A Novel Event-Based Consistency Model for Supporting Collaborative Cyberinfrastructure Based Scientific Research

Ahmet Fatih Mustacoglu[1,2], Ahmet E. Topcu[1,2], Aurel Cami[3], Geoffrey Fox[1,2]
[1]*Community Grids Lab, Indiana University, Bloomington, IN, 47404, USA*
[2]*Department of Computer Science, Indiana University*
[3]*Department of Biomedical Informatics, University of Pittsburgh*
{amustaco, atopcu, gcf}@cs.indiana.edu, camiau@cbmi.pitt.edu

ABSTRACT

Grid computing is an emerging technology one of whose main focuses is the coordinated resource sharing among dynamic groups of individuals, institutions and resources. In collaborative Grid environments, data and system consistency play a crucial role for the correct functioning of the system. In this paper we propose a novel event-based consistency model and discuss the implementation and integration of this model with Semantic Research Grid (SRG)—a collaborative system consisting of tools and services for supporting Cyberinfrastructure based scientific research.

KEYWORDS: Events, Grid, collaboration, authentication and authorization, Cyberinfrastructure based research, annotation, academic search.

1. Introduction

The Grid concept is closely related to the "flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources" [1]. Grid systems provide collaborative environments where people work together and share a set of resources to achieve common goals. Various heterogeneous resources comprise the Grid, and it is crucial that they communicate in a well-known and consistent way. The Open Grid Services Architecture (OGSA) defines open standards for these communications aiming to leverage the interoperability among resources coming from different sources. In the OGSA service model everything is represented as a Grid service, which is a Web service with minor extensions [2]. Multiple users can collaboratively create and manage a digital object, which is a content object in a digital environment, in Semantic Research Grid (SRG) system. The SRG system aims to provide "a community-centric platform of tools and services that integrate the major existing annotation tools, academic search tools, and scientific databases into the Cyberinfrastructure based scholarly research" [3]. To advance the level of interoperability, we have designed and implemented SRG system, which can also be deployed as a Grid service, as a Web service technology. Due to the nature of the shared resources in Grid and collaborative environments, collaborative systems face significant challenges in providing overall system consistency. One key challenge is the maintenance of the data consistency within the system. In the SRG system, Digital Objects (DOs) are created and managed by the multiple users. A DO is a collection of metadata for a citation in a digital environment. The Dublin Core Metadata Initiative (DCMI) is an organization dedicated to promoting the widespread adoption of interoperable metadata standards and developing specialized metadata vocabularies for describing resources that enable more intelligent information discovery systems" [4], and in SRG system we represent our DO by storing various metadata fields such as title, URL, authors, publication type, chapter, edition, editor, journal, publisher, volume, issue, conference name, conference URL, conference location, conference date, etc. In the SRG system multiple users can collect data and metadata from various sources and this may lead to undesired data corruption that may occur in the system. For instance, one user can create a DO by collecting metadata from Google Scholar, and a second user obtaining the metadata for the same DO from a scientific database can overwrite the first user's digital object metadata, leading to the deletion of user defined tags in the DO.

Our approach for solving the data consistency problem for the SRG system is based on the notion of *event* and *dataset*. In our proposed framework, event is a time-stamped action on a DO, and each dataset represents certain states of the system. If a user would like those changes to take effect on the digital object, then the user needs to select the minor events, which represents the desired changes to the digital object, to create a dataset. In the process of getting the most recent digital object metadata, datasets are applied on top of the initial metadata of the digital object.

Event-based mechanisms have been previously used in

several systems, such as in the processing of digital media [5], in the representation of collaborative activities in an intensive care unit [6], in heterogeneous media-event processing systems [7], in annotating XML documents for remote sharing [8], and in a content management system [9].

This paper discusses the event-based consistency model for the SRG system and expounds its implementation and integration with SRG. The rest of this paper is organized as follows: Section 2 provides a discussion of the SRG system. Section 3 gives the details of authentication and authorization mechanism for SRG system. Section 4 explains the event-based consistency model for the SRG system. Section 5 describes the creation of datasets and application to the DO metadata. Section 6 explains history of a DO and the rollback mechanism. Section 7 gives a discussion of the future work.

