Towards Access Control for Visual Web Model Management

Guanglei Song[1] Kang Zhang[1] Bhavani Thuraisingham[1] Jiannong Cao[2]

[1]*Department of Computer Science,*
University of Texas at Dallas
Richardson, Texas 75083-0688 USA
{gxs017800, kzhang, bhavani.thuraisingham }@utdallas.edu

[2]*Department of Computing*
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
csjcao@comp.polyyu.edu.hk

Abstract

With the advance of E-Commerce over Web-based information, the interoperability of isolated XML repositories and databases over the Internet has drawn an increasing interest recently. Little effort, however, has been made to preserve necessary autonomy and security of each individual XML repository or database during information exchange or evolution. Generic model management has been intensively researched and also implemented in a prototype since its first introduction. Security related research is yet to be conducted for model management. This paper presents a uniform security model for access control specifications of heterogeneous data models over the Web. Based on the uniform representation, we present security extensions to our previous work on visual model management operators for managing access control specifications to allow heterogeneous Web data models to exchange information over public networks.

1. Introduction

Web-based information interchange is particularly important in electronic commerce (EC) applications, where basic transactions such as vendor registrations, bidding submissions, requests for quotes, and contracts are increasingly realized by exchanging appropriate digital documents [Dam02]. The huge success of the Web as a platform for EC and information dissemination has brought an increasing awareness of the fact that document exchange over the Internet should meet security requirements such as fine-grained authenticity, and access control, involving data units at the level of granularity stipulated by the communicating parties. According to Samarati *et al* [Sam96], authorizations are specified on portions of a HTML document, yet no semantic context similar to that provided by XML [Bra00] can be supported. While HTML has its inherent limitations, XML has great potential to provide fine-grained security features. XML access control has been a hot research topic since the first set of access control specifications was proposed [Dam00]. Recently, the continuing demand for information sharing has shifted interest from stand-alone XML repositories to inter-connected and large-scale cooperative XML systems [Dam01]. As more and more Web information sharing occurs on the Internet, there are two categories of heterogeneous information interpretability among EC Web sites.

- **Single-site interpretability:** On a single Web site, each layer has its own design artifacts, such as UI design, database schema, and XML schema. While a user interface is typically designed and represented in a markup language, practical E-Commerce systems have enormous information about customers, vendors, commodities, and others that most likely reside in relational databases. These data in different formats inevitably needs to exchange information.
- **Multi-site interpretability:** With the advance of B2B models, more and more Web sites need to exchange information over the Internet. Each Web site may have its own schema. When two Web sites exchange information, they need to deal with heterogeneous formats via a common schema or other translation resort. For example, a Web service could be formed by travel agencies, hotels, restaurants, and car rental companies to provide a user with all-in-one traveling planner.

Even though every individual data model may have highly secure access control specifications and enforcement mechanism, the federation of data models is not necessarily secure. Security of a union of systems depends on the weakest link. When information of different models is interchanging, it opens a window for attack. It is necessary to securely manipulate multiple models. Most previous work however concentrated on the access control of individual data models or management of model without security concern, such as Model management, which is a new approach to metadata management that offers a high-level programming interface [Ber00] and avoids object-at-a-time primitives. It reduces the amount of programming needed for metadata intensive applications by manipulating models with generic operators. Our previous work provides a visual model management architecture to match the interactive nature of and ease the use of model management system [Son04].

This paper focuses on the security properties of model management, and explores various issues and solutions to achieve secure model management for Web data models. The remainder of the paper is organized as follows. Section 2 introduces the model management concept. Section 3 overviews two security models over the Web. Section 4 presents an abstract security model and proposes security extensions to model management operators. Section 5 discusses the future research directions based on the work in this paper. Section 6 compares related work, and Section 7 concludes the paper.

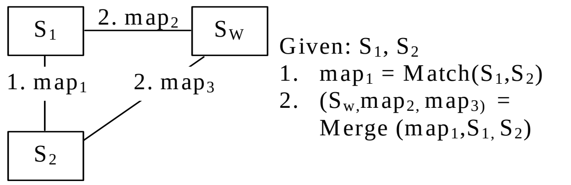
2. Model Management

Model management treats models and mappings at a high level of abstraction and regards them as bulk objects. Model management environment offers operators that generalize the transformation operations for various metadata applications. The main model management operators are described as the following [Ber03]:

- Match – takes two models as input and returns a mapping between them.
- Compose – takes a mapping between models A and B and a mapping between models B and C, and returns a mapping between A and C.
- Diff – takes a model A and mapping between A and some model B, and returns the sub-model of A that does not participate in the mapping.
- ModelGen – takes a model A, and returns a new model B that expresses A in a different representation (i.e. data model).
- Merge – takes two models A and B and a mapping between them, and returns the union C of A and B along with mappings between C and A, and C and B.

