RE-INTEGRATING THE RESEARCH RECORD

The Scientific Annotation Middleware is a set of components and services that support the creation and use of annotation metadata describing data objects and the semantic relationships among them. It captures aspects of data processing history and the research process and federates the results into a coherent human- and machine-interpretable research record.

Five hundred years after Leonardo da Vinci created his research notebooks, we can still look at them and follow his logic and experimental procedures. Ironically, it would be much harder to decipher a contemporary colleague's notebook; so much of the information required for understanding the work is in various parameter and data files, external databases, the internal logic of the acquisition and analysis software used, and the referenced literature. If we ask more of the modern notebook—that it be remotely accessible and sufficiently complete to guide ongoing work—it becomes completely inadequate. Given that these requirements are typical of the next-generation discovery- and informatics-based science projects anticipated across a wide range of disciplines, a records system that replaces paper and meets the additional requirements of today's research environment is sorely needed.

One route to such a system would be to gather all collaborating parties and, have them standardize their data formats, metadata definitions, and the records process to follow. These parties would then encode the results into communications protocols and schema to implement the system. They would also have to request modifications to any third-party applications used in their community to make them compatible with the system. Such a model forces the community to incur coordination costs upfront.

As anyone who has ever been involved in a standardization effort knows, the effort required increases rapidly as the number of people and the scope of the standard grow. Unfortunately from this perspective, modern science is becoming a global, cross-disciplinary effort that requires growing numbers of researchers to agree on increasing detail about their data and the processes used to create and interpret it. For example, some scientific communities now expect the results reported in publications (such as protein geometries) be submitted in standard formats that make them easily available for further analysis. Furthermore, the trend toward informatics and distance collaboration is blurring the traditional distinction between community-published data and group or private data; researchers soon might have to provide levels of provenance and other annotations on their community contributions that approach or surpass (by including tacit knowledge) the richness of their internal project records. Additionally, researchers

Copublished by the IEEE CS and the AIP

JAMES D. MYERS, ALAN R. CHAPPELL, AND MATTHEW ELDER
Pacific Northwest National Laboratory
AL GEIST AND JENS SCHWIDDER
Oak Ridge National Laboratory

could find themselves working in multiple communities, for example, a chemist could end up providing information relevant to studying atmospheric chemistry, combustion, materials science, geology, and so on, which would require them to work with multiple, incompatible standards.

Clearly, global standardization is not practical as a means of reintegrating the scientific record considering the number of independent data sources and data sinks that exist, the wide range of records-related and scientific metadata that would need to be considered, and the emerging requirements to share intermediate, exploratory results across scientific communities.

The Scientific Annotation Middleware (SAM) project is an attempt to use emerging technologies to separate the initial capture and storage of data and metadata from its subsequent presentation to others and to shift the focus from up-front standardization to on-demand mapping of the original data and metadata into schemas of interest.

Scientific Annotation Middleware

SAM's key concept is the idea of a "schemaless" data store that can accept arbitrary input and dynamically registered translators that map data and metadata into the formats and schema expected by applications and underlying data repositories. This lets researchers capture records-related information using an arbitrary combination of electronic notebooks, applications, agents, and problem-solving environments. Researchers can later define how this information should be translated into forms interpretable in other contexts, such as into a standard schema used in their community, the input format a collaborator's software requires, or the schema of a records management tool or an automated workflow system. In the SAM model, users can maintain data in its original format while defining a view that lets them see all the information via a single interface. Simultaneously, they can define views of this data that conform to the conceptual models of particular applications, groups, institutions, or communities. Thus, even if a given result was created using specific inputs, was part of a specific project, had dependencies on the values of certain measurements reported in the literature, was discussed on a specific notebook page, and had a limited range of validity, and this information were all recorded by different applications and agents, SAM could federate it into the schema understood by a generic "graphical relationship browser." The SAM project Web site (www.scidac.org/SAM) outlines several more complex possibilities.