2. SRG SYSTEM

The SRG provides a collaborative Cyberinfrastructure based scientific research environment. Its tools and services are backed by databases which store user and community specific data and metadata and are configured into three applications: (1) A model for scientific research which links both traditional simulations and observational analysis to the data mining of existing scientific documents; (2) A model for a journal web site supporting both readers and the editorial function; (3) A model for a natural collection of related documents such as those of a research group or those of a conference [3].

The SRG system is composed of the following modules: (A) Session and Event Management; (B) Digital Object Management; (C) Annotation Tools; (D) Search Tools; (E) Authentication and Authorization (explained in section 3); (F) Other. Next, we give a brief description of the functionality provided by each module. A detailed description of the implementation of these modules may be found in [3].

2.1 Session and Event Management Module

The goal of this module is to store user specific data such as user authentication credentials, minor events (described in section 4) to a DO, and the selected view option.

A session is a user's state information, and maintained on the server side [10]. From the moment user logged in the SRG system, authentication credentials, any changes made to a DO, and view options for metadata fields of a DO are all saved in the user session. When a user logs out from the SRG system, all unused minor events (modifications to a DO) for a dataset creation are removed from the current user's session. View options provide selections to show or hide the selected metadata fields of a DO.

2.2 Digital Object Management Module

This module: (1) allows the user to insert a DO, which represents metadata content of an object in a digital environment, manually into one of the local/remote SRG databases; (2) Integrates PubsOnline software—"an open source tool for management and presentation of databases of citations via the Web" [11]—into the SRG system. It provides a search mechanism for local/remote databases of SRG. (3) Provides access to the history of a DO, from its entry into SRG system to present; (4) Allows a user to view detailed information about a DO; (5) Allows a user to update any metadata fields of a DO, which is saved into session as a minor event for this DO; (6) Provides an interface to perform basic and advanced search in the local database (the two search modes are similar, except that the latter allows the user to perform a more refined search).

2.3 Annotation Tools Module

This module provides an interface to the annotation tools: Delicious [12], CiteULike [13], and Connotea [14]. It allows a user: (1) to upload DOs data and metadata to one of these annotation websites; (2) to download DOs data and metadata from one of the annotation websites into one of the local/remote SRG system databases; (3) to transfer DOs data and metadata between these annotation websites.

2.4 Search Tools Module

This module implements a mechanism to make a search by keyword using Google Scholar or Windows Live Academic search tools. Search results will be populated and displayed using interface implemented in the system. If user has a write access for any databases in the system, selected DOs can be stored into the system for specific user or group. If user already has same DO in selected database, he/she has option to update DO using event-base mechanism defined in section 4.

2.5 Other Modules

Several other modules are part of the SRG system. These include: (1) User registration module; (2) Username and password recovery module; (3) User's profile management module, where a user can modify personal information, change system password, and request subscription to available SRG system groups; (4) DO metadata view options module, which provides

options for each metadata fields of DOs to display/hide them in the SRG system.

3. AUTHENTICATION AND AUTHORIZATION

In collaboration systems users may access other user's private data unless a protection mechanism is incorporated in the system [15]. We have designed and implemented an authorization module for the SRG system in order to protect the databases and DOs. This module provides mechanisms for: (1) Login management; (2) Logout management; (3) Digital object and database(s) access rights management; (4) Administrative tools.

Different kind of *access control methods* have been previously used in various systems. These methods include the access-control matrix, the role-based access control (RBAC), and the task-based access control (TBCA) [15]. We have adopted the access-control matrix model with the addition of supporting multiple groups and multiple users for each object. The model should support for any changes for group of people. There are set of objects (DO or database, access rights) pairs for users. Digital Objects may be created by the users of the SRG system in several ways: (a) using annotation tools (Delicious, CiteULike, and Connotea); (b) using search tools (GS, WLA); (c) manually, through "Insert New Citation" interface. There are two types of options defined to make these operations: (a) *Public:* DO creation must be associated with at least one group; (b) *Private:* DO can only be assessed by owner of it unless access rights are given to other users or groups. For both public and private operation, read and write permission given to user or group for the inserted DO.

