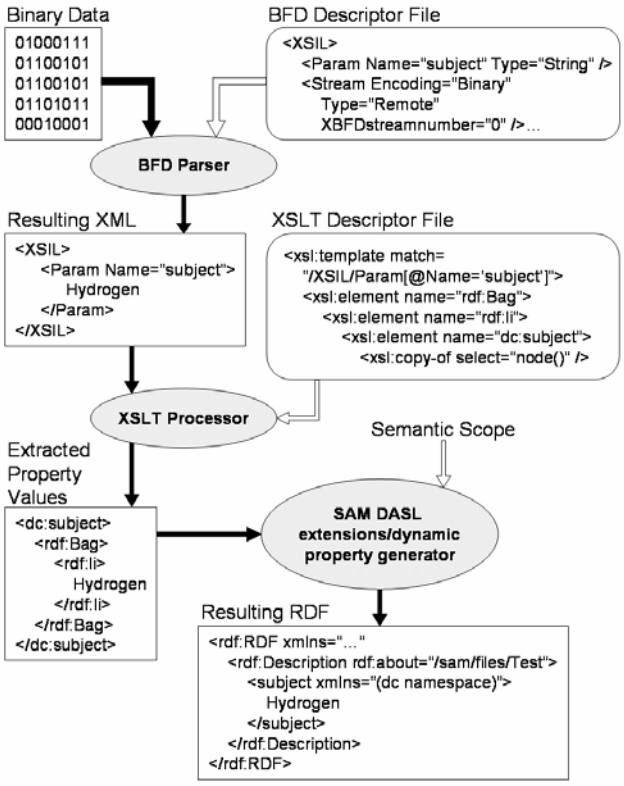

Figure 3. A simple example showing the process within SAM to extract metadata from within files and generate webDAV properties and RDF available as a query resonse. (The optional invocation of a web service before the XSLT step is not shown.)

More recently, we have been implementing RDF related capabilities as extensions to the implementation of the DAV Searching and Locating (DASL) SEARCH method now available within Slide. DASL defines a basic grammar having an SQL-like format, as well as a means to define extended grammars. The basic grammar supports returning a set of properties for all resources within a specified scope and meeting the specified

conditions. For example, one could request the DAV:displayname of all documents within "/projects" whose dc:creator property includes "Jane Smith". For SAM, we have extended this grammar in two ways. First, we allow the scope to be specified in terms of a root resource and a set of properties to follow and a depth, allowing the query to be run over a subgraph analogous to that returned through the pedigreerdf property. Second, we have extended the select mechanism to enable RDF-encoding of the return value, i.e. returning the set of properties on matching nodes as a set of RDF statements generated using the conventions discussed previously. Implementing these capabilities through the SEARCH method instead of through properties allows the set of properties to follow and the depth limit to be specified per query rather than configured per server. Further, it separates the list of properties to be returned from those used to define the scope.

5. Discussion

SAM's ability to separate the effort required for making data semantics explicit from the development and use of scientific applications has a number of potential benefits in the context of community-wide collaborations and grid-based computing. Most directly, SAM allows the costs of describing data semantics explicitly to be born by third parties and/or delayed until the benefits of such labeling can be realized. With SAM's approach, groups wishing to take advantage of metadata-based searching, provenance tracking, annotation services, and other semantic capabilities, do not have to involve the developers of all of the domain software they intend to use in reaching agreement on shared ontologies and upgrading software. Instead, groups can independently define metadata extractors/annotators as needed that expose as much or as little semantic detail as required, mapping it directly into the desired vocabulary. This elevates the concept of a virtual organization as an administrative unit managing access controls and allocations to one that may also manage shared semantics.