These operators are applied to models and mappings as a whole, rather than to their individual elements. The operators are generic in the sense that they can be utilized for different kinds of models and scenarios.

Consider a typical example of building a database federation. Suppose we are given a mapping map_1 from a database S_1 to another database S_2, and wish to build a federation of the two databases, where S_2 is similar to S1 (Figure 1). First we call Match (S_1, S_2) to obtain a mapping map_2 between S_1 and S_2, which shows where S_2 is the same as S_1. Second, we call Merge (map_1, S_1, S_2) to obtain a mapping map_3 between S_2 and S_W and a mapping between S_1 and S_W. Comparing to programming the whole system for individual requirements, using model management reduces considerable programming effort by composing generic operators.



Given: S_1, S_2

1. map_1 = Match(S_1,S_2)
2. (S_w,map_2, map_3) = Merge (map_1,S_1, S_2)

Figure 1. Using model management to help generate a database federation

3. Current Access Control Models

Recently many security models for various data models have been proposed. We survey existing access control models and classify security models related to Web data security into two categories as XML related models and database related models.

3.1 XML access control Models

Several authorization-based XML access control models have been proposed. Each authorization rule consists of *subject*, *object*, *action*, *access* and some other extensions. Damiani's approach associates a specific authorization sheet with each XML document/DTD expressing the authorizations on the document [Dam00], The approach is further extended by enriching the authorization types supported by the model [Dam02], providing a complete description of the specification and enforcement mechanism. An access control environment for XML documents and techniques to deal with authorization priorities and conflict resolution issues are proposed by Bertino and Ferrari [Ber02]. Although our uniform security model is based on existing XML authorization models such as [Dam00a], we focus on how to use the uniform representation to provide security extension for Web data model management, and none of the above XML authorization models addresses the interaction between different access control models.

3.2 Database access control models

Database access control models can be further classified into two categories: multilevel security models [Jaj90, San98] and discretionary security models. A multilevel security model assigns each data object (e.g., a tuple) as well as each subject (e.g., a user) a security level (or class). It enforces the following two rules: (1) a level L_i subject can never read a level L_j data object unless L_i >= L_j; (2) a level L_i subject can never write a level L_j data object unless L_j >= L_i. Multilevel security models are seldom used in commercial applications due to their restrictive nature. A discretionary security model allows the creator of a data object x to own all the privileges associated with x and to grant some of the privileges to other users so that various access control policies could be enforced. Discretionary security models are dominant in commercial data management. Although several more expressive and flexible discretionary security models are proposed [Jaj97, Jaj01], most database systems implement discretionary access control models similar to the one implemented in System R [Gri76]. System R only directly supports table or column level authorizations. Role-based access control [San96] is not implemented in System R but implemented by most existing DBMSs such as Oracle. Similarly approaches in the context of object-oriented databases have also been presented [Fer94, Rab91].

4. Security Extension to Model Management

To manage access control specifications for heterogeneous data models, an abstract access control model is desirable. This section presents a visual representation of an abstract access control model and extends model management operators with security property.

4.1 Uniform Abstract Access Control Model

The development of an access control system requires the definition of subjects and objects against which authorization must be specified and access control must be enforced [Dam02]. We define a uniform abstract access control model, which consists of a set of rules, each being a tuple of five elements: *subject*, *object*, *action*, *authorization*, and *propagation*.

Access control regulates access to the data, such as HTML documents, XML documents, and databases called *objects*. Those who try to access these objects are called *subjects*.

Usually subjects can be referred to on the basis of their identities or its locations from which requests originate. Some models combine the two together to give more details for subjects, while we provide an abstract representation for each subject by a unique user-defined identifier, such as teacher, administrator. A set Obj of uniform identifies (URI) [Ber98] denotes the resources to be protected, i.e. objects. For XML documents, URI is extended with XPath [W3c01], for identifying the elements and attributes within a document. In database area, the granularity of identifiers is down to table or at most column. To uniformly identify various models over the Internet, we propose a novel identifier UPath. In a similar way to XPath, UPath identifies objects of data models by path, e.g. tables and columns of a relational schema, elements and attributes of an XML schema. We borrow some ideas from relational calculus to represent objects in a relational database by UPath. For example, E-business company none.com has a database D, which includes table sample. Then the UPath of the column stock_no of sample will be /D/sample/stock_no. In the access control model, we propose the URI + UPath as the object path expression that will be illustrated by an example in the next section. Subjects can take a set of actions, including *read*, *write*, *update*, and *delete*. Authorization specifies the negative or positive response to requests, i.e. *grant* or *deny*. Our model specifies the propagation as *local* or *recursive*, referring to the influence to the object locally or recursively to its corresponding child objects.