In SRG system, DOs stored for users and groups need to be protected from other users. The SRG system distinguishes between three kinds of users: *Owner* is initiator of the DO and database creation, *Group* which is the any available groups in SRG system; *Other* users. The owner of DO and database can specify the DO and database permissions for all three kinds of users mentioned above.

We have used a model similar to UNIX file system [16].The Unix RWX bits corresponds to *Read, Write, and Execute* operation for each file and directory. In SRG system, DO correspond to the file element and database corresponds to the directory element. For each DO and database, there are three types of access rights defined in the systems. Access rights, summarized in Table 1, are used in the implementation of the authorization module. Authenticated users can create databases dynamically. A user needs to have write permission for the database in order to put DOs into the database. SRG system allows user to create user's own local/remote databases. All access control lists and permissions for authorization are defined and stored in the central database.

Table1. Access Rights

Access Right	DO(Digital Object)	Database
Read	read DO	access to database for viewing DOs
Write	modify DO	insert DOs into database
Delete	delete DO	remove database from system

The access matrix [17] describes data protection in operating system. This defines permissions that each subject holds for each object [15]. As an example in Table 2, users and groups are shown in the rows and DOs in the columns. In this example, Bob is the owner of the DO_1, DO_2, and DO_3. He has read(R), write (W) and delete (D) accesses for these DOs. $Group_1$ has write access for DO_1, and read access for DO_3.Alice is other user and has only read access to DO_2. The system allows having only one owner of a DO and database. However, there might be more than one group for DO and database. Bob can modify DO_1, DO_2, and DO_3 user access rights or give them permissions to groups and other users.

Table2. Access Control List

Name	DO_1	DO_2	DO_3	User
Bob	R,W,D	R,W,D	R,W,D	Owner
$Group_1$	W		R	Group
$Group_2$	R,W	R,W	R	Group
Alice		R		Other
Henry	R,W			Other

Owner of the DOs may give access rights to other users. We have defined two methods for this operation: a) Owner can give permissions for all DOs stored in the database. User can give permissions to any user or groups which can be accessed by all users of that group for selected database. However, users and groups need to have required permissions for owner's selected database. b) Owners can modify their DO permissions for users and groups.

Having permissions for DOs and databases for users and groups doesn't provide a complete authorization scheme. We need to have level of controls of the users and groups to complete authorization module. Administration of Authorizations is used for having flexible authorization mechanism [18]. So have defined level of control of authorization in the system having Super Administrator (SA) and Group Administrator(GA).The system allows having more than one SA. An existing SA can add other SAs to the system. SA can assign any user to become GA, and remove GA from group. Each group should at least one GA. if user creates a new group, users automatically become GA.

Users are allowed to belong to more than one group. User can make a request to member of any group in the system. However, GA needs to confirm the request made by user. So SA controls groups and users. GA controls users in the group.

4. EVENT-BASED CONSISTENCY MODEL FOR THE SRG SYSTEM

Collaborative systems allow people to work together on a common task and share resources to pursue their goals. System consistency and a mechanism to avoid undesired changes in the system are critical issues in such systems. Since people are working on the same resources, they could modify the same resources. This may lead to inconsistent state of the system. Data and metadata can be entered into the SRG system from various online sources, such as social bookmarking websites, academic search tools, scientific databases, and journal and conference content management systems. Users are allowed to overwrite or modify the existing digital object in the system; this may lead to various consistency issues in SRG system. For example, one user can create an entry for a digital object by downloading data and metadata from Connotea bookmarking website (including tagging metadata). Later, another user may insert into the system the result of a WLA containing the same digital object data and metadata. The second user could choose to overwrite the existing digital object; hence she will cause the tagging of the existing digital object to be deleted. As another example, one user can create an entry for a digital object in one of several ways: (1) manual entry using the new digital object entry user interface; (2) from annotation tools by using the download tool interface; (3) search tools interface (WLA Search or GS Advanced Search). Then the user can tag this digital object with his/her tags. After that, a second user can perform a local/remote database search via PubsOnline [11] or Local/Advanced Local, and the second user can modify any metadata field of the digital object including tagging metadata. So, this will lead to losing the previous metadata of the DO. Another source of inconsistency of the data is unintentional user mistakes.