The decision to base SAM initially on webDAV and an open source content repository implementation has allowed us to quickly gain practical experience. Most important in the context of this paper has been a collaboration with the Collaboratory for Multiscale Chemical Sciences (CMCS). As reported elsewhere, CMCS has integrated SAM into its collaborative framework (named "KnECS") and has made heavy use of SAM's metadata extractor and translation capabilities to customize the framework and portal for chemical science. CMCS gathers metadata using extractors, through web forms, and from webDAV-enabled applications, PSEs, and web services. CMCS provides a number of general tools that make use of the federated metadata ranging from a data browser and metadata-based search tool to a provenance graphing portlet. Feedback from the CMCS project has been invaluable in refining SAM capabilities and prioritizing development and, while the CMCS project is ongoing, their experience suggests that the decoupling SAM allows will be very important in allowing groups to assemble a comprehensive, living corpus of semantically tagged data and for scaling and evolving collaborative tools in general.

Discussions with CMCS, other collaborators, and developers interested in semantic technologies in general have identified a number of strengths and limitations of SAM's current capabilities and indicated several promising directions for enhancements. While WebDAV and our mapping of properties to RDF statements clearly provide only a subset of what RDF can encode, they have largely proved sufficient to represent the metadata being produced by traditional scientific applications as well as by tools such as electronic notebooks. The webDAV PROPFIND and PROPPATCH methods are conceptually similar to the HTTP extensions proposed as part of the URI Query Agent Model [17] for accessing semantic information, and, by emphasizing access based on a resource URL, webDAV presents a set of information very similar to the Concise Bounded Description of a resource, i.e. a subgraph of outbound relationships [18].

In general, SAM's configurable mechanism for mapping between metadata in files and webDAV properties has worked well. While we anticipate migrating from BFD to DFDL as implementations appear, which should broaden the range of files than can be handled and simplify script development, and we may add some enhancements such as a mechanism to allow extractors to be registered for multiple file types at once, the current capabilities largely address the requirements that have been identified.

The mapping between webDAV properties and RDF and the interface(s) to RDF are less mature and we expect a number of changes. While the conventions we've implemented appear to cover most of the current use cases, there is clearly a desire from developers and end users to have more control over the layout of property values – simply alternate ways of specifying multiple relationships within a property and one level of reification. Further, we anticipate a need to represent more complex graphs in the future as new semantic applications are developed. Towards these ends, we intend to provide a mechanism analogous to that used to move from metadata in files to properties to allow the mapping from properties to RDF to be configured on a per property basis. Following the DFDL model - annotating an XML schema with instructions on how to populate an instance of the schema from ASCII/binary data, this might involve the annotation of an RDF Schema or OWL description with instructions for creating an instance from webDAV

properties. We also intend to investigate adding a SPARQL-based [19] grammar for DASL, implementing the URIQA MPUT, MGET, and MDELETE HTTP methods, and/or implementing semantic grid service interfaces as they are standardized. Lastly, while SAM currently maps semantic relationships to webDAV properties and stores them as such, if the usage of RDF increases, we can potentially invert the mapping direction and use a native RDF store and map to webDAV properties dynamically from the RDF rather than the other way around.

6. Conclusions

With the capabilities reported here, SAM now provides a complete binary-to-RDF pathway for exposing semantic information implicit in science applications and their output file formats. We believe that SAM demonstrates the viability of a bridging approach to include existing scientific applications in semantic data grids. Further, the use of SAM in projects such as CMCS is beginning to demonstrate the value of this approach in reducing integration and system evolution costs in collaborative systems. Absent a strong driver for upgrading current scientific software to be semantically explicit, the ability to provide and track semantic information without having to rewrite an existing application will be needed for quite some time. While SAM is still evolving and does not implement a full SDG, we believe that the concepts being explored within SAM will be critical to the successful realization of SDGs capable of seamlessly integrating an evolving mix of applications and supporting collaboration at the scale required for next-generation information intensive research.