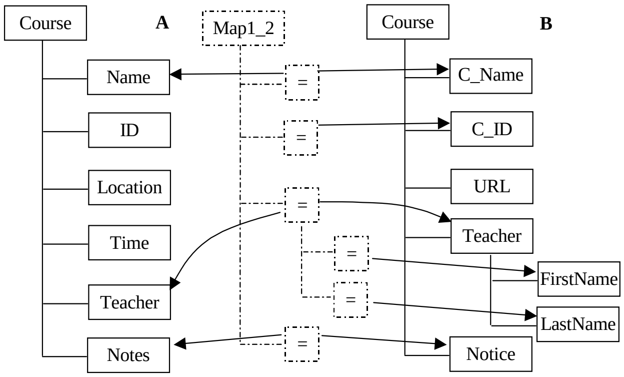


Figure 2. Course schemas for two schools

Figure 3 shows a visual representation of access control rules, represented by an access graph. Each rule is a link between a subject and an object. A subject is represented by a labeled rectangular connecting to objects that are represented by labeled eclipse. A gray eclipse represents recursive access, and a white eclipse indicates local access. Label of each link, R or W in the example, represents the activity. Circle and cross on the link represent grant and deny of access respectively.

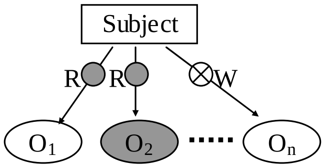


Figure 3.Visual representation of access control rules

4.2 An Illustrative Example

Two schools offer two types of courses, i.e. traditional classroom courses and distance learning courses, for students to register online. Assume the two schools wish to provide a uniform online course registration system, and have to deal with the data defined in two schemas. Figure 2 shows the two schemas, A for classroom courses and B for distance learning courses. Both schemas are XML Schemas.

Each school has a set of local access control rules as shown in Tables 1 and 2. In Table 1, Rules 1 and 2 restrict public access only to Name and Teacher of a Course. A student can read everything about a Course, while a teacher can change Notes of a Course.

Management of individual authorization rules has been intensively investigated. If a user tries to access the course information, the user will be first authenticated and then the access control enforcement mechanism will query and prune the information according to the access control rules (ACRs). Figure 4 illustrates the access control process. Enforcement mechanisms of XML access control are surveyed by Luo *et al* [Luo04].

Table 1. Access Rules for Model A

	Subject	Object	Action	Authorization	Propagation
1	Public	/Course/Name	Read	Grant	L
2	Public	/Course/Teacher	Read	Grant	L
3	Student	/Course/	Read	Grant	R
4	Teacher	/Course/	Read	Grant	R
5	Teacher	/Course/Notes	Write	Grant	L

Table 2. Access Rules for Model B

	Subject	Object	Action	Authorization	Propagation
1	Everyone	/Course/C_Name	Read	Grant	L
2	Everyone	/Course/Teacher	Read	Grant	R
3	Student	/Course/	Read	Grant	R
4	Lecturer	/Course/	Read	Grant	R
5	Lecturer	/Course/URL	Write	Grant	L
6	Lecturer	/Course/Notes	Write	Grant	L

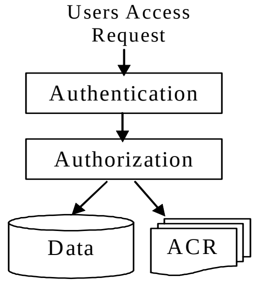


Figure 4. Access Control Enforce Framework

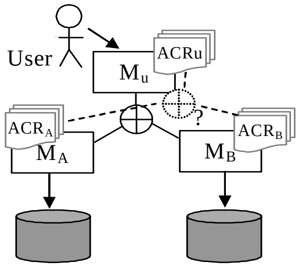


Figure 5. Unified course registration online system

If the two schools want to exchange credits and create unified courses, a straightforward approach is to create a unified model and migrate existing data to conforming to the new model.

