# Structure, sharing and preservation of scientific experiment data

Sangmi Lee Pallickara, Beth Plale, Scott Jensen, Yiming Sun
*Department of Computer Science, Indiana University*
*{leesangm, plale, scjensen, yimsun}@cs.indiana.edu*

## Abstract

*In mesoscale meteorology, the quantity of information has increased significantly due to sophisticated data distribution schemes combined with developments in sensors and instruments capable of monitoring the lower several kilometers of the atmosphere at higher levels of resolution. This paper introduces myLEAD, a personalized information management tool for geoscience users. MyLEAD eases the data and information overload on the scientist by providing explicit solutions to the problems of structure, sharing, and preservation. This paper describes strategies within the myLEAD system to personalize data product and representation which ultimately leads to personalized workspaces and collaborative environments. We also include experimental results from some of the experiments that we conducted.*

**Keywords:** Scientific Computing, Metadata, Grid Services, LEAD, Experiment data, OGSA-DAI

## 1. Introduction

Managing the vast quantity of data and information used during the course of computational science investigations has become unmanageable for the individual scientist as the computational capability of the models with which he works increase in size and power. The meteorologists whom with we work, for instance, are striving towards on-demand weather forecasting, which draws on data from dozens of observational sources to generate what the meteorologists call "ensemble runs," that is, runs that consist of 500 or more instances of the model run simultaneously. We describe in this paper our solution to easing the data and information overload on the scientist by explicit solutions to the problems of structure, sharing, and versioning of a scientist's information and data products. At the core of our solution is a metadata database for storing metadata about data products.

A scientist, in the process of carrying out a single computational experiment, can touch hundreds to thousands of data products. The products might include the specific input and output files used by the computational model, configuration files, log files, notebook annotations, consulted web pages, related papers. Aspects of managing this kind of data have been investigated. For instance, annotation systems document the investigative process and are intended to replace a scientist's notebook [1],[2]. Researchers have also investigated information-based systems that draw connections to related literature [3]. Our system instead addresses the soaring number of data products used in and generated by computational scientists during their investigations. As models grow in complexity (nesting a local phenomena model within a global phenomena model) and in scale (in response to grid computing's potential to vastly increase resources available to users), manual management of data products is rapidly pushing beyond what an individual can cope with on his or her own. While our work incorporates some of the aspects of the annotation system, its primary focus is automated, scalable, and distributed metadata management.

The amount of data available to a computational scientist is overwhelming. Particularly in mesoscale meteorology, where researchers investigate mesoscale weather phenomena such as flash floods and tornados, hundreds of sensors and instruments continuously gather atmospheric conditions. The number of sources will increase significantly as small radars of a size suitable for mounting on a cell phone tower are deployed to augment data collections in particular local regions. We view this plethora of data as a vast information space that a researcher must make sense of in order to carry out next generation forecasting. The information space is analagous to the Internet. The number of web pages making up the Internet is far too vast for any individual to grasp as a whole. Users of the Internet are comfortable in the vast space, however, because tools exist for tagging, searching, and publishing. But these tools are poorly suited to managing scientific data. The Internet's data sharing model is written by a single source, read-only by many. Tagging a page is no more than a placeholder for a later read. But scientific data products are frequently manipulated, and the resulting product taking on a unique identity. Further, the search and download model is not well suited to a common operation in computational investigation, that is, of connecting a model to the data products that it needs in order to run. We see a data management solution in computational science as being guided by the following requirements. Scientists

- want total control over their data products
- want the ability to share products but retain control over what gets shared, and with whom,
- want rich search criteria over the vast information space but don't necessarily want to write SQL queries.
- need help managing experiment products generated over an extended period of time (i.e., years),
- want high level of reliability – data must always be accessible, and
- want the ability to work locally.

