

Towards Access Control for Visual Web Model Management

Guanglei Song¹ Kang Zhang¹ Bhavani Thuraisingham¹ Jiannong Cao²

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.polyu.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 [10]. 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* [26], authorizations are specified on portions of a HTML document, yet no semantic context similar to that provided by XML [5] 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 [7]. Recently, the continuing demand for information sharing

has shifted interest from stand-alone XML repositories to inter-connected and large-scale cooperative XML systems [9]. As more and more Web information sharing occurs on the Internet, security of information exchange with a single Web site and among different Web sites has to be addressed.

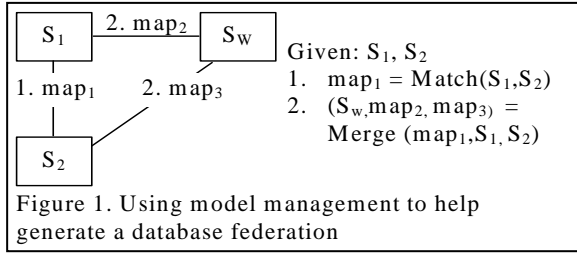
Even though every individual data model may have highly secure access control specifications and enforcement mechanism, the federation of data models is not necessarily secure. Security of a union of systems depends on the weakest link. When information of different models is interchanging, it opens 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 models without security concern. Model management is a new approach to metadata management that offers a high-level programming interface [1] 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 ease the use of model management systems [29].

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 compares related work, and Section 6 concludes the paper.

2. Model Management

Model management environment offers operators that generalize the transformation operations for various metadata applications. The main model management operators include Match, Compose, Diff, ModelGen, Merge and so on [2].

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 S_1 (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.

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.

- **XML Access Control Models**

Authorization-based XML access control models consist of a set of authorization rules. Each 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 [7, 8, 10]. An access control environment for XML documents and techniques to deal with authorization priorities and conflict resolution issues are proposed by Bertino and Ferrari [4]. Although our uniform security model is based on existing XML authorization models, 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.

- **Database Access Control Models**

Database access control models can be further classified into two categories: *multilevel security models* [14, 28] and *discretionary security models*. 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. Most database systems implement discretionary access control models similar to the one implemented in System R [13]. Role-based access control [27] is not implemented in System R but implemented by most existing DBMSs such as Oracle. Similar approaches in the context of object-oriented databases have also been presented [12, 24].

4. Security Extension to Model Management

4.1 Uniform Abstract Access Control Model

To manage access control specifications for heterogeneous data models, an abstract access control model is desirable. The development of an access control

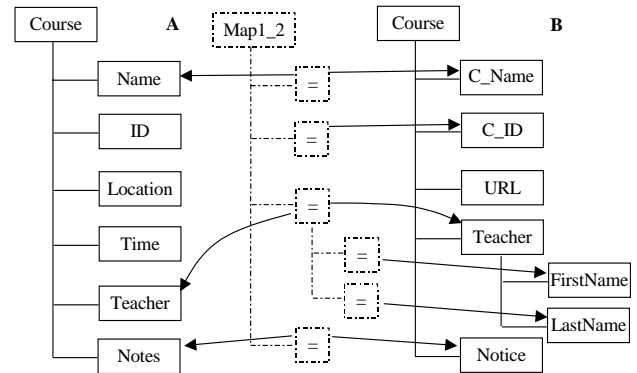


Figure 2. Course schemas for two schools

system requires the definition of subjects and objects against which authorization must be specified and access control must be enforced [10]. 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 locations from which requests originate. 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) [3] denotes the resources to be protected, i.e. objects. For XML documents, URI is extended with XPath [31], 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*. Similar 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, an 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 3 shows a visual representation of access control rules, represented by an access graph. Each rule is a link from a subject to an object. A subject is represented by a labeled rectangle connecting to objects that are represented by labeled eclipses. A gray eclipse represents recursive access, and a white eclipse indicates local access. The label of each link, R or W in the example, represents the activity. A circle and a cross on the link represent grant and deny of access respectively.

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 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.

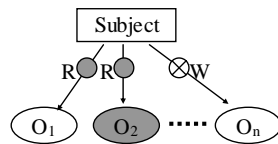


Figure 3. Visual representation of access control rules

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). Enforcement mechanisms of XML access control are surveyed by Luo *et al* [18].

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.

A model management system eases the process by generic operators like Match and Merge.

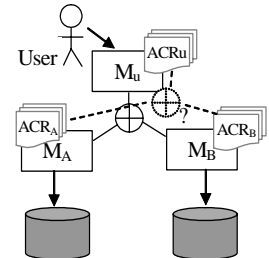


Figure 4. Unified course registration online system

Figure 4 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 a mapping of access control rules. In addition to

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

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 M_1 and M_2 respectively. \square

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 5 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 .

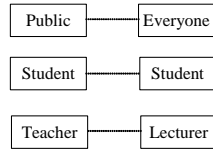


Figure 5. The Map_a example

The Match operator can be realized by a schema-matching algorithm, which has been investigated for several years. Many techniques [25], 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 5.

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.

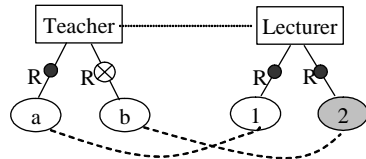


Figure 6. Graphical representation of access control rules

Models M_1 and M_2 have access control rules ACR_1 and ACR_2 respectively. S_1 and S_2 are subjects of ACR_1 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

$\forall (s_1, s_2) \in Map_s \quad \forall (o_1, o_2) \in Map_{1,2} \quad \forall a$
 $(grant(s_1, o_1, a) \leftrightarrow grant(s_2, o_2, a))$, 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 . \square

To produce safe subject mappings, we propose an approach called *security isomorphism*, which 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. 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: $O(s_1, s_2) = \{(o_1, o_2) | o_1 \in G(s_1) \text{ and } o_2 \in G(s_2), \text{ and } s_1 \text{ and } s_2 \text{ are two subjects, and } (o_1, o_2) \text{ is a mapping}\}$. \square

As shown in Figure 6 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, $SS(s_1, s_2) = -1$. \square

We present the following algorithm to compute the similarity of two subjects, and then match subjects by using an algorithm like stable marriage [17].