To avoid such undesired changes in the system, it is necessary to have a mechanism for restoring the system to any previous state. There are numerous systems which provide mechanisms for restoring the state of the system to any previous state. For example, in the Windows XP operating system, if the system crashes, then the tool called "System Restore" can be used for restoring the system to the last working point. As another example, many developers of the same project works on the same source code and they use one of the versioning systems such as *Concurrent Versions System* (CVS) [19] or *Subversion* (SVN) [20] to access and submit their changes. They do modifications on the code and they submit their changes into the repository. If any of the developer needs to retrieve the previous version of the code, then they can obtain it through the versioning system that they are using in their project. As a final example, Wiki systems allow their users to add, remove, change and edit a common digital content. By using "Recent Changes" page and "Revision History" function from the change log are being used for restoring the previous version of the content [21]. To solve this problem, history and rollback mechanism have been implemented in SRG system and explained in detail in section 6.

To solve the consistency problem in the SRG system, we have designed a novel event-based consistency model based on the concepts of *event* and *dataset*. An *event* is commonly defined as the act of changing the value of an attribute of some object [22]. Storing all the events about an object enables the actions on this object to be reviewed and undone [23]. An event may also be defined as an action with a time stamp and a message [24]. In our model, we adopt the view of an event as a time-stamped action on a digital object, which only maintains the modifications to an object. Every event is tied to a particular user in SRG system. We distinguish between *minor* and *major* events: insertion of a new digital object into the system or deletion of an existing digital object from the system is considered a major event; modifications to existing digital objects are considered minor events. Examples of modification are: deleting one or more fields of a digital object, changing the value of one or more fields of a digital object by adding or deleting metadata, and so on.

Another concept underlying the consistency model of the SRG system is that of *dataset*. A dataset is a collection of minor events related to a user. Datasets allow users to group the modifications to a digital object. Once a user logged into the system, all minor events are stored in the current user session (described above in Session and Event Management). In the SRG system, once a user logs in, the user's session is instantiated and later accessed through the *JavaServer Pages* (JSP) which provide a mechanism to build web content that has both static and dynamic components. During the user's session, *minor* events are saved into the user's session until they are used for creating datasets. There are two important issues requiring attention during the process of dataset creation (described in section 5 in detail): (a) Events that are selected as members of a dataset must belong to the same digital object (we do not want to include into a dataset events belonging to different digital objects). (b) The

order of the events is a key factor in that the events related to a DO are applied in the order they occur.

A dataset may be created by a user from the available minor events in the current session. Associated with each digital object, there is an initial set of digital object metadata. This initial set of metadata is represented by a major event, and it may come from different sources such as social bookmarking websites, academic search tools or manual insertion through the SRG user interface for new DO entry. Digital object metadata of a record at a certain point is the result of applying all the available ordered datasets to the initial digital object metadata (explained in detail in section 5).

5. CREATION OF DATASETS AND APPLICATION TO THE DIGITAL OBJECT (DO) METADATA

By using the initial metadata of a digital object and by applying dataset(s) on top of it, one can retrieve any version of a DO. Hence, in case of an error, we can restore the system to a previous safe state by using the related dataset for that state.

Currently, users can select any existing minor events belonging to the same digital object to create datasets. Also, in the current system a user can apply the selected minor events during the dataset creation process to simulate the current digital object metadata after creating a dataset. Minor events have no affects on the digital object until they are used for creating datasets. Once the dataset(s) are created for a digital object, then they are going to have effects on the latest digital object metadata based on their metadata, which comes from this dataset's events. Hence, each dataset and their events are evaluated to apply their metadata during the retrieval of a digital object metadata. Unless the user defines one or more datasets on the collection of events for a particular user session, all the stored events will be lost when the session ends.