The model we propose for managing scientific data products is the personal metadata catalog. The **personal metadata catalog** is a collection of descriptions of the digital products that are of value to a particular scientist during the course of a computational experiment. The catalog is organized around the notion of the *experiment*, loosely defined as the investigation of mesoscale weather phenomena. It can involve any number of model runs and be carried out over multiple days. A scientist's personal metadata catalog is managed by a service called myLEAD[4], which manages many instances of personal catalogues simultaneously. The myLEAD service is implemented across distributed and replicated databases augmented with additional functionality including a web service interface so that it can interact with other services on a computational grid. The tool is being built as part of the Linked Environments for Atmospheric Discover (LEAD)[5], a research project into building cyberinfrastructure to advance mesoscale meteorology.

The focus of this paper is on the capability built on top of the database to satisfy the user requirements: structure, sharing, and preservation. Relational databases, upon which the personal catalog is built, are fully structured. But we do want to hide as much of the structure as possible to ease the use of the catalog, particularly in storing products to the catalog and in querying for them later. Sharing of data products is vital, but so is the need for a scientist to completely trust that products managed by the catalog remain secure. Enabling different levels of sharing while maintaining trust is an important goal. A scientist may work within the myLEAD data space for 5 years – during the course of her PhD studies for instance. The metadata catalog must preserve old experiments, and save important versions of experiments for an extended period of time. Structure, sharing, and preservation are orthogonal concepts that form a 3-dimensional space. All data products in reside at a point in this 3-dimensional space. Catalog performance is optimized for the kinds of access that we expect in the scientific investigation environment. Early performance results demonstrate that there is still room for improvement.

This paper is organized as follows. Section 2 describes the architecture of the myLEAD service. The structure, sharing, and preservation capabilities of the system are described in Section 3. In Section 4 we give baseline performance of the system. Related work appears in Section 5. The paper concludes with future work in Section 6.

## 2. Architecture

myLEAD consists of a set of distributed services. A service instance resides at each site in the LEAD testbed. Each of the five sites in the LEAD testbed will run a persisent server-side service, shown at the bottom of Figure 1, and client-side service, shown in the middle of Figure 1. Portal access is through a single portal. Users local to a site will have their "personal metadata catalog" managed by the myLEAD service at that site. Users, then, are partitioned across the sites. One site will run a master instance of the myLEAD service. This master instance serves as a replica to all satellite sites. Replication to the master will occur on a regular basis. Our current plan is for nightly updates.

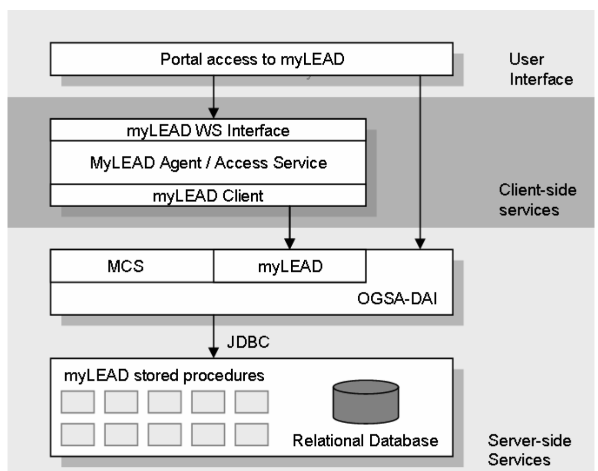


**Figure 1: architecture and component interaction**

The server-side service instance is a persistent Grid service built on top of a relational database. It extends the Globus Toolkit MCS[6] and OGSA-DAI[7]. MyLEAD extends the database schema with support for spatial and temporal attributes, and organizes the data around the notion of "experiments". These are supported by adding methods for database access and making performance improvements over MCS by means of database stored procedures. Eventually, replicated myLEAD servers will be distributed across the LEAD test bed, achieving reliability through a master-client replication scheme.

The myLEAD agent service is a transient, short-lived service that serves a single user for a single session. Meanwhile, the access service is a permanent Web service accessible from other Web services directly. Both types of client-side services manage stateful interactions between the user and the server. At the user-interface level, a client (user) interacts with myLEAD through the LEAD portal service. The tool provides several portlets for managing, browsing and searching personal information spaces.