1. Produce linguistics similarity $LS(s_1, s_2)$;
2. $SS(s_1, s_2) = 0$;
3. For each pair of subjects
4. For each rule in ACR
5. If $grant(s_1, o, a)$ and $grant(s_2, o, a)$ then
6. $O(s_1, s_2) \leftarrow o$
7. Else if violation exists then
8. $SS(s_1, s_2) = -1$; break;
9. End if
10. If $SS(s_1, s_2) \neq -1$
11. $SS(s_1, s_2) = |O(s_1, s_2)| / N$;
12. If $SS(s_1, s_2) \geq 0$ then
13. $SIM(s_1, s_2) = w * LS(s_1, s_2) + (1-w) * SS(s_1, s_2)$;
14. Else
15. $SIM(s_1, s_2) = -1$;
16. For each subject s_1 in S_1
17. If $\text{Max}(SIM(s_1, s_2))$ and $SIM(s_1, s_2) > 0$ then
18. $Map_s \leftarrow (s_1, s_2)$;

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. \square

Though we concentrate on discretionary subjects matching, the solution presented here can also be used for matching subjects of multi-level security models.

such as Teacher and Lecturer into Teacher, and share the same access authorization.

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 in our future work.

5. Related Work

Many proposals on access control mechanisms have been presented in both database literature [13, 27, 15, 16,

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

4.4 Other Operators with Security Property

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 . \square

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 7. 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 5, 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,

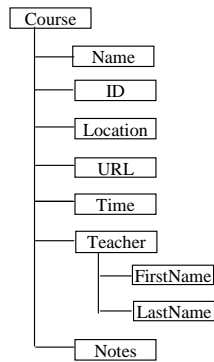


Figure 7. Result merged schema

14, 28] and XML area [10, 7, 4]. There are, however, few proposals on access controls across heterogeneous data models, and the most related work are the work on secure XML federations [32] and XML security models using relational databases [18]. Tan also proposed an idea of using RDBMS to handle access controls for XML documents, in a rather limited setting [30]. Farkas *et al* developed algorithms to automate the access control rules transformation process, while preserving the Access Control requirements of the original systems [11]. 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 [19, 20] and SFA [21] match objects. Buneman *et al* described a theoretical foundation of merge [6]. Pottinger *et al* presented the Merge operator based on the BDK algorithm [23]. Most of the approaches only concentrate on part of model management without any discussion on security issues. Rondo [22] is the first complete prototype of the generic model management system. None of these proposals addresses security extensions for any model management operators.

6. 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 networks.

References

- [1] P.A. Bernstein, A. Halevy, and R.A. Pottinger, A Vision for Management of Complex Models, *SIGMOD Record*, 29(4), 55-63, 2000.
- [2] P.A. Bernstein, Applying Model Management to Classical Meta Data Problems, *Proc. 2003 CIDR Conference*, Asilomar, CA, Jan, 2003.
- [3] T. Berners-Lee, R. Fielding, U. Irvine, and L. Masinter, Uniform resource identifiers (URI): Generic Syntax, <http://www.isi.edu/in-notes/rfc2396.txt>, 1998.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] C. Farkas, A. Stoica, P. Talekar, APTA: an Automated Policy Translation Architecture, *Int. Conf. Computer, Communication and Control Technologies*, 2003.
- [12] 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.
- [13] 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.
- [14] S. Jajodia and R. Sanhu, "Toward a Multilevel Secure Relational Data Model", *ACM SIGMOD*, May 1990.
- [15] 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.
- [16] 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.
- [17] L. Lovasz and M. Plummer, *Matching Theory*, North-Holland, Amsterdam, 1986.
- [18] 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.
- [19] J. Madhavan, P. A. Bernstein, and E. Rahm, Generic Schema Matching Using Cupid, *Proc. 27th VLDB Conf.*, Roma, Italy, Sep. 2001, 49-58.
- [20] J. Madhavan and A. Y. Halevy, Composing Mappings Among Data Sources, *Proc. 29th VLDB Conf.*, Berlin, German, Sep 2003, 572-583.
- [21] 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.
- [22] S. Melnik, E. Rahm, and P. A. Bernstein, Rondo: A Programming Platform for Generic Model Management, *Proc. SIGMOD 2003 Conf.*, San Diego, CA, June 2003, 193-204.
- [23] R. A. Pottinger and P. A. Bernstein, Merging Models Based on Given Correspondences, *Proc. 29th VLDB Conf., Berlin*, Germany, 2003, 826-873.
- [24] 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.
- [25] Rahm, Erhard and P. A. Bernstein. A Survey of Approaches to Automatic Schema Matching, *VLDB Journal*, 10(4): 334-350, 2001.
- [26] 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.
- [27] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, Role-Based Access Control Models, *IEEE Computer*, 29 (2), 1996.
- [28] R. Sandhu, F. Chen, The Multilevel Relational (MLR) Data Model, *IEEE Trans. Information and System Security (TISSEC)*, 1 (1), 1998.
- [29] 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.
- [30] 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.
- [31] World Wide Web Consortium (W3C). XML path language (XPath) 2.0. *World Wide Web Consortium (W3C)*. <http://www.w3c.org/TR/xpath20>, 2001.
- [32] 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.