Model management system eases the process by generic operators like Match and Merge. Figure 5 shows the scenario of unifying the two models by the two operators. ACR$_A$ and ACR$_B$ are access control rules for M$_A$ and M$_B$ respectively. M$_u$ is the unified model of models M$_A$ and M$_B$. ACR$_u$ is a set of access control rules for model M$_u$. The model management system matches and merges M$_A$ and M$_B$ to generate M$_u$, but is not able to automatically generate ACR$_u$. Users have to construct ACR$_u$ manually from the scratch. It is error-prone, time-consuming, and highly risky to manually manipulate access control rules in a large scale such as an EC Web site. To ease the process, a security extension for model management operators, like Match, to automatically manage access control rules is desirable.

4.3 Schema Matching with Security Property

Regarding the security of data models, the Match operator only matches objects, called *object matching*. For the example in Figure 2, the Match operator takes models A and B as input, and produces mapping Map$_{1\_2}$ but not mapping for access control rules. In addition to the object matching, the match operator is extended to match the subjects of two access control rules, called *subject matching*. If two subjects have similar or the same accesses, they are mapped as the same subject. For example, "Public" in Table 1 is the same as "Everyone" in Table 2, so they are regarded as one subject in the unified model.

Match with a security extension takes two input models, each having a set of access control rules attached. The extended Match operator is defined as follows:

Definition 1: *(Map$_m$, Map$_a$) = Match ((M$_1$, ACR$_1$), (M$_2$, ACR$_2$)), where M$_1$ and M$_2$ are two data models, ACR$_1$ and ACR$_2$ are access control rules of models M$_1$ and M$_2$ respectively.* □

The result (Map$_m$, Map$_a$) contains two mappings, Map$_m$ between objects of input models and Map$_a$ between the subjects of two sets of access control rules. Figure 6 shows the result of subject mapping for access control rules, where Public of ACR$_1$ is equal to Everyone of ACR$_2$, and Teacher of ACR$_1$ plays the same role of Lecturer of ACR$_2$.

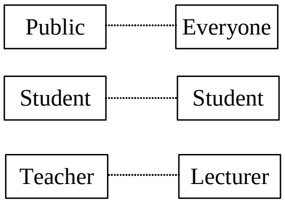


Figure 6. The Map$_a$ example

The Match operator can be realized by a schema-matching algorithm, which has been investigated for several years. Many techniques [Rah01], such as graph isomorphism, natural language processing, machine learning, and data mining, have been proposed. The algorithm for subject matching can be implemented in a similar way. For example, linguistically Public is similar to Everyone, and some matching algorithms would create a mapping between the two subjects as shown in Figure 6.

Existing matching algorithms however do not consider security properties of access control rules, and produce poor and sometimes risky mappings. For example, assume that the example in Section 4.2 has one subject for each model, Admin for model A, and Administrator for model B. While Admin is a Web site administrator and has full access to everything, Administrator cannot write notes of a course. A matching algorithm would produce a mapping between Admin and Administrator, thus introduce a possible violation of access rules of ACR_2, e.g. an Admin user can write notes of a course in model B. A security extension of match should produce a safe subject mapping defined as follows.

Models M_1 and M_2 have access control rules ACR_1 and ACR_2 respectively. S_1 and S_2 are subjects of ACR1 and ACR_2. Map_{1_2} is the object mapping between M_1 and M_2. Map_s is a subject mapping between S_1 and S_2.

Definition 2: Map_s is *safe* if and only if

$\forall$ $(s_1, s_2) \in Map_s$ $\forall$ $(o_1, o_2) \in Map_{1\_2}$ $\forall$ a (grant $(s_1, o_1, a) \leftrightarrow$ grant (s_2, o_2, a)), where s_1 and s_2 are subjects of S_1 and S_2, o_1 and o_2 are objects of M_1 and M_2, a is an action, grant (s_1, o_1, a) means that the s_1 is granted to perform action a on o_1. □

To produce safe subject mappings, we propose the following three approaches.

1. Security filter: The most straightforward approach is to transplant the object matching algorithm to match subjects with a security filter attached to the back end to remove the violation of access control rules. Once the filter finds a violation, it removes the subject mapping. The approach is safe, but may impair effectiveness of the matching algorithm. For example, if a Supervisor in model B has full access to model B like Admin of model A, and security filter cannot match Supervisor with Admin, since Admin is chosen to match Administrator in the first place.

2. Security dimension: Matching algorithms, such as SFA [Mel02], and Cupid [Mad01], initially calculate two objects' similarity measured by structural pattern, background knowledge, but no security was considered. The approach is to provide security as another dimension of similarity. Our experiment shows that the approach does not accurately remove violating mappings. The reason is that security is only one dimension of similarity, and the algorithm may still produce violating mappings.