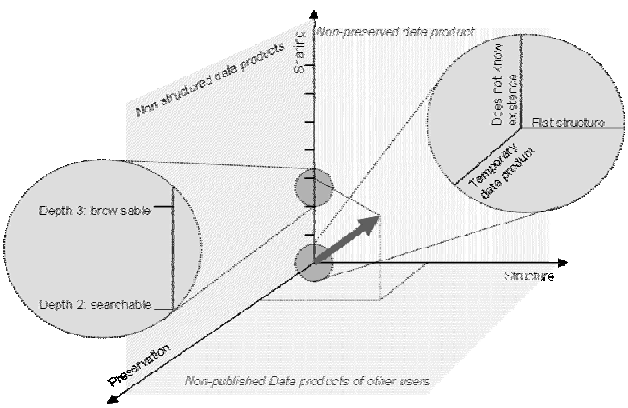


**Figure 2: three-dimensional space defined by structure, sharing, and preservation**

## 3. Structure, Sharing and Preservation

Structure, sharing, and preservation are orthogonal concepts that can be viewed as forming a 3-dimensional space in which all data products reside. As illustrated in Figure 2, sharing is on the Y-axis, structure is on the X-axis, and preservation is on the Z-axis. The origin is blown up in the bubble in the upper right of the figure. At the origin on the structure axis is flat structure that increases in complexity as one moves away from the origin. Sharing at the origin is zero sharing. Products are not visible, and outsiders are not aware of the existence of a product. Sharing increases incrementally as one moves away from the origin. The preservation axis is time based. At the origin are timely, temporary data products. Products age as they move away from the origin. The surface defined by the X- and Y-axes – depth of structure and depth of sharing— identifies data sets which have not been preserved. For instance, a product still may be actively involved in a computation, or may have already been discarded. Similarly, the Z- and Y-axes, depth of time and depth of sharing, define a surface of datasets which are not organized in a user's hierarchical structure, such as the data product in the public storage area.

### 3.1 Structure

During discussions with users about the personal metadata catalog, it became apparent that scientists want scalable storage, fast queries, and rich search criteria but do not want to have to understand the nuances of a fully-structured relational database when writing queries. By storing a metadata description of a product as a first-class entity, one has more freedom to define richer metadata over which a product is searched. Of course, populating a metadata description is a difficult problem. We are addressing this problem, but it falls outside the scope of this paper. Realistically, we cannot expect scientists to remember the details of how metadata descriptions are stored when querying the catalog. That is, the structure should to a large degree be transparent. The challenge is to provide structure over the vast space without forcing the user to know the full structure of the space.

Structural transparency means that a distributed system should hide its distributed nature from its users, while appearing and functioning as a normal centralized system[8]. The structural transparency provides a customized user view of the experimental data product to each client; therefore the users do not need to be aware of how a resource is physically located and organized. The depth of the structure is to some degree under the control of the users and to some degree under the control of myLEAD. MyLEAD provides a set of fundamental elements to structure the workspace, such as experiments, and collections. However, users can utilize the basic structure flexibly. For example, an experiment can comprise multiple collections, and collection can comprise multiple collections. The depth of structure is difficult to define in a general way. Hence, we define the depth of the structure as the depth of a particular data product in the hierarchy of the user's workspace.

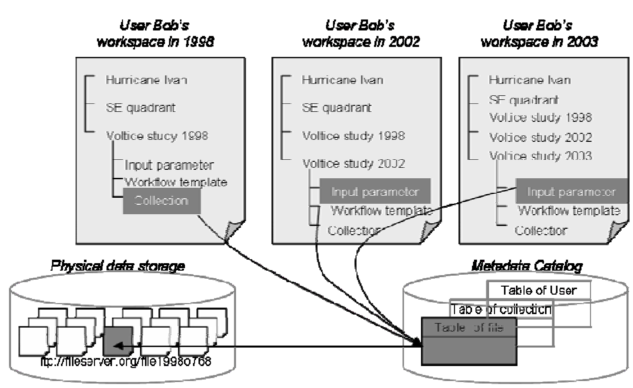


**Figure 3. Personalized view with structural transparency**

As depicted in Figure 3, the physical data product is stored in the long-term mass storage. During a computational experiment, the system generates metadata entities that point to the data products. Those metadata entities are stored in the myLEAD metadata catalog. The physical mass storage has its own storage structure, and the myLEAD metadata catalog has its own database table structure. However, users are provided a customized view so that they can access the data without any knowledge of the database or the physical storage structure. Moreover, users can organize their own customized working space with the data product and information about their preference.