SAM is a compromise between standards' artificial clarity and real research's organic nature, combining aspects of research notebooks and more structured scientific data management systems. While such a model cannot enforce data-model integrity as well as traditional databases, it is well suited to a write-and-annotate usage model typical of records and to the lightweight federation of independent components. Although implementing and managing such a flexible system might seem difficult, we believe progress on many fronts is making it practical.

Background

Our primary introduction to scientific records management was in the context of electronic notebook (EN) systems' development. We can trace the concept of ENs back at least to the mid 1980s. Since then, the idea of ENs as a relatively direct replacement for personal paper notebooks has evolved to encompass distributed use, multimedia annotations, entry automation, the creation of derivative data sets, and legally defensible records capabilities.[1]

Our efforts began in 1994 and continued as part of a three-way exploration of EN concepts involving researchers at Pacific Northwest National Laboratory, Lawrence Berkeley National Laboratory, and Oak Ridge National Laboratory in the US Department of Energy's DOE2000 Collaboratory program (www.csm.ornl.gov/enote). The DOE2000 notebooks have made significant contributions in the areas of personal and distributed group productivity as well as in defining minimal notebook schema and providing extensibility through programming interfaces.[2,3] In particular, the DOE2000 notebooks were based on the concepts of a typed resource with a URI-style identifier and arbitrary textual key–value pair properties. We included just seven required properties in the initial DOE2000 notebook schema adding a few more over time, primarily to manage digital signature information. This model minimized the EN's knowledge of the data content and format and thus the coupling to applications used to capture, view, and analyze it. Such minimalization provided simple yet powerful capabilities for integrating the notebook with instruments and analysis software and for dynamically configuring third-party components to create and view new data types.

With this architecture, ENs can integrate manually entered notes (such as text and drawings) with the more structured output of scientific instruments and analysis software in a single page view. Although this is an advantage over paper notebooks, the model of an EN as a stand-alone system has since become limiting. Increasingly, problem-solving environments, portals, workbenches, databases, and other information management systems

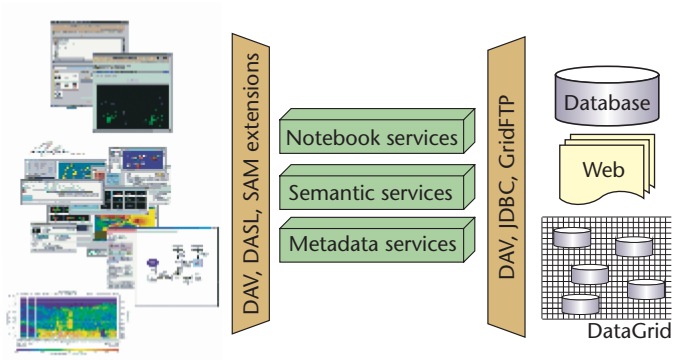

Figure 1. Scientific Annotation Middleware (SAM) service layers presenting a federated view of data to applications, portals, agents, and electronic notebooks.

expect to store the data and interpret its metadata (such as author and type). This has led us to see the EN as one way to view data from federated stores that are equally accessible to other applications, agents, and environments.

Issues that EN researchers have encountered are mirrored in the more general context of the Internet; in fact, EN systems have benefited greatly by adopting technologies such as the World Wide Web and HTML, which enabled standardized distribution of hyperlinked multimedia. The desire to provide multiple views and to expose data's meaning to applications led to the development and use of the Extensible Markup Language (XML) and Extensible Stylesheet Language Transformations (XSLT). These efforts continue under the Semantic Web banner and involve additional technologies (such as the resource description framework [RDF]) that enable richer descriptions of concepts and their relationships and provide more powerful means of describing and automating mappings between different conceptual models.[4]

Another direction from which the issues of federating and annotating information have been tackled is from the perspective of database management systems. There is significant activity (particularly in the bioinformatics community), in capturing both the processing history of data stored in community databases and the growing understanding of the community through annotations, which has strong parallels in motivation and in the technical directions being pursued.[5] Data Grid implementations, such as the Storage Resource Broker, which target large archives, are also evolving in similar directions.[6]