3. Security isomorphism: The approach calculates the similarity of subjects from not only linguistics but also semantics of access control rules, and generates subject mapping based on the isomorphism of ACRs.

Among above three algorithms, the security isomorphism algorithm generates more accurate mappings than security filter does, and also can be proved to generate only safe. Therefore we choose the approach as security extension of our framework and discuss it in more detail as following.

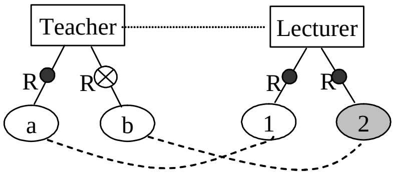


Figure 7. Graphical representation of access control rules

The algorithm matches subject's access rules to calculate the similarity of two subjects. Similarity of two subjects consists of LS (linguistic similarity) and SS (semantics similarity). If s is a subject of access rules, we represent G (s) as a set of objects that S has access and D (s) as the set of objects that S was prohibited to access.

Definition 3: Overlap set between two subjects: *$O(s_1, s_2) = \{(o_1, o_2)| o_1 \in G(s_1)$ and $o_2 \in G(s_2)$, and s_1 and s_2 are two subjects, and (o_1, o_2) is a mapping$\}$.*□

As shown in Figure 7 the overlap set between Teacher and Lecturer is the mapping (1, a).

Definition 4: Semantic similarity between two subject nodes: *$SS(s_1, s_2) = |O(s_1, s_2)| / N$, where $N = |G(s_1)| + |G(s_2)| - |O(s_1, s_2)|$, and mapping (o, p) does not exist such that $o \in G(s_1)$ and $p \in D(s_2)$ or $o \in G(s_2)$ and $p \in D(s_1)$. Otherwise, $SS(s_1, s_2) = -1$.* □

We present the following algorithm to compute the similarity of two subjects, and then match subjects by using an algorithm like stable marriage [Lov86].

```
1.  Produce linguistics similarity LS(s1, s2);
2.  SS(s1,s2) = 0;
3.  For each pair of subjects
4.      For each rule in ACR
5.          If grant(s1, o, a) and grant(s2, o, a) then
6.              O(s1, s2) ← o
7.          Else if violation exists then
8.              SS(s1, s2)=-1; break;
9.          End if
10.     If (SS(s1,s2) != -1
11.         SS(s1, s2) = | O(s1, s2)|/ N;

12. If SS(s1, s2) >=0 then
13.     SIM(s1, s2) = w* LS(s1, s2) + (1-w) * SS(s1, s2);
14. Else
15.     SIM(s1, s2) = -1;
16. For each subject s1 in S1
17.     If Max ( SIM(s1, s2)) and SIM(s1,s2) >0 then
18.         Maps ← (s1, s2);
```

Algorithm 1. Subject matching

Theorem 1: Algorithm 1 generates safe mappings.

Proof: The algorithm computes the similarity between any pair of subjects in two input models based on the object mapping. Any possible violation will be identified in lines 7-8 by marking the semantic similarity as -1. Through line 10-12, the SS value will finally prevent mapping between any two violating subjects in line 17.

Therefore the algorithm generates the mapping between those pairs of subjects that have no possible violation of access control rules. According to Definition 2, the generated mapping is a safe mapping.□

Since discretionary security models are dominant in commercial applications, we concentrate on discretionary subjects matching. However, the solution presented here can also be used for matching subjects of multi-level security models.

4.4 Merge Operator with Security Property

Having the mapping of two models, one can merge two models to generate a federation and exchange information freely. The security extension of Merge eases the process by automatically generating access control rules for the merged data model. We define the Merge operator with security extension as following:

Definition 5: *$(M_3, ACR_3, Map_{1\_3}, Map_{2\_3})$ = Merge $(M_1, M_2, Map_{1\_2}, ACR_1, ACR_2, Map_a)$, where M_1 and M_2 are input data models, and Map_{1_2} represents the mapping between M_1 and M_2. Map_a represents the mapping between two access control rules ACR_1 and ACR_2.* □

A Merge operator generates M_3, Map_{1_3}, and Map_{2_3}. The result model M_3 for the previous example is shown in Figure 8. Mapped elements in M_1 and M_2 are collapsed into one element in the new model, such as Name and C_Name into Name.

Other than object merge, the security extension of the Merge operator also merges access control rules into a set of result access control rules, i.e. ACR_3. The process of merging two access control rules is called *access merge*.

Access merge is based on subject mappings. As shown in Figure 6, Map_a denotes the relationship between all possible subjects of two input access control rules. The two mapped subjects should be collapsed into one subject, such as Teacher and Lecturer into Teacher, and share the same access authorization.