## 3.2 Sharing

Sharing of the products of a computational investigation is an essential part of scientific research. MyLEAD facilitates sharing of data products between group members by means of “publishing”. A researcher can publish, or make visible to a larger group, a data product, collection of products, or an entire experiment to a particular group, including another individual. The product owner requires controlled levels of access to shared products, however. Because of the sensitive nature of the data products produced during computational science investigations, the act of publishing a data product must be completely under the control of the product’s owner. Additionally, we envision that half of the timely data products will remain unshared. myLEAD must also support the K-12 educational setting, where the instructor of the class shares materials with students and guides students through exercises. Some form of role-based access is needed because instructors require privileges beyond those given to students. Students, on the other hand, can share products within their working group. Finally, it is incumbent on the myLEAD service to provide a user interface that builds a scientist’s trust over when, where, and how publishing is being carried out.

A general purpose grid security infrastructure is inadequate for implementing publishing. The Grid Security Infrastructure (GSI)[9], based on PKI and SSL, contains tools for controlling access to the resources based on the user’s capability on that resource by means of the Community Access Service (CAS)[10]. However, CAS does not support fine-grained access control to data objects. Other solutions, such as the OGSA-DAI service[7] for grid-based database management services maps certificate-based access of GSI to the traditional table and row access control provided by the database management system. None of these approaches provides the level of sharing needed in myLEAD.

The depth of sharing represents the level of access that the owner of a data product gives to other users. Healey [11] defines five states of asynchronous collaboration with respect to grounding of artifacts. Our scale of data sharing is expanded to contain eight depths as follows:

- Depth-0: Participant (P) is unaware that experiment data product (E) owned by user (U) exists.
- Depth-1: P is aware that E exists,
- Depth-2: P can search E,
- Depth-3: P can browse the content of E,
- Depth-4: P can access E and its contents,
- Depth-5: P can remove and write E,
- Depth-6: P is aware that other participants are sharing E.
- Depth-7: P can delegate E to another participant (P’).

In depth-0 the experimental data product E owned by a user U is not visible to any group member P. We expect that the majority of data products will reside at depth-0. At depth-1, syntactic information about the product E is exposed, so user P can learn of the existence of E but not obtain the contents of its metadata. At depth-2 sharing, user P can obtain well-organized metadata about the data product, however P cannot retrieve the product itself. At depths-3, 4, and 5, user P can read the metadata, retrieve the product, and remove the product P. Depth 6 and 7 indicates more sophisticated control for data product E. This is useful for an administrator or higher role participant such as the instructor of a class. Depth 7 is the most complete data sharing state of collaboration. Generally, the owner or creator of the data product can have the sharing state of depth-7.

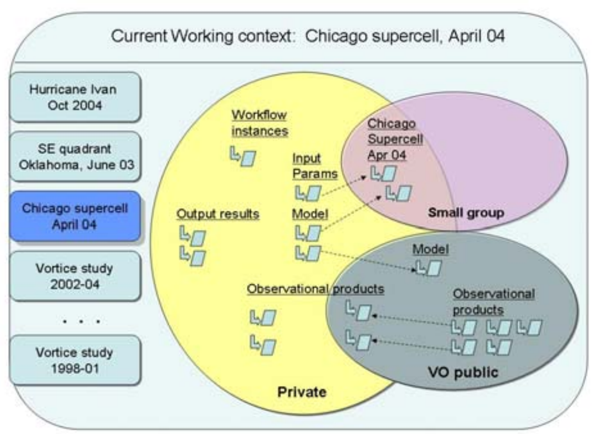


**Figure 4: user interface to information space showing current experimental context and levels of sharing of various data products**

