Published in final edited form as:

Proc SPIE. 2009 February 11; 7264(726402): . doi:10.1117/12.811716.

Securely and Flexibly Sharing a Biomedical Data Management System

Fusheng Wang^a, Phillip Hussels^a, and Peiya Liu^a

^a Integrated Data Systems Department, Siemens Corporate Research, Princeton, NJ, USA

Abstract

Biomedical database systems need not only to address the issues of managing complex data, but also to provide data security and access control to the system. These include not only system level security, but also instance level access control such as access of documents, schemas, or aggregation of information. The latter is becoming more important as multiple users can share a single scientific data management system to conduct their research, while data have to be protected before they are published or IP-protected. This problem is challenging as users' needs for data security vary dramatically from one application to another, in terms of who to share with, what resources to be shared, and at what access level. We develop a comprehensive data access framework for a biomedical data management system *SciPort*. *SciPort* provides fine-grained multi-level space based access control of resources at not only object level (documents and schemas), but also space level (resources set aggregated in a hierarchy way). Furthermore, to simplify the management of users and privileges, customizable role-based user model is developed. The access control is implemented efficiently by integrating access privileges into the backend XML database, thus efficient queries are supported. The secure access approach we take makes it possible for multiple users to share the same biomedical data management system with flexible access management and high data security.

1. INTRODUCTION

Biomedical database system needs not only to address the issues of representing, authoring, and manipulating complex scientific data, but also to provide flexible and secure access control to maximize the usage of the system. However, data access control has to be carefully managed to prohibit unwanted data disclosure. This is especially critical as many research data are sensitive before they are published or IP-protected. Indeed, users' need for data security varies dramatically from one application to another. For example, for one university department, around one thousand students and researchers need to have their own individual spaces for managing their experimental data on a single server, and data have to be invisible from each other. For another clinical trial research site, the data are visible for everybody, but different users hold different privileges such as reading, authoring, organizing and sharing. The high diversity of use cases makes it challenging to manage users, their privileges, and access controls on resources.

In a collaborative environment, there are often many users sharing the same Scientific Database Management System, and the access control of data have to be enforced for data security. The requirements for access control can vary, as demonstrated in the following example use cases.

Use Case 1: Single Space, Differentiated Privileges. For example, for one clinical trial research group at UCI, the data on the Server are available for everybody. However, users can be associated with different privileges, for example, i) reader with read-only permission of all documents; ii) author with permission to author documents; iii)

publisher who can publish documents to a Central Server for sharing; iv) organizer who has read/write permissions on folders, schemas and reports; and v) administrator who has all privileges and can create users and assign privileges. Thus, NIH project and grant reviewers can login as a reader to evaluate the data; nurses can be authorized as author to collect patient data, etc.

Use Case 2: Personalized Space. For the Pathology and Laboratory Department of University of Pennsylvania, around one thousand students and researchers need to have their own individual spaces for managing their experimental data on a single Server, and their data are totally isolated from others.

Use Case 3: Multiple Spaces. For one Siemens Medical Research Collaboration, there are hundreds of research organizations to deliver data to Siemens, and these research organizations are divided into ten zones across the US. Each zone will provide a single server for the research organizations in the zone. While each research organization will share data within itself, the data are often sensitive and not yet published, and they need to keep the data confidential from other research organizations who are sharing the same server.

The high diversity of use cases makes it challenging to manage users, their privileges, and access controls on documents. We develop a space and role combined security model to provide comprehensive and convenient access control and security management.

To provide efficient query support under the access control framework, the access control is implemented efficiently by integrating access privileges into the backend XML database.

2. OVERVIEW OF SCIENTIFIC DATA MANAGEMENT WITH SCIORT

SciPort was first developed as a scientific data management system for scientific data modeling, data authoring, management, viewing, searching and exchange. SciPort takes an XML based approach for data modeling, schema representation, and storage and queries,^{1, 2, 3} and is supported by a backend native XML database. In SciPort, scientific data are represented as XML modeled documents, and documents are also organized into a hierarchy way for convenient browsing.

2.1 SciPort Document Model

SciPort was first developed as a scientific data management system for scientific data modeling, data authoring, management, viewing, searching and exchange. SciPort takes an XML based approach for data modeling, schema representation, and storage and queries.