The example in Figure 1 shows N datasets, named Dataset-1 ,…, Dataset-N, for a given digital object. Each dataset is composed of a number of minor events, and each dataset modifies the digital object metadata based on the events in the dataset. In our event-based model, all available datasets of a digital object will be applied on top of the initial digital object metadata based on their increasing creation time to retrieve the latest digital object metadata. During the application process, we apply each dataset and its associated events in the increasing order of their creation time.

To build a digital object metadata for a certain point, we just apply the related dataset(s) on top of the initial digital object metadata based on their creation time. As a result,

Current DO metadata = Initial DO metadata (dataset (0))

$$+\sum_{k=1}^{n} \text{dataset}(k) \text{ for this DO}$$

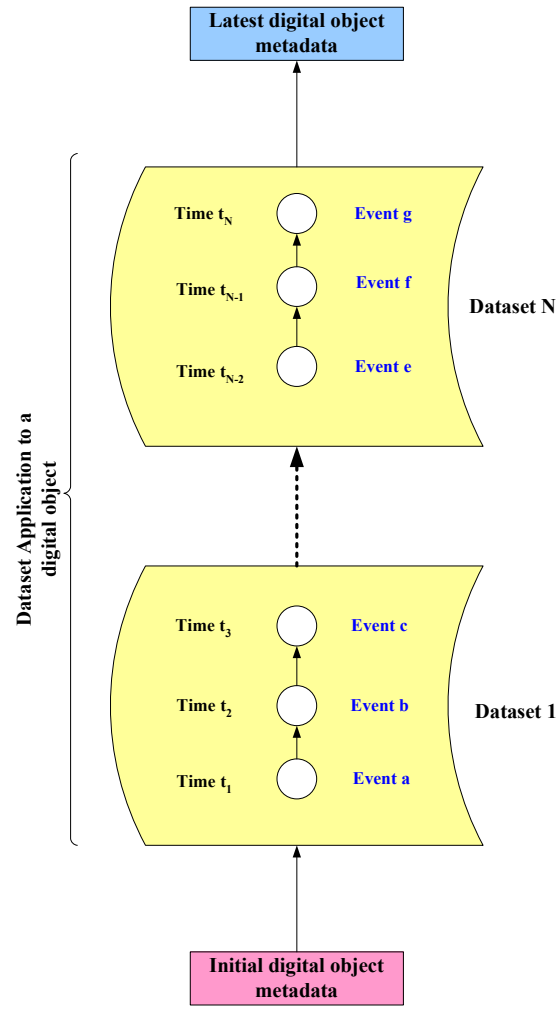

Figure 1. Retrieving the latest digital object metadata

6. HISTORY OF A DIGITAL OBJECT AND THE ROLLBACK MECHANISM

Using the mechanism described in Section 4, all the changes that have occurred to a DO are stored in the user session as minor events. They do not have any effect on the current value of the DO unless minor events are used for creating a dataset. Once a dataset is created by using minor event(s), the dataset is applied to initial DO metadata during the latest DO retrieval process.

To allow users to restore the state of the system to any previous state we have implemented a module that allows

the user to view the history of each DO and to undo any changes (rollback). In the history tool of the SRG system, each digital object has an initial entry and a list of time-stamped datasets, which represents the changes made to the digital object if there is any, in their history. Once, a user selects a time-stamped dataset, the selected state of the dataset is shown and compared with the latest metadata of the DO in a new page to the user. Furthermore, users can rollback to the selected state by using the rollback button at the bottom of the new page.

7. FUTURE WORK

Currently, the different users can override each other's DOs. A desirable feature of the system would be to merge the changes of various users. We intend to do this by distinguishing between *static* and *dynamic* metadata fields of a DO. A field is considered static if it should not be changed after the initial creation of a DO. On the other hand, a field is considered dynamic, if it can change after the initial creation of a DO. For example, authors, title are static fields, since authors of a citation will not be changed over the time. Tags, notes, or comments are examples of dynamic fields.