**Building trust.** In order to build a sense of trust in the scientist that the myLEAD metadata catalog can protect data products from the prying eyes of others, it is important that publishing of data products be carried out from a single "place" in the portal. Figure 4 illustrates an example of how trust might be conveyed. The figure illustrates a set of contexts down the left side of the diagram. As can be seen, we associate context with a current working experiment, in this case the Chicago supercell of Apr 2004. Within this context the user is given a graphical view of the products that are private (within the large oval), shared with a small group (upper right hand oval), and those that are visible to the world (lower right oval.) In this view, the act of publishing can be viewed as a drag-and-drop operation on a specific data product or collection of products. By restricting publishing to this screen only, and giving visible clues that the user is entering a trusted set of pages, we hope to convey to the user a sense of trust over the privacy of his/her products.

### 3.3 Preservation

Data preservation conveys multiple factors of understanding of data products and experiments along the axis of time. At its essence, the axis of preservation represents how an experiment ages in the data-centric scientific computing service. The unit of preservation in myLEAD is the "experiment". An experiment has structure that could be nested several layers deep, and contains a collection of data products We adopt the notion of the "landmark event" from the Elephant file system[12] for distinguishing when a new version of an experiment exists. A user can mark a version as a landmark or the system users heuristics to mark other version as landmark versions.

After a data product is created during the course of an investigation, it is often stored in a storage. A metadata catalog manages access to the storage repository to ensure consistency between the storage repository and the metadata associated with the data product. Accessibility to the data product should be guaranteed for duration that the user specified. A repository could be as simple as a local file system, but the solution we are investigating is the employment of a storage repository such as Replica Locator Service (RLS) [13], Storage Resource Broker (SRB) [14], or Storage Resource Manager (SRM) [15]. These repositories generally provide a grouping abstraction for storing files and provide location-transparent data storage and access by mapping a physical path name to a location-transparent logical name. Though the myLEAD metadata catalog could work with any of these repository tools, SRM and SRB tightly couple their own metadata catalog to their storage system.

## 4. Performance Evaluation

The purpose of this evaluation is to profile our initial effort in enabling user requirements: structure, sharing, and preservation. MyLEAD extends the Globus MCS by means of extending the schema by including support for spatial and temporal attributes. In our early evaluation, we fix the number of users to one and measure the cost for various types of queries. For the experiment, we have two major services of myLEAD: the myLEAD client, and the myLEAD server. The myLEAD client resides on a dual processor Dell PowerEdge 6400 Xeon server (700MHz PentiumIII), 2GB RAM, 100GB Raid 5, RedHat 7.2, JDK1.4.2. The myLEAD server is deployed on a dual processor 2.0 MHz Opterons, 16GB RAM, GENTOO Linux. The myLEAD server is built on top of the OGSA-DAI version 3.0, Globus MCS version 3.1 and provides access to the database platform, mySQL-version 5.0.0. The myLEAD client and the myLEAD server are interconnected through a 1 Gbps switched Ethernet LAN.