Design

One of SAM's primary design goals was to avoid imposing schema and an overall system architecture on researchers that they then must implement across all applications. Rather, SAM adapts to work with existing architectures and provides incremental benefits. We envisioned SAM as a tool that could be lightweight enough to use simply to share data or to implement as an EN. At the same time, it should have enough power to allow such a use to be incrementally extended to give a federated view of data in a portal or problem-solving environment. Ultimately, the use scenario could be extended to implement a comprehensive records system. We incorporated this philosophy into our requirements analysis, which led to specific design goals:

- SAM should allow data to be stored in its native format and avoid or delay the costs of developing a self-documenting or standardized representation.
- SAM should be able to capture the metadata, provenance, and annotation information in files and present it in a federated view with information from other applications and manual entry.
- SAM should be able to store arbitrary metadata about data stored in underlying structured stores.
- Using SAM should not preclude direct access to underlying data stores.
- Third parties should be able to define the metadata schema and data formats they wished to use independent of those chosen by the data–metadata providers.
- Applications should be able to ignore information outside their schema or to dynamically discover all information that has been federated.
- Third-party tools should be able to monitor activity in SAM and use it to trigger their workflows.

In essence, these requirements argue for minimizing the design-time coupling of scientific applications with community data–metadata systems and consumers of data–metadata (provenance tracking systems, other scientific applications, and so on), which we believe in turn argues for a "configurable middleware" approach. Figure 1 shows a high-level view of this model, with SAM providing a layered set of services between independent applications and data stores. These layers build on each other, with metadata being the most basic:

- Metadata management services provide base mechanisms for generating, federating, and translating annotation and relationship metadata.
- Semantic services provide support for advanced searching, relationship browsing, and pattern recognition.

- Notebook services provide mechanisms related to records management collections, digital signatures and time stamps, pagination, annotation display mechanisms, and so on.

Implementation

SAM's primary client-side interface is the *Web-based distributed authoring and versioning* protocol. WebDAV adopts the Web's HTTP model of resources accessed via a URL, adds standard methods for creating new collections (directories) and resources, and defines new functionality for adding and querying arbitrary string or XML key–value properties associated with each resource.[7,8]

WebDAV is an Internet Engineering Task Force standard extension to HTTP that, like Web services, uses XML to encode the payload of service requests. It was originally designed to support collaborative authoring, but DAV "documents" are not restricted to text-oriented formats and should be considered analogous to files or binary large objects. Extensions to WebDAV, such as DAV Searching and Locating (DASL), Advanced Collections, and Versioning, which are currently in development, promise additional relevant capabilities (www.webdav.org/specs).

WebDAV is quickly gaining popularity in the Web industry. A variety of WebDAV-capable Web servers (for example, from Apache Software Foundation, IBM, and Microsoft), clients (such as MS Office), and content management systems (Tamino and Oracle) are currently on the market, and development kits are available for Java, C++, and Python. There are also utilities that make WebDAV servers appear as shared network file systems, enabling non-WebDAV-aware applications to directly access shared data.[9]

We chose the Jakarta Project's Slide content management system (jakarta.apache.org/slide) and WebDAV implementation as the starting point for SAM development. Slide is written in Java and implements WebDAV as a servlet that calls an underlying Slide engine. We chose Slide because of its combination of a relatively complete WebDAV implementation, open-source license, ongoing development, and useful internal interfaces.

During the past two years, we have made modifications to Slide to implement SAM functionality, primarily in the metadata management and notebook services layers.