Scientific experiments often consist of complex steps or processes. A complex scientific experiment can be modeled as a set of atomic objects. The context information of such objects can then be represented through a unified document model, *SciPort Document*. SciPort Document can represent both (nested) structured data, files and images.

A SciPort document includes several objects: i) Primitive Data Types/Fields. Primitive data types are used to represent structural data, including *integer*, *float*, *date*, *text*, and Web-based data types such as *textarea*, *radiobutton*, *checkbox*, *URL*, etc.; ii) File. Files can be linked to a document through the file object; iii) *Reference*. A reference type links to another SciPort document; iv) Group. A group is similar to a table, which aggregates a collection of fields or nested groups. There can be multiple instances for a group, like rows of a table; and v) Category. A category relates a list of fields, e.g., “patient data” category, “experiment data” category, etc. Categories are used only at the top level of the content, and categories are not nested (Figure 1).

2.2 XML Based Implementation

The model can be best implemented as XML. The hierarchical nature of the data model fits very well with the tree based XML data model. Users can also easily define their own schemas which are internally represented with an XML-based schema definition language. Here we take the native XML database approach to manage scientific documents, where standard XML query language XQuery⁴ is supported. We provide two options: Oracle Berkeley DB XML,⁵ an open source embedded XML database, and IBM DB2 with pureXML^{TM} support. SciPort provides comprehensive Web-based tools to support authoring and searching.⁶

2.3 Hierarchical Organization of Scientific Data

To provide an intuitive way to present data, we develop a hierarchical model to organize scientific data, thus documents can be quickly browsed and identified through the hierarchy.

A hierarchy provides a folder-like data organization. There can be multiple levels defined, and each folder can also be associated with a *slot document* to describe the folder. Each folder can contain subfolders or schemas with which documents are created and grouped. A *report* can be defined for each folder to provide data aggregation of all data under the folder. A sample hierarchy for a clinical trial site is shown in Figure 2.

A hierarchy contains the following objects: i) *Level*, definition of a folder type, e.g., “patient”, “measurement”. A level can be associated with schemas to collect documents, and a slot schema to attach descriptive document for the folder; ii) *Folder*, an instance of a level type; iii) *Slot Document*, a document used to describe the folder; iv) *Report*, a tabular summary of information under a folder; for example, a report can be defined to summarize the average tumor size of patients; v) *Schema*, associated with a level for collecting and grouping documents under a folder; and vi) SciPort Documents.

3. THE SPACE AND ROLE BASED SECURITY MODEL

There are four fundamental resources in SciPort: document, schema, folder and level, especially, folders can be nested. While user privileges on these resources can be specified at individual level, this is extremely cumbersome and not feasible to use. Instead, we propose a Space Based Security Model to simplify the solution.

3.1 The Space Based Security Model

As shown in Figure 3, based on the hierarchy of a Server, resources (documents and folders) can be divided into separate spaces. Each space has its own “territory”, and there is no overlap between spaces.

Definition of a space—A space is an aggregate of resources defined on top of a hierarchy, consisting of documents and folders under one or multiple folder paths. No overlap of spaces is allowed, but one user can reside in multiple spaces. All folders and documents created under a space belong to the same space. Folder names and slot documents of the folders (if any) on the to-root-path of a space is automatically shared as read accessible. For example, users in Space B illustrated in Figure 3 can always access information on its parent folder and the root folder.

A space can be represented with its *root folders*, the top folders for the space. By representing a folder with an XPath, a space is then a combination of multiple XPaths. e.g., Space C can be represented as: `/folder[id="A1"]/folder [id="B2"]/folder [id="C2"]` plus `/folder[id="A1"]/folder[id="B3"]`.

Users in the same space are always able to read all the information in the space, but privileges can vary.

3.2 Privileges and Roles

The different user privileges on the resources are summarized in Figure 4. Privileges can be summarized with three major categories: i) *Authoring*: create, delete, update, and upload on resources; ii) *Reading*: read and download a resource; and iii) *Publishing*: publish a resource to a Central Server.¹ We propose to use roles to simplify the management of privileges.

As shown in Figure 5, the following roles are predefined in a space.