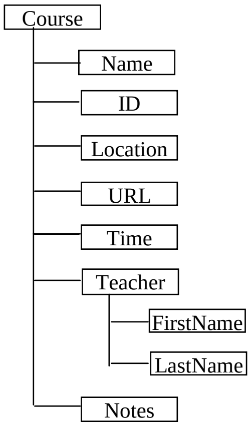


Figure 8. Result merged schema

It is easy to merge subjects and we just merge mapped subjects together and copy unrelated subjects to the output. For the example in Section 4.4, three subjects are generated in the output, i.e. Public, Student, and Teacher.

The key point of access merge is to generate new access control rules for each output subject. The access merge should insure a safe output while keep maximum access for subjects.

Suppose the Merge operator takes input models M_1 and M_2. If the mapped subjects have the same access to the same mapped objects in both models, then the related two rules can be merged to one output rule, e.g. Rule 1 of ACR_1 (in Table 1) and Rule 1 of ACR_2 (in Table 2).

Apart from the above case, three other cases need to be handled carefully as follows:

1. Unmapped subjects

Subject S_1 for M_1 is not mapped and is a new subject to M_2. Since it is only effective on the elements in M_1, we simply add the related access control rules to the result.

2. Unmapped objects

Object O_1 for M_1 is not mapped and is a new object to M_2. We simply add the two rules to the result.

3. Conflicts

The mapped rules of mapped subjects may have different authorizations, propagation rules or actions for the mapped objects. Conflict arises when two rules are merged. Whether to grant the access of the merged subject to the merged object would be a delicate issue. Our solution is to assure the maximum safe access for users, and prevent security violation caused by Merge while still being flexible and adjustable. Table 3 shows the resulting access rules for merged models.

5. Discussion and Directions for Future Research

This paper has discussed access control for model management. Essentially we have provided the foundation for work on secure model management. We have focused mainly on applying various models proposed for access control on XML documents as well as on models and schemas for model management. There are many areas that need further work as discussed below.

5.1 Formalization and Other Operators

The access control models discussed here are

Table 3. Access Control Rules for Merged Model

	Subject	Object	Action	Authorization	Propagation
1	Public	/Course/Name	Read	Grant	L
2	Public	/Course/Teacher	Read	Grant	L
3	Student	/Course/	Read	Grant	R
4	Teacher	/Course/	Read	Grant	R
5	Teacher	/Course/URL	Write	Grant	L
6	Teacher	/Course/Notes	Write	Grant	L

somewhat informal. The next step is to expand on the work proposed here and develop a formal model and prove that security properties are maintained during the mappings. The access control rules essentially control access that a user can have to the various documents. However a user can receive legitimate responses and subsequently make sensitive associations. Such a problem has come to be known as the inference problem. Extensive work has been carried out on applying security constraint processing for the inference problem [Thu95]. We need to apply intelligent inference to the access rules to achieve more personalized model management.

Other operators also need to extend with security properties, such as ModelGen. After the ModelGen operation, some objects of the original model may be removed, and the security extension of the ModelGen operator needs to adjust the access control rules for the generated model. We will extend other visual model management operators with security properties as we do in Sections 4.2 and 4.3 as our future work.

5.2 Semantic Web and Privacy

Research on semantic Web has been significantly advanced since Berners-Lee first conceived the concept (http://www.w3.org/DesignIssues/Semantic.html). Some preliminary work on security for RDF resources description framework and security for semantic Web has been conducted [Car04] and [Thu04]. Our work has discussed operations on models and schemas for XML documents and databases. A step forward will be its application to the semantic Web, addressing the security issues in real world applications such as e-business and e-services.

There has been some work on the secure publishing of XML documents [Ber04]. The idea here is to have untrusted third party publishers and yet maintain authenticity and completeness of documents when users request documents from the owner. Finally we need to consider privacy rules for model management. Privacy can be regarded to be an aspect of security where personal information is protected from unauthorized individuals. The question is can we apply the techniques developed for secure model management for privacy control?

The above are some of the challenges. Much work has been carried out on model management as well as on data and applications security. This paper attempts to merge the ideas from the two fields that will results in secure model management.