Security

Slide implements a username and password-based authentication mechanism and access control list-based authorization. We've added interfaces to let Slide use external authentication and authorization mechanisms so that SAM can function as middleware. Any Java Authentication and Authorization Service provider—an external username and password database, a Kerberos server, or public key certificate/Grid security infrastructure (www.globus.org/security/overview.html)—can be used to authenticate users. Any external service that implements a simple method requesting a yes–no response given the user's credential, the resource they are accessing, and the action they wish to perform, can be used to provide access control. Thus, access control is not limited to stored lists but could be based on policies such as certain users being limited to off-hours access or more sophisticated ones defined using systems such as Akenti or CAS, which let aspects of the authorization policy be delegated to stakeholders or community leaders.[10,11]

Datastore Federation

Slide can store WebDAV resources and properties in a variety of relational databases, assuming they've been configured to have the required tables. For SAM, we are extending this mechanism to support user-defined database schema and other data stores such as other WebDAV servers and Data Grid implementations. We have developed a mechanism to describe the mapping from requested WebDAV resources and properties to the underlying database schema to support, for example, making a molecular properties database available via WebDAV in such a way that community annotations that the database does not support directly can be added as new WebDAV properties. As in this example, properties that do not map to the underlying database can be stored in a secondary location, such as a database using Slide's standard schema.

Event Generation

To make activities in SAM visible to third-party software, we've modified Slide to produce Java Messaging Service events whenever the WebDAV resources are accessed or modified. We used the open source OpenJMS package for this purpose (openjms. sourceforge.net). SAM publishes notifications under two topics, one for WebDAV requests that do not change the stored data—for example, `GET`—and one for those that do, for example, `PUT` and `PROPATCH`. Each event contains details about the request to aid in filtering messages, but a subscribing application must access SAM directly (and have appropriate permissions to access the specified resource) to retrieve the viewed or modified content.

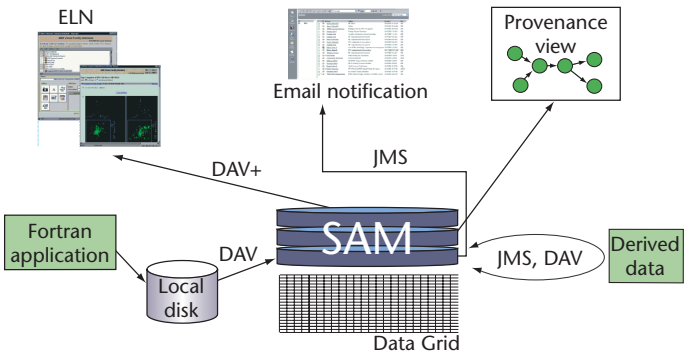

Figure 2. Scientific Annotation Middleware use scenario. Multiple applications contribute to the research record. Researchers can use SAM to generate an integrated view of the research process while also contributing to its automation.

Metadata Generation

Slide implements the basic WebDAV functionality for creating, modifying, retrieving, and deleting resources and properties. When people, applications, agents, and problem-solving environments create resources, Slide generates a few standard properties that describe the resource, such as its type, size, and creation date. SAM extends this mechanism to allow generation of user-specified properties. Those properties can be based on the contents of uploaded data, which could be in an arbitrary binary, ASCII, or XML format. The properties that should be generated are user-specified based on the data's Multipurpose Internet Mail Extension (MIME) type. MIME types are assigned using standard Web conventions—for example, based on the file name extension if not otherwise specified—with one addition. Because many different types of XML files are commonly given an .xml extension, users can configure SAM to run a user-defined XSLT script to determine a MIME type for .xml files, either by reading the MIME type from an element in the file or by inference based on the existence of specific elements.

Depending on the MIME type, SAM can then run registered scripts for that type to generate properties. For data in an XML format, a single XSLT script is run. For binary and ASCII data, two scripts are required. The first defines the details of the data format using the Binary Format Description (BFD) language. BFD, a Pacific Northwest National Laboratory–developed extension of the Extensible Scientific Interchange Language, lets a standard BFD engine extract the file's information into an XML format (www.scidac.org/SAM/bfd).[12] It allows specification of the data's low-level formatting, for example, `binary`, `little-endian`, `base64 encoded`, or `ASCII`, the layout of the data in terms of strings, integers, floats, and arrays of these primitives, and logic based on embedded parameters and flags. Based on this description, the BFD engine can generate XML output for subsequent XSLT processing to produce WebDAV properties. Thus, researchers can configure SAM to expose information from data submissions, such as the data files and parameters used to generate the submitted data or the chemical species the data relates to, in XML-formatted WebDAV properties without modifying the data itself.