Merging different tags is an easy task since it is not needed to impose an order on the merged set of tags. On the other hand, merging comments from different users in a logical order is a more difficult problem. Initially we intend to just merge the comments of different users like: <user1>comment1 ;< user2>comment2… etc.

8. CONCLUSION

In this paper we discussed a novel event-based consistency model for Supporting Collaborative Cyberinfrastructure Based Scientific Research and the implementation of the proposed framework in SRG system. Furthermore, we described the current state of the development of the event-based consistency system and outlined several directions of future work.

REFERENCES

[1] F. Ian, K. Carl, and T. Steven, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations." vol. 15: Sage Publications, Inc., 2001, pp. 200-222.

[2] C. K. Ian Foster, Jeffrey M. Nick, Steven Tuecke, "The Physiology of the Grid," Argonne National Laboratory.

[3] G. C. Fox, A. F. Mustacoglu, A. E. Topcu, and A. Cami, "SRG: A research grid for Cyberinfrastructure based scientific research," Community Grids Lab, Indiana University 2006.

[4] D. C. M. I. (DCMI), "Dublin Core Metadata Initiative (DCMI)."

[5] S. Rahul, L. Zhao, K. Pilho, P. Derik, and J. Ramesh, "Event-based modeling and processing of digital media," in *Proceedings of the 1st international workshop on Computer vision meets databases* Paris, France: ACM Press, 2004.

[6] P. Liliane, B. Nathalie, Herv, and Chaudet, "Event oriented representation for collaborative activities in an intensive care unit," in *Proceedings of the 2005 annual conference on European association of cognitive ergonomics* Chania, Greece: University of Athens, 2005.

[7] K. Pilho and J. Ramesh, "Heterogeneous media events processing systems," in *Proceedings of the 2004 ACM SIGMM workshop on Effective telepresence* New York, NY, USA: ACM Press, 2004.

[8] B. Olivier, "Event points: annotating XML documents for remote sharing," in *Proceedings of the 2005 ACM symposium on Document engineering* Bristol, United Kingdom: ACM Press, 2005.

[9] A. B. R. W. T. v. E. E. d. Maat, "A Content Management System based on an Event-based Model of Version Management Information in Legislation," pp. 19-28, 2004.

[10] IBM WebSphere Session Management web site. http://www.informit.com/articles/article.asp?p=332851&rl=1

[11] A. M. Scott, K. Richard, L. Matt, and S. Craig, "PubsOnline: open source bibliography database," in *Proceedings of the 33rd annual ACM SIGUCCS conference on User services* Monterey, CA, USA: ACM Press, 2005.

[12] Delicious web site. http://del.icio.us

[13] CiteULike web site. http://www.citeulike.org

[14] Connotea web site. http://www.connotea.org

[15] T. William, A. Gail-Joon, P. Tanusree, and H. Seng-Phil, "Access control in collaborative systems." vol. 37: ACM Press, 2005, pp. 29-41.

[16] S. T. Andrew and R. Robbert Van, "Distributed operating systems." vol. 17: ACM Press, 1985, pp. 419-470.

[17] W. L. Butler, "Protection." vol. 8: ACM Press, 1974, pp. 18-24.

[18] B. Elisa, J. Sushil, and S. Pierangela, "A flexible authorization mechanism for relational data management systems." vol. 17: ACM Press, 1999, pp. 101-140.

[19] Open Source Version Control web site. http://www.nongnu.org/cvs/

[20] Subversion web site. http://subversion.tigris.org/

[21] Wiki - Wikipedia web site. http://en.wikipedia.org/wiki/Wiki

[22] S. R. David and K. Balachander, "An event-based model of software configuration management," in *Proceedings of the 3rd international workshop on Software configuration management* Trondheim, Norway: ACM Press, 1991.

[23] G. Fox, "Collaboration within an Event based Computing Paradigm," 2001. http://aspen.ucs.indiana.edu/collabtools/extras/indianafeb01.html

[24] S. P. a. G. Fox, "A Scalable Durable Grid Event Service," 2001. http://grids.ucs.indiana.edu/ptliupages/publications/GESOverview.pdf