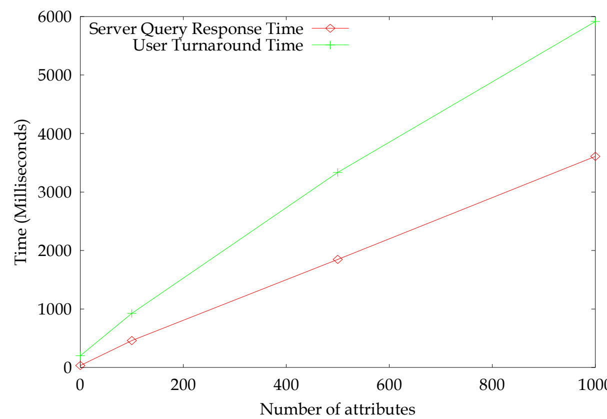


**Figure 5 (a): Increase of the attribute creation time by the increase of the number of attributes**

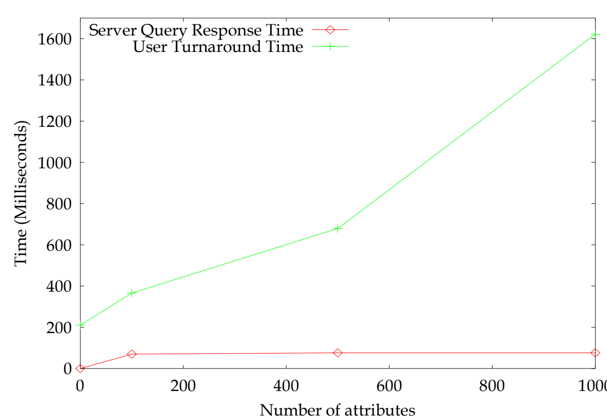


**Figure 5 (b): Increase of the attribute query time by the increase of the number of attributes**

Figure 5-(a) and (b) plots the *user turnaround time*, which is the cost for retrieving results from the myLEAD server under an increasing number of 'attributes'. Here the turnaround time is measured as the average of the request-responses between myLEAD client and myLEAD server. It includes two metrics to ease the analysis; *server query response* time and *query deployment time*. Server query response time is a measure of the amount of time it takes for a myLEAD server to complete a mySQL query request and return the result set. The query deployment time is the transit time from the user to the myLEAD server and vice versa. The query deployment time includes generating the OGSA-DAI document, parsing the document, and the network communication time between the myLEAD server and client.

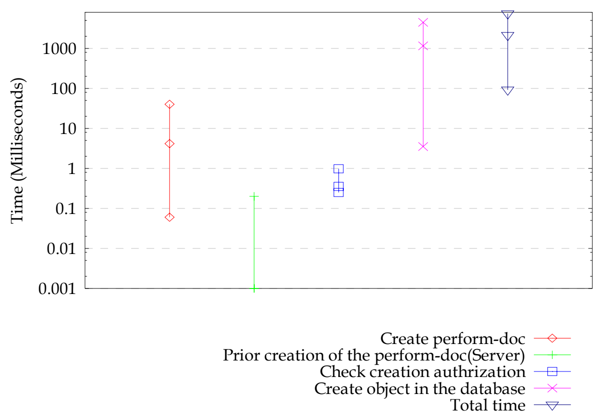


**Figure 6 (a): Partial cost of creating attribute in myLEAD**

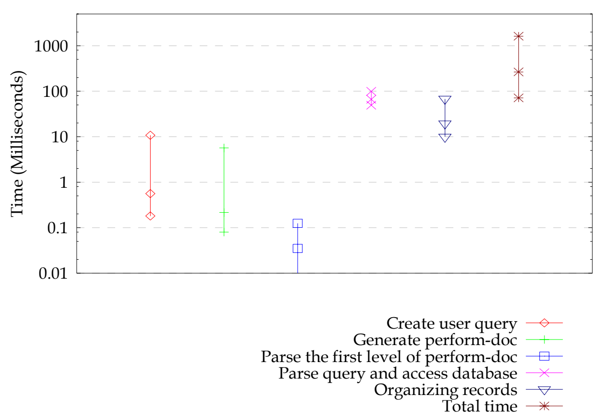


**Figure 6 (b): Partial cost of querying attribute in myLEAD**

The increase in creation time corresponding to the increase in the number of attributes was obvious. To measure this cost, we measured the time of creating a 'file' element which contains increasing number of attributes.

Similarly, there is an increase in the query time corresponding to an increase in the number of attributes which belong to a given 'file' element. It should be noted that the returned document contains the 'file' element matching the entire set of attributes. However, as shown in figure 5-(b), compared to the creation, the user turnaround time is almost not affected by the server query response time. The most significant factor in the increase of the user turnaround time is the increase in the query deployment time. It shows that the size of the returned query affects the performance significantly.