Data Translations

Users can also register BFD and XSLT scripts to define translations of data that should be made available. SAM exposes the list of translations available for a given data set by generating a *hastranslations* property that gives information about the output format available, the translator used, and the WebDAV URL where anyone can retrieve or copy the translation. Because we anticipate the translated data sets will be much larger than individual properties, SAM currently creates translations dynamically when they are requested, versus generating and storing them during data submission. Consequently, it is possible to register new translations for existing data; the next request to read the hastranslations property will show the new formats available.

Copy Tracking

The WebDAV protocol defines a `COPY` operation to generate a new resource at a specified URL from an existing one without bringing the data to the local computer. To help track the data's origin, SAM adds a *source property*, which becomes part of the pedigree of the resource, to the new copy, which links it to the original URL. Thus, in circumstances where a research makes a personal copy of a shared resource, SAM will help document the data's provenance.

Electronic Notebook

As an initial step in creating the notebook services layer, we implemented the functionality required to generate a notebook compatible with the publicly available Electronic Laboratory Notebook (ELN) client. The newest ELN 5.0 version improves the user interface's isolation from the server implementation's specifics and includes more administrative capabilities. It is compatible with both the original CGI-based ELN server and SAM. In the SAM-based notebook server, the chapters, pages,

and notes in the notebook are stored as WebDAV resources, and the chapter/page/note tree structure is stored as WebDAV properties associated with those resources. Thus, the notebook's structure and its contents are directly available to other WebDAV-enabled applications. Adding a data set to the notebook reduces either to creating a new WebDAV resource that is then directly accessible to other applications or to creating a property linking an existing resource with a particular page.

SAM's Usage

The capabilities described earlier are just those included in the initial release of SAM but, taken together, they give a good idea of the project's general philosophy and the types of interactions SAM is meant to enable. Figure 2 details how researchers can use SAM to generate an integrated view of the research process while also contributing to its automation. Using a SAM-based notebook, a researcher describes his or her planned experiment. When the researcher collects data using a commercial instrument control program, she saves it to a network drive, which is mapped to an underlying SAM instance. A quick selection in the notebook links the new data into the notebook, and it appears on the page rendered in an interactive graph. In parallel, SAM publishes a notification that the data has appeared. The notification triggers other configured utilities: one that send an email to project team members, and another that uses the new data to produce a derived data set, which is also saved to the SAM server. A project team member copies a translated version of the data into his or her workspace and proceeds with the next task in the plan. Given the final product's URL, anyone granted access to the project's work area can trace backward, following links in WebDAV properties to discover the input data and relevant notebook entries.

The Collaboratory for Multiscale Chemical Science project is currently implementing scenarios such as this (http://cmcs.ca.sandia.gov). CMCS is an effort (involving one of the authors) in DOE's National Collaboratories program that uses SAM as a component of a portal-centric system designed to facilitate collaboration, data exchange, and provenance tracking across multiple chemistry subdisciplines. Figure 3 shows one of several components of the CMCS portal that interact with SAM: a pedigree browser that lets researchers view a selected data set's provenance across multiple links. CMCS also uses SAM to send email alerts and to link chemistry applications, such as the WebDAV-aware Extensible Computational Chemistry Environment and new chemistry Web services into their system.