- *Reader*. A user of reader role can read everything in the space. Reader role is the most fundamental role for every user in the space.
- *Author*. A user of author role can further create and upload documents in the space. Besides, the author will also be the owner of the documents he creates and holds the delete and update privileges on them.
- *Organizer*. Besides author role privileges, an organizer can further manipulate schemas and folders (create, delete, update and upload).
- *Publisher*. A publisher is granted the privileges to publish and share resources to a Central Server.¹
- *Space Administrator*. A space administrator will own all the privileges of reader, author, organizer and publisher, and is able to manage users and the roles.

The above are roles defined on a space, and apply to all objects in the space, i.e., an organizer can edit any folder –except removing the space root folders.

There is another special role *owner*, which is defined on a specific document. A document owner can also assign more owners to the document with update and delete privileges.

Another special role is the *system administrator* who owns all the privileges of reader, author, organizer and publisher, and can administrate spaces and users (including space administrator) across all spaces. Note that only system administrators can remove a space, and neither space organizer nor space administrator can delete a folder which is among the root folders of the space.

The combination of the space and roles based approaches in SciPort not only provides a comprehensive security model to support database sharing for multiple users, but also minimizes the complexity of access control management.

3.3 Example Use Case Implementations

For Use Case 1, the whole server will share a single space, and users with different roles are created: one administrator account for the IT personnel, one author account for a nurse, and organizer created for a Ph.D. student to setup folders and collect experimental data, and one public reader account created for NIH grant reviewers. The principle investigator has a publisher account to selectively publish and share the data.

For Use Case 2, the IT coordinator is the system administrator, and a personalized space is created for each student, who is assigned with an organizer role.

For Use Case 3, for each server, each research site is defined with a space based on the location on the hierarchy, e.g., a folder “UCLA” in the branch “Siemens Research Collaboration USA → MR → UCLA” of the hierarchy “Research Collaboration →

Modality → Research Site". A space administrator is created for each space, who is authorized to manage users and their privileges.

4. CONFIGURABLE ACCESS CONTROL THROUGH ROLE CUSTOMIZATION

Customizable Roles

To provide flexible access controls, roles can also be customized by combining multiple privileges.

5. SECURELY RETRIEVING DATA

Next we discuss how we can effectively support the security management on retrieving data.

Secure document retrieval from the hierarchy

A user can be associated with one or multiple space(s). For data browsing through the hierarchy, only the user's space folders (together with the to-root-path folders) are retrieved. This can be done through applying the XPath of the space folders to the hierarchy.

Secure document retrieval from document collections in the XML database

Since each document can be associated with only a single space, it can be filtered through an XML attribute in the document root. For any query submitted by a user, the query will be automatically constrained with an additional condition on matching a document's space with the user's space(s). For example, if the user's space is "B" and "C", the condition will be: /document[@space= "B" or @space = "C"].

6. IMPLEMENTATION

SciPort uses a hybrid approach to manage data. User management is implemented with relational database backend through Hibernate, and resources (hierarchy, documents and schemas) are implemented as XML data type supported in native XML database (e.g., the pureXML support in DB2).

A system administrator can setup the system by creating spaces first, as shown the example (Figure 6). Once a space is ready, the system administrator can create space administrators (Figure 7). Space administrators will able to manage their own users in the space.

The system administrator will also setup hierarchy folders and assign folders to spaces, as shown in Figure 8.

7. CONCLUSION

For large scale biomedical data management system, system sharing is essential to maximize the usage and lower the cost. This leads to the requirements of flexible access control management and highly secured data in the system. In this paper, we present our work on implementing a comprehensive access control model and user management on top of an existing scientific data management system – *SciPort*. *SciPort* provides flexible and secure multi-level access control on resources, customizable role-based user management, and efficient queries under security constraints. The system has been successfully used for scientific data management and sharing in NIH research consortia, clinical trials, and large scale research collaboration inside Siemens.

Acknowledgments

The project is funded in part by the National Institutes of Health, under Grant No. U54 CA105480-05.