Figure 6-(a) and (b) presents the partial costs for major processes comprising the user turnaround time. We measured the performance with a set of simple queries having an increasing number of attributes. Each high-low line plots the range from the highest to the lowest average measurements in the set of the queries under measurement. Each average measurement is calculated as an average of 100 times measurements per query. Figure 6-(a) illustrates the attribute creation time with its partial costs. Each high-low line captures the cost of each process, which is varied by the types of query. From the left to right, the first high-low line plots the time which is measured during the step of enclosing the body of the OGSA-DAI perform-document within a valid XML document in the myLEAD client. The second line plots the time for creating an OGSA-DAI perform-document prior to processing the query in the myLEAD server. The third line represents the time to query the database through a stored procedure to see if the user should be allowed to create a new object. Since this is very simple query, this represents the minimum time needed on the myLEAD server. The fourth line plots the time for creating an object in the database.

Similarly, figure 6-(b) presents the attribute query time with partial costs. The first high-low line plots the time for creating a user query in the myLEAD client. The myLEAD client provides APIs to build an object that contains the query and specifies criteria at multiple levels. This measurement represents the time require to convert that object into the body of the perform-document. Next line represents the time needed to take the body of the perform document created and create a valid XML perform document by adding necessary information for the OGSA-DAI service. The third line represents the time needed to parse the first level of the perform-document. At this point, most of the query is still contained in a single string and passed to the next step. The fourth line plots the measurement of the procedure which parses the query and queries the database. The results are stored in a temporary table. The last line plots the time for organizing the record in a temporary table before constructing the perform-document to return a response to the user.  As can be seen, the performances of each process are plotted in a large range.

## 5. Related work

Other projects are exploring metadata management for scientific user communities working in a Web service environment. MyGrid[16] manages information for bioinformatics researchers using a native XML database. Similar to the myLEAD, myGrid gathers and stores information for a user while she conducts science experimentation. MyGrid Information Repository (IR) emphasizes support for handling textual documents, such as index support, because much bioinformatics information is textual. Meteorology forecasting is driven by real-time handling of large binary files. Hence, myLEAD is investigating automated metadata generation. The Network for Earthquake Engineering Simulation Grid (NEESgrid) metadata catalog [17] is built from Resource Description Framework (RDF). RDF's basic building block – the subject-predicate-object – supports the representation of complex relationships between entities at a very fine granularity. Fine granularity comes at a cost of query processing. myLEAD trades off the granularity for more efficient query processing achievable with SQL's strong theoretical foundation.

MySpace [18] is a tool developed for astronomy researchers to manage the large federation of data archives the community shares. It creates "swatches" (shared spaces that cross multiple file systems) shared by a community in the data archives for cached and persistent data and provides common query access over the cached and persistent space, both of which are organized as file systems. Because myLEAD is oriented toward rich application metadata, a file system solution is not viable.

## 6. Conclusion and Future Work

myLEAD is scheduled for release April 2005. The advanced functionality, of which the foundation is established in this paper, forms our ongoing work. Immutable experiments allow users to reuse experiments without destroying the integrity of the earlier project. The user must have complete control over the mechanisms for publishing data products to the larger community. The system must also convey visual cues that build the user's trust— perhaps by guiding the user through a set of Web pages in a portlet that conveys a sense of entering a secure space. Publishing of data products would then be allowed only within that secure space.

### References

[1] Sriram Krishnan, Randall Bramley, Dennis Gannon, Rachana Ananthakrishnan, Madhusudhan Govindaraju, Aleksander Slominski, Yogesh Simmhan, Jay Alameda, Richard Alkire, Timothy Drews, and Eric Webb. The XCAT Science Portal. In *Scientific Programming*, volume 10(4), pages 303-317, 2002

[2] Camon E., Barrell D., Lee V., Dimmer E., Apweiler R. The Gene Ontology Annotation (GOA) Database - An integrated resource of GO annotations to the UniProt Knowledgebase. In *Silico Biology* 4(1): 5-6 (2004).