7. Acknowledgements

This work was supported as part of the Scientific Annotation Middleware (SAM) project. Employees of Battelle Memorial Institute, which operates Pacific Northwest National Laboratory for the US Department of Energy under contract DE-AC06-76RL01830 and Oak Ridge National Laboratory under contract De-AC05-00OR22725, wrote this manuscript. The authors also acknowledge helpful discussions and ongoing collaborations with members of the Collaboratory for Multiscale Chemical Science (CMCS) project.

8. References

[1] Bettina Berendt, Andreas Hotho, Gerd Stumme, "Towards Semantic Web Mining", *International Semantic Web Conference (ISWC)*, 2002

[2] James D. Myers, Alan R. Chappell, Matthew Elder, Al Geist, Jens Schwidder, "Re-Integrating The Research Record", *Computing in Science and Engineering*, May/June 2003, pp. 44-50

[3] Jakarta Slide Java Content Management System Website, http://jakarta.apache.org/slide/index.html

[4] Distributed Authoring and Versioning (DAV) website, http://webdav.org

[5] J. Whitehead, "WebDAV: Versatile Collaboration Multi-protocol", *IEEE Internet Computing*, vol. 9, no. 1, Jan/Feb 2005, pp. 66-74. http://www.soe.ucsc.edu/~ejw/papers/dav-ic-2005-final.pdf

[6] DAV Searching and Locating (DASL) Draft Specification, http://webdav.org/dasl

[7] webDAV Projects, http://webdav.org/projects/

[8] J. Novotny, S. Tuecke, and V. Welch, "An Online Credential Repository for the Grid: MyProxy", *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*, IEEE Press, August 2001.

[9] Binary Format Description (BFD) Language Home Page, http://collaboratory.emsl.pnl.gov/sam/bfd/

[10] Roy Williams, "XSIL:Java/XML for Scientific Data", July 2000, http://www.cacr.caltech.edu/SDA/xsil

[11] Document Format Description Language (DFDL), http://forge.gridforum.org/projects/dfdl-wg/

[12] Dublin Core Metadata Initiative, http://dublincore.org/

[13] Stephen Buswell, "Extracting Semantics from XML Structure", http://www.w3.org/2001/sw/Europe/reports/xslt_schematron_tool/

[14] James Myers, Elena Mendoza, and Bonnie Hoopes, "A Collaborative Electronic Notebook", *Proceedings of the IASTED International Conference on Internet and Multimedia Systems and Applications (IMSA 2001)*, August 13-16, 2001, Honolulu, Hawaii

[15] James D. Myers, Thomas C. Allison, Sandra Bittner, Brett Didier, Michael Frenklach, William H. Green, Jr., Yen-Ling Ho, John Hewson, Wendy Koegler, Carina Lansing, David Leahy, Michael Lee, Renata McCoy, Michael Minkoff, Sandeep Nijsure, Gregor von Laszewski, David Montoya, Carmen Pancerella, Reinhardt Pinzon, William Pitz, Larry A. Rahn, Branko Ruscic, Karen Schuchardt, Eric Stephan, Al Wagner, Theresa Windus, Christine Yang, "A Collaborative Informatics Infrastructure for Multi-scale Science", *Proceedings of the Challenges of Large Applications in Distributed Environments (CLADE) Workshop*, June 7, 2004, Honolulu, HI, pp. 24-33

[16] A. Winter, B. Kullbach, V. Riediger: An Overview of the GXL Graph Exchange Language © Springer Verlag: S. Diehl (ed.) Software Visualization · International Seminar Dagstuhl Castle, Germany, May 20-25, 2001 Revised Lectures, available from http://www.gupro.de/GXL/index.html

[17] Patrick Stickler, "URIQA: The Nokia URI Query Agent Model", specification, 2003-2004, http://swdev.nokia.com/uriqa/URIQA.html

[18] Patrick Stickler, "CBD – Consise Bounded Description", specification, 2003-2004 http://swdev.nokia.com/uriqa/CBD.html

[19] "SPARQL Query Language for RDF", W3C Working Draft, http://www.w3.org/TR/rdf-sparql-query/