References

1. Wang F, et al. An adaptable xml based approach for scientific data management and integration. In SPIE Medical Imaging. 2008
2. Wang F, Rabsch C, Kling P, Liu P, Pearson J. Web-based collaborative information integration for scientific research. ICDE. 2007:1232–1241 .
3. Wang F, Liu P, Pearson J, Azar F, Madlmayr G. Experiment management with metadata-based integration for collaborative scientific research. ICDE. 2006
4. XQuery 1.0: An XML Query Language. <http://www.w3.org/TR/xquery/>
5. Oracle Berkeley DB XML. <http://www.oracle.com/database/berkeley-db/xml/>
6. SciPort Wiki. <https://sciportserver.scr.siemens.com/mediawiki/>

General information
Patient
Annotation

Tumor Collection:

Name: 0019-2
UID: 1.2.288.3.1220941281.3804.1194800177.85
AuthorName: cds2
DateTime: 2007-11-15
ImagingObservationCharacteristicCollection:

CodingSchemeDesignatorRADREX
CodeMeaning: LIDC Sphericity 2
CodeValue: REX4032

DataCollection:

Diameter: 77.4222259521484

SpatialCoordinateCollection:

X	Y	Z	ImageReferenceUID
373	149	0	1.3.6.1.4.1.9328.50.1.8956
397	287	0	1.3.6.1.4.1.9328.50.1.8956

AIMFile:

0019-2.xml

Name: 0019-3
UID: 1.2.288.3.1220941281.3804.1194801748.163
AuthorName: scd3
DateTime: 2007-11-15
ImagingObservationCharacteristicCollection:

CodingSchemeDesignator: RADREX
CodeMeaning: LIDC Fluid
CodeValue: REX4011

CodingSchemeDesignator: RADREX
CodeMeaning: LIDC Texture 2
CodeValue: REX4061

DataCollection:

Diameter: 67.5987167358399

SpatialCoordinateCollection:

X	Y	Z	ImageReferenceUID
312	247	0	1.3.6.1.4.1.9328.50.1.8956
418	186	0	1.3.6.1.4.1.9328.50.1.8956

AIMFile:

0019-3.xml

File

DICOMImage:

CT.1.3.6.1.4.1.9328.50.1.8956.dcm

Figure 1.
A Sample SciPort Document

Figure 2.
A Sample SciPort Hierarchy

Figure 3.
Space Based Security Model

	Permission	Document	Schema	Folder	Level
Author	Create	Create a document	Create a schema	Create a folder or level document	Create a level
	Delete	Delete an existing document	Delete a schema	Delete a folder or level document	Delete a level
	Update	Update a document	Update a schema	Update a folder or level document	Update a level
	Upload	Upload a document or file	Upload a schema	Upload a folder and level document	Upload a level
Read	Read, download	Read and download a document	Read and download a schema	Browse a folder	Read a level
Publish	Publish to Central Server	Publish a document to the Central Server	Publish a schema to the Central Server	Publish a folder to the Central Server	Publish levels to the Central Server

Figure 4.
Summary of Privileges

	Role	Document	Schema	Folder	Level
Space	Reader	Read and download	Read	Read	Read
	Author	Read, create, upload, delete, update on owned documents	Read	Read	Read
	Organizer	<i>Same as author</i>	Read, create, delete, update, upload	Read, create, delete, update, upload	Read
	Publisher	Read and publish	Read and publish	Read and publish	Read and publish
	Administrator	All (reader, author, organizer and publisher) + administrate users in its space			
System	Administrator	All (reader, author, organizer and publisher) + administrate spaces and users			

Figure 5.
Summary of Roles in a Space

Name	Description	
Stanford University		
University of California, Irvine		
University of California, Los Angeles		
University of California, San Francisco		
University of Southern California		

Figure 6.
Sample Spaces on one Local Server

The screenshot shows a web form for creating a user. It includes the following elements:

- Name:** A text input field containing "UCLA admin".
- Email:** A text input field containing "ucla.admin@ucla.edu".
- Administrator:** A checkbox that is currently unchecked.
- Roles:** A dropdown menu with "University of California, Los Angeles" selected. A secondary dropdown menu is open, showing a list of roles: "Administrator", "Reader", "Organizer", and "Author".
- Buttons:** "Save" and "Cancel" buttons are located below the roles dropdown.

Figure 7.
Example of creating users

by **Level 1: Folder 1**

Folder 1

Common fields

Name*: University of California, Lc

Description: Siemens research partners at UCLA

Space: University of California, Los Angeles

Allowed Schemas

- Group
- Stanford University
- University of California, Irvine
- University of California, Los Angeles
- University of California, San Francisco
- University of Southern California

* required fields

save folder cancel

Figure 8.
Example of assigning a folder to a space