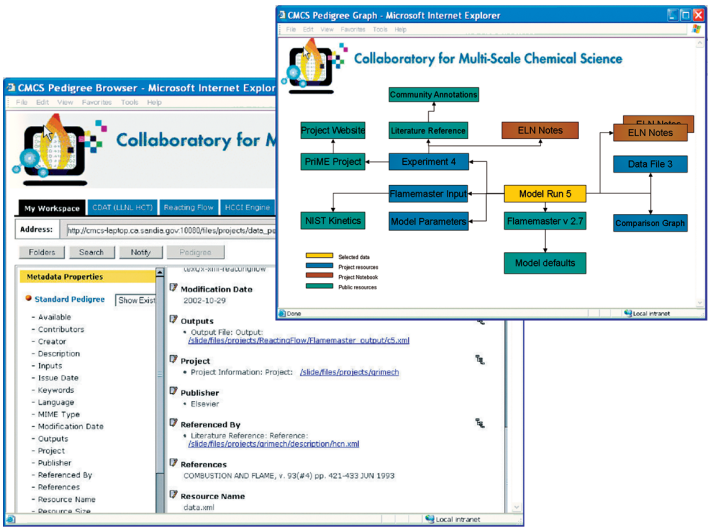

Figure 3. The Collaboratory for Multiscale Chemical Science (CMCS) Pedigree Browser Interface showing graphical and hypertext representations of a data set's provenance and other metadata.

Although SAM is already showing promise, we anticipate a variety of areas in which additional work will be needed to realize the full potential of this schema-less approach. While SAM's metadata generation and translation capabilities lower the barriers to federation, additional support for creating the required XML scripts (that is, some form of graphical tool for describing the desired mappings) will probably be needed. As more scientific software is exposed through Web services, we would like to extend SAM to let these services be registered as metadata generators and translators.

Even with these additions, the generality and power of the translation capabilities will be limited by XML's power. SAM's semantic services layer is designed to use more powerful languages developed in the Semantic Web community such as RDF and the Web Ontology Language to provide richer translation mechanisms. Analogous to the way SAM's XML-based translations can, once registered, be used by WebDAV-enabled applications that have no knowledge of SAM's added features, we hope to minimize the extent to which applications must understand semantic languages —for example, by making the results of a semantic inference available as an XML WebDAV property.

A final direction for SAM is to reexamine the concept of ENs. Systems such as the current ELN incorporate significant records-related functionality such as digital signatures and time stamps. However, their user interfaces and underlying functionality were not designed with the richer,

federated record that SAM enables. It will be important to deconstruct notebooks to let multiple applications access notebook functionality, at the level of services and to incorporate display elements such as a table of contents or page displays in their user interfaces.

We believe middleware such as SAM will enable composite scientific data management systems that can efficiently meet the needs of next-generation science and, after 500 years, supplant paper notebooks as the primary research record.

Acknowledgments

We acknowledge Elena Mendoza and Michael Peterson of Pacific Northwest National Laboratory's Computational Science and Mathematics Division for their respective contributions to the initial design discussions for SAM's metadata management services and for testing, debugging, and packaging ELN version 5.0. We also acknowledge the Collaboratory for Multiscale Chemical Science (CMCS) project team for extremely helpful feedback. The US Department of Energy supported this work through the DOE2000 program. Employees of Battelle Memorial Institute, which operates Pacific Northwest National Laboratory for the US Department of Energy under Contract DE-AC06-76RL0 1830 and Oak Ridge National Laboratory under Contract DE-AC05-84OR21400, wrote this manuscript.