6. Related Work

Many proposals on access control mechanisms have been presented in both database literature [Gri76, San96, Jaj97, Jaj01, Jaj90, and San98] and XML area [Dam02, Dam00, and Ber02]. There are however few proposals on access controls across heterogeneous data models, and the most related work are the work on secure XML federations [Wan02] and XML security models using relational databases [Luo04]. Tan also proposed an idea of using RDBMS to handle access controls for XML documents, in a rather limited setting [Tan01]. Farkas *et al* developed algorithms to automate the access control rules transformation process, while preserving the Access Control requirements of the original systems [Far03]. They studied and developed methods to automatically translate Access Control Lists and Bell-LaPadula models to ASL. They concentrated only on the access control rules while we manipulate the related schemas at the same time.

On the other hand, various systems for model management have been presented. Cupid [Mad01, Mad03] and SFA [Mel02] match objects. Buneman *et al* described a theoretical foundation of merge [Bun92]. Pottinger *et al* presented the Merge operator based on the BDK algorithm [Pot03. Most of the approaches only concentrate on part of model management without any discussion on security issues. Rondo [Mel03] is the first complete prototype of the generic model management system. None of these proposals addresses security extensions for any model management operators.

We need to examine other types of models. For example, Sandhu and others have recently proposed a model called Usage Control (UCON) [Par04], which controls access to the usage of a document. We need to examine the applicability of such models to model management.

7. Conclusion

This paper has proposed the first security extension to model management operators. We provided uniform abstract access control rules for heterogeneous data models and a visual representation of the access control model. Having presented approaches for automatic generation of subject matching and proved that the security isomorphism algorithm generates safe mappings. The paper also discussed the security issues involved in other operators. The security extensions to our previous work on visual model management operators provides automatic generation mechanism for managing access control specifications to allow heterogeneous Web data models to exchange information over public network.

References

[Ber00] P.A.Bernstein, A. Halevy, and R.A. Pottinger, A Vision for Management of Complex Models, *SIGMOD Record,* 29(4), 55-63, 2000.

[Ber03] P.A. Bernstein, Applying Model Management to Classical Meta Data Problems, *Proc. 2003 CIDR Conference*, Asilomar, CA, Jan, 2003.

[Ber98] T. Berners-Lee, R. Fielding, U. Irvine, and L. Masinter, Unfiorm resource identifiers (URI): Generic Syntax. *http://www.isi.edu/in-notes/rfc2396.txt, 1998.*

[Ber02] E. Bertino and E. Ferrari. Secure and Selective Dissemination of XML Documents, *IEEE Trans. Information and System Security (TISSEC),* 5(3): 290 – 331, Aug. 2002.

[Bra00] T. Bray, J. Paoli, C. Sperberg-Mcqueen, and E. Maler, Extensible Markup Language (XML) 1.0 (2nd Edition), *World Wide Web Consortium (W3C),* http://www.w3.org/TR/REC-xml, 2000.

[Ber04] E. Bertino, B. Carminati, E. Ferrari, B. Thuraisingham and A. Gupta, Secure Third Party Publication of XML Documents, IEEE Trans. on Knowledge and Data Engineering. October 2004

[Bun92] P. Buneman, S.B. Davidson and A. Kosky, Theoretical Aspects of Schema Merging, *Proc. 3rd Int. Conf. Extending Database Technology*, Vienna, Austria, Mar. 1992, 152-167.

[Car04] B. Carminati, E. Ferrari, and B. Thuraisingham, Using RDF for Policy Specification and Enforcement, *Proc. of DEXA Workshops 2004*: Zaragoza, Spain, August 2004.

[Dam00] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, Securing XML Documents. *Proc. EDBT 2000 Konstanz, Germany, Lecture Notes in Computer Science*, Vol. 1777, Springer, New York, March, 2000, 121–135.

[Dam00a] E. Damiani, S. De Capitani Di Vimercati, S. Paraboschi, and P. Samarati. Design and Implementation of an Access Control Processor for XML Documents, *Computer Networks*, 33(6): 59–75, 2000.

[Dam01] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, Fine Grained Access Control for SOAP E-Services, *Proc. 10th Int. World Wide Web Conference*, Hong Kong, China, May, 2001.

[Dam02] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, A Fine-Grained Access Control System for XML Documents, *ACM Trans. Information and System Security (TISSEC),* 5(2)169-202, May 2002.

[Far03] C. Farkas, A. Stoica, P. Talekar, APTA: an Automated Policy Translation Architecture, *Int. Conf. Computer, Communication and Control Technologies*, 2003.

[Fer94] E. Fernandez, E. Gudes, and H. Song. A Model of Evaluation and Administration of Security in Object-Oriented Databases, *IEEE Trans. Knowledge and Data Engineering (TKDE)*, 6(2):275–292, 1994.