[3] Katy Börner, Richie Hazlewood & Sy-Miaw Lin: Visualizing the Spatial and Temporal Distribution of User Interaction Data Collected in Three-Dimensional Virtual Worlds. In *Sixth International Conference on Information Visualization*, London, England, July 10-12, 2002, IEEE Press, pp. 25-31.

[4] Beth Plale, Jay Alameda, Bob Wilhelmson, Dennis Gannon, Shawn Hampton, Al Rossi, and Kelvin Droegemeier User-oriented Active Management of Scientific Data with myLEAD In *IEEE Internet Computing special issue on Internet Access to Scientific Data*, Vol. 9, No. 1, Jan/Feb 2005, pp. 27-34.

[5] Kelvin K. Droegemeier, V. Chandrasekar, Richard Clark, Dennis Gannon, Sara Graves, Everette Joseph, Mohan Ramamurthy, Robert Wilhelmson, Keith Brewster, Ben Domenico, Theresa Leyton, Vernon Morris, Donald Murray, Beth Plale, Rahul Ramachandran, Daniel Reed, John Rushing, Daniel Weber, Anne Wilson, Ming Xue, Sepideh Yalda, Linked environments for atmospheric discovery (LEAD): Architecture, Technology Roadmap and Deployment Strategy, *To appear 21st Conf. on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, January 2005

[6] G.Singh et al. A Metadata Catalog Service for Data-Intensive Applications. In *ACM/IEEE Supercomputing* 2003, IEEE CS Press, 2003, pp.33—49

[7] OGSA-DAI Open Grid Services Architecture Data-Access and Integration http://ogsadai.org.uk

[8] G. Coulouris, J. Dollimore. And T. Kindberg, *Distributed Systems Concepts and Design*, 3rd Edition, Addison-Wesley, 2001

[9] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. A Security Architecture for Computational Grids**.**. In *Proc. 5th ACM Conference on Computer and Communications Security Conference*, pp. 83-92, 1998.

[10] L. Pearlman, V. Welch, I. Foster, C. Kesselman, S. Tuecke. A Community Authorization Service for Group Collaboration. In *Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, 2002.

[11] Patrick Healey & N. Bryan-Kinns, Analysing asynchronous collaboration in People and Computers XIV - Usability or Else! In *Proceedings of HCI 2000*. pp. 239-254, 2000

[12] D. S. Santry, M.J. Feeley, N.C. Hutchinson, A.C. Veitch, R. W. Carton, and J. Ofir. Deciding when to forget in the elephant file system. In *Proceedings of the 17th ACM Symposium on Operating Systems Principles*, pp. 110-123, Dec. 1999.

[13] S. Vazhkudai, S. Tuecke, I. Foster. Replica Selection in the Globus Data Grid. In *Proceedings of the First IEEE/ACM International Conference on Cluster Computing and the Grid (CCGRID 2001)*, pp. 106-113, IEEE Computer Society Press, May 2001.

[14] SRB Storage Resource Broker, http://www.sdsc.edu/srb/

[15] A. Shoshani, A. Sim, and J. Gu, Storage Resource Managers: Middleware Components for Grid Storage, In *Proc. of IEEE Conference on Mass Storage Systems and Technologies* (MMS) 2002

[16] R.D. Stevens, A.J. Robinson, and C.A. Goble, "MyGrid:Personalized Bioinformatics on the information Grid," In *Proc. of 11th Int'l Conf. Intelligent Systems for Molecular Biology*, Int'l Soc. For Computational Biology, 2003

[17] J. Peng and K. Law, Reference NEESgrid Data Model, Technical Report, Network for Earthquake Engineering Simulation Grid (NEESgrid). NEESgrid-2004040, 2004

[18] C.L.Qin et al.,Myspace: Personalized Work Space in Astrogrid. In *Proc. of UK e-Science All Hands Meeting*, University of Southhampton, 2003, pp. 361-365; www.allhands.org/uk/2003/proceedings.