References

1. J. Myers, E. Mendoza, and B. Hoopes, "A Collaborative Electronic Notebook," *Proc. IASTED Int'l Conf. Internet and Multimedia Systems and Applications* (IMSA 01), ACTA Press, 2001, pp. 334–338; www.emsl.pnl.gov:2080/docs/collab/presentations/papers/ELN.IMSA.final.pdf.
2. E.S. Mendoza et al., "EMSL's Electronic Laboratory Notebook," *Proc. WebNet '98 World Conference*, AACE Press, 1998.
3. J.D. Myers, "Collaborative Electronic Notebooks as Electronic Records: Design Issues for the Secure Electronic Laboratory Notebook (ELN)," *Proc. 2003 Western MultiConf.*, The Society for Modeling and simulation Int'l, 2003, pp. 13–22.
4. T. Berners-Lee, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*, Harper, 1999.
5. C. Goble, "The Grid: From Concept to Reality in Distributed Computing," *Bioinformatics World*, Oct. 2002, www.bioinformaticsworld.info/feature3b.html.
6. A. Rajasekar, M. Wan, and R. Moore, "MySRB & SRB—Components of a Data Grid," *Proc. 11th Int'l Symp. High Performance Distributed Computing* (HPDC-11), IEEE CS Press, 2002, pp. 301–310.
7. Y. Goland et al., "HTTP Extensions for Distributed Authoring—WEBDAV," RFC 2518, Feb. 1999, http://andrew2.andrew.cmu.edu/rfc/rfc2518.html.
8. R.T. Fielding, "Web-Based Development of Complex Information Products," *Comm. ACM*, vol. 41, no. 8, Aug. 1998, pp. 84–92.
9. *WebDrive*, South River Technologies, www.webdrive.com.
10. M.R. Thompson et al., "Authorization Policy in a PKI Environment,", *Proc. 1st Ann. NIST Workshop on PKI*, NIST, 2002, pp. 149–161.
11. L. Pearlman, "A Community Authorization Service for Group Collaboration," *Proc. IEEE 3rd Int'l Workshop on Policies for Distributed Systems and Networks*, IEEE CS Press, 2002, pp. 50–59; www.globus. org/research/papers.html#CAS-2002.
12. K. Blackburn et al., "XSIL: Extensible Scientific Interchange Language," *Proc. 7th Int'l Conf. High-Performance Computing and Networking*, Springer, 1999, pp. 513–524.
13. I. Foster, "The Grid: A New Infrastructure for 21st Century Science," *Physics Today*, no. 55, 2002, pp. 42–50.
14. D.E. Atkins, "Revolutionizing Science and Engineering through CyberInfrastructure: Report of the National Science Foundation Blue Ribbon Advisory Panel on CyberInfrastructure," 2003, www.cise.nsf.gov/evnt/reports/toc.htm.

James D. Myers is the lead investigator on the US Department of Energy-sponsored Scientific Annotation Middleware project and is the chief technical officer for the DOE sponsored Collaboratory for Multiscale Chemical Science project. He is also a chief scientist in the Computational Science and Mathematics Department at Pacific Northwest National Laboratory. He received his BA in physics from Cornell University and his PhD in chemistry from the University of California at Berkeley. He is a member of the IEEE Computer Society and the ACM. Contact him at jimmyers@pnl.gov.

Al Geist is a senior research scientist at Oak Ridge National Laboratory and leads the Computer Science Research Group. His interests include solar energy, materials science, biology, solid state physics, numerical linear algebra, parallel computing, scientific computation, and distributed computing. Contact him at gst@ornl.gov; www.csm.ornl.gov/~geist.

Jens Schwidder is a research and development assistant at Oak Ridge National Lab. His current primary focus is the security framework for the SAM project. He received an MSc in parallel and scientific computation from the University of Liverpool, England, and a Diplom.-Ing. (FH) in computer systems from the FHTW in Berlin, Germany. Contact him at schwidderj@ornl.gov.

Matthew S. Elder is a scientist at Pacific Northwest National Laboratory, where he has worked for the past 2 years developing data integration software. He is currently focused on integrating existing scientific data with the Scientific Annotation Middleware. He received his BS in computer science from Western Washington University, and is currently working on his MS at Washington State University. Contact him at Matthew.Elder@pnl.gov.

Alan R. Chappell received a BS in mechanical engineering and an MS in operations research from North Carolina State University. He is a senior development engineer with the National Security Directorate at Pacific Northwest National Laboratory. He works primarily in human-centered approaches to knowledge management and systems for information exploitation. Contact him at Alan.Chappell@pnl.gov.