[Gri76] P. P. Griffiths and B. W. Wade, An Authorization Mechanism for a Relational Database System, *ACM Trans. Database System (TODS),* 1(3): 242 – 255, Sep. 1976.

[Jaj90] S. Jajodia and R. Sanhu, "Toward a Multilevel Secure Relational Data Model", *ACM SIGMOD*, May 1990.

[Jaj97] S. Jajodia, P. Samarati, V. S. Subrahmanian, and E. Bertino, A Unified Framework for Enforcing Multiple Access Control Policies, *ACM SIGMOD*, 474 – 485, May 1997.

[Jaj01] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian, Flexible Support for Multiple Access Control Policies, *ACM Trans. Database Systems (TODS),* 26 (2): 214 – 260, June, 2001.

[Lov86] L. Lovasz and M. Plummer, *Matching Theory*, North-Holland, Amsterdam, 1986.

[Luo04] B. Luo, D. Lee, W. Lee, P. Liu, A Flexible Framework for Architecting XML Access Control Enforcement Mechanisms, *Proc. VLDB Workshop on Secure Data Management in a Connected World (SDM),* Toronto, Canada, August 2004.

[Mad01] J. Madhavan, P. A. Bernstein, and E. Rahm, Generic Schema Matching Using Cupid, *Proc. 27th VLDB Conf.*, Roma, Italy, Sep, 2001, 49-58.

[Mad03] J. Madhavan and A. Y. Halevy, Composing Mappings Among Data Sources, *Proc. 29th VLDB Conf.*, Berlin, German, Sep 2003, 572-583.

[Mel02] S. Melnik, H. Garcia-Molina and E. Rahm: Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching, *Proc. 18th ICDE*, San Jose CA, Feb 2002.

[Mel03] S. Melnik, E. Rahm, and P. A. Bernstein, Rondo: A Programming Platform for Generic Model Management, Proc. *SIGMOD 2003 Conf.*, San Dieago, CA, June 2003, 193-204.

[Par04] J. Park, X. Zhang, and R. Sandhu, Attribute Mutability in Usage Control, *Proc. 18th IFIP WG11.3 Working Conference on Database and Application Security*, Sitges, Catalonia, Spain July 25-28, 2004.

[Pot03] R. A. Pottinger and P. A. Bernstein, Merging Models Based on Given Correspondences, *Proc. 29th VLDB Conf., Berlin*, Germany, 2003, 826-873.

[Rab91] F. Rabitti, E. Bertino, and G. Ahn. A Model of Authorization for Next-Generation Database Systems, *ACM Trans. Database Systems (TODS)*, 16(1):89–131, 1991.

[Rah01] Rahm, Erhard and P. A. Bernstein. A Survey of Approaches to Automatic Schema Matching, *VLDB Journal*, 10(4): 334-350, 2001.

[Sam96] P. Samarati, E. Bertino, and S. Jajodia, An Authorization Model for a Distributed Hypertext System, *IEEE Trans. Knowledge and Data Engineering* (TKDE), 8(4): 555–562, 1996.

[San96] R.Sandhu, E. Coyne, H. Feinstein, and C. Youman, Role-Based Access Control Models, *IEEE Computer*, 29 (2), 1996.

[San98] R. Sandhu, F. Chen, The Multilevel Relational (MLR) Data Model, *IEEE Trans. Information and System Security (TISSEC)*, 1 (1), 1998.

[Son04] G.L. Song, K. Zhang, and J. Kong, Model Management Through Graph Transformations, *Proc. 2004 IEEE Symp. Visual Languages and Human-Centric Computing*, IEEE CS Press, Rome, Italy, September 2004, 75-82.

[Tan01] K.-L. Tan, M. L. Lee, and Y. Wang. Access Control of XML Documents in Relational Database Systems, *Proc. Int. Conf. on Internet Computing (IC),* Las Vegas, NV, Jun. 2001.

[Thu95] B. Thuraisingham, and W. Ford, Security Constraint Processing in a Multilevel Secure Distributed Database Management System, *IEEE Trans. on Knowledge and Data Engineering*, April 1995

[Thu04] B. Thuraisingham, Security Standards for the Semantics Web, *Accepted by Computer Standards and Interface Journal*, 2004.

[W3c01] World Wide Web Consortium (W3C). XML path language (XPath) 2.0. *World Wide Web Consortium (W3C).* *http://www.w3c.org/TR/xpath20*, 2001.

[Wan02] L. Wang, D. Wijesekera and S. Jajodia., Towards Secure XML Federations, *Proc. 16th IFIP WG11.3 Working Conference on Database and Application Security*, July 28-31, 2002.