

Mediation Security Specification and Enforcement for Heterogeneous Databases

Li Yang, Raimund K. Ege
School of Computer Science
Florida International University
Miami, FL 33199, USA
lyang03|ege@cs.fiu.edu

Huiqun Yu
Department of Computer Science
East China University of Sci & Tech
Shanghai 200237, China
yhq@ecust.edu.cn

ABSTRACT

Information sharing among heterogeneous databases requires increasing attention and poses a pressing need. Successful protection of information by an effective access control mechanism is a basic requirement for interoperation among heterogeneous data sources. However, most existing work on access control focuses on access control model and security related information, little effort is devoted to exploring access control on the semantic-related heterogeneous databases. No current work resolves the security challenges faced by heterogeneous databases, i.e., context-aware authorization, semantic heterogeneity of databases and policy specification at different points. This paper proposes a mediation security system that secures the information sharing among heterogeneous databases with the promise to meet the above requirements. The proposed system models the flexible access control requirements specific to mediation security while providing uniform access for heterogeneous data sources.

Categories and Subject Descriptors

K.6.5 [Management of Computing and Information Systems]: Security and Protection; D.4.6 [Operation System]: Security and Protection—*Access controls*

General Terms

Security, Design

Keywords

Mediation, security, RBAC, XML

1. INTRODUCTION

More and more individuals and organizations, whose data sources are heterogeneous, collaborate through mediation technique to serve the client applications. The users' access

to their authorized information should be permitted, however, the access to the unauthorized information should be denied. For the safety reason, the latter is more important. Mediation security system is designed to cope with security issues in collaborative computing environment. Mediators are intelligent middlewares that sit between information system clients and sources. They perform functions such as integrating domain-specific data from multiple sources, reducing data to an appropriate level and restructuring the results into object-oriented structure. Security checking is interposed between external accessors and data sources to be protected. We formalize the role of the mediation security that has the responsibility and the authority to assure that no inappropriate information leaks an enterprise domain.

Only when the access constraints on information from different databases are assured, the effective information sharing and interoperation can be possible. Regarding to the security issues, the major requirements of mediation efforts can be summarized as: (1) Users of an application should be able to access information from any or all of the sources in a uniform and consistent way. Specialized knowledge of the individual sources should not be required. (2) Users of an application must be denied access to any information that is not allowed from either global level or source level. In other words, both the global and local policies must be respected. (3) Access control is context-aware.

Although several projects have been carried out to support fine-grained access control in context-based or multiple policies environment, none of the existing work satisfies the above requirements for heterogeneous databases. Authorization and access control mechanisms available today are at preliminary stage [4]. In [5], Dawson presented a modeling and architectural solution to the problem of providing interoperation while preserving autonomy and security of the local sources based on the use of wrappers and a mediator. But in the authorization they did not consider context information that frequently determines the access control especially in the information sharing under the interconnected environment. Their work was based on the MAC model that supports less flexibility compared with role-based access control (RBAC) model [13, 9] in our system. E. Damiani et al. [4] exploited XML's own capabilities, allowed the definition and enforcement of access restrictions directly on the structure and content of the documents. The algorithm to compute the user visible view of XML document based on subject's authorization rules was presented with DOM tree labeling and transformation, which is closer to our work.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'05 March 13-17, 2005, Santa Fe, New Mexico, USA
Copyright 2005 ACM 1-58113-964-0/05/0003 ...\$5.00.

Author-X [7] is the implementation of [4], and both of them base on DAC model. A. Gabillon [1] used XPath language to address XML fragments while regulating XML document access, offering the possibility of defining content-based authorization rules. XACL [10] integrated security features such as authorization, non-repudiation, confidentiality and an audit trail for XML document. However, above four related works disregard the pressing need for interoperability and information sharing among databases, especially the semantic-related heterogeneous data sources. Instead of creating more roles based on the context variation in W. Yao [17], we associate the context constraints with the permission assignment in order to avoid the role explosion [2].

In this paper we show how to secure information upon interoperability in the context of a distributed mediation system where access to data is enforced by RBAC, and the security policy specification at different data sources may differ. Our objective is to allow data from heterogeneous data sources available to external applications in such a way that their autonomy and security are not compromised.

This paper is organized as follows: Section 2 presents the mediator specification for interoperability in mediation system. Section 3 proposes the security policy specification applied to mediation system. Section 4 explains the access control enforcement in mediation system. Section 5 is the conclusion.

2. MEDIATOR SPECIFICATION

R.K.Ege et al. [8] and L.Yang et al. [16] proposed the adaptive three-layered mediator architecture as well as the mediation security enforcement architecture. In order to achieve unified identification of mediation object, the transformation from heterogeneous data model to homogeneous data model is performed by wrappers. But the semantic heterogeneity still exists in both terminology and structural aspects. A *mediation system* is defined and how to resolve the semantic gap among heterogeneous sources is elucidated. With the promise of integration heterogeneous databases, a mediation system includes three parts: a *global (mediated) schema type*, a set of *source schemas type* and *mapping relation*. A *global schema type* is tree type whose labels are terms of a global vocabulary, distinct from source schema type used in the labels defining local data schema. The global schema vocabulary has been chosen to unify the local vocabulary and represent a specific domain of interest.

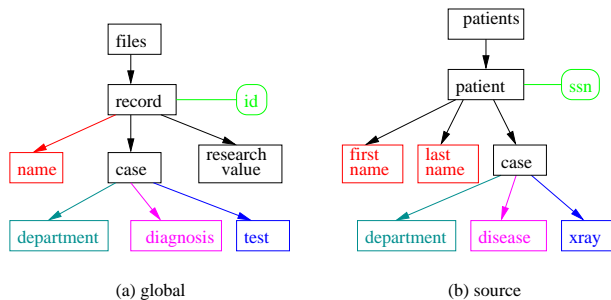


Figure 1: Mapping between global view and source view

DEFINITION 2.1 (MEDIATION SYSTEMS). A mediation sys-

tem M is a triple $(G, \{L_i\}, \{M_i\})$, where G is a global schema, $\{L_i\}$ is a set of

enterprise management. Moreover, DAC and MAC can be configured to RBAC [14]. The RBAC model has the following main components [12]:

1. U, R, P and S (users, roles, permissions and sessions respectively), where P is the Cartesian product of operation OP and objects Obj ,
2. $PA \subseteq P \times R$, a many-to-many permission to role assignment relation,
3. $UA \subseteq U \times R$, a many-to-many user to role assignment relation,
4. $user_sessions : U \rightarrow 2^S$, a function mapping each user u to a set of sessions.
5. $session_roles : S \rightarrow 2^R$, a function mapping each session s to a set of roles $session_roles(s) \subseteq \{r | (user(s), r) \in UA\}$ (which can change with time) and session s has the permissions $\bigcup_{r \in session_roles(s)} \{p | (p, r) \in PA\}$.

3.2 Mediation Security Policy Specification

In the mediation system, particular requirements occur: Firstly, multiple heterogeneous data sources participate in the system, and the autonomy of security policy specification at each source level is allowed. That means security policy specification should support security policy specification at different points, i.e., global level, source level. Secondly, we face an interactive and interconnected system where context information like time, location, access history, for instance, frequently determines the access control decision [11]. Therefore, in order to enforce the context-aware and fine-grained access control policies, permissions and permission assignment should support context information like the attribute from user, from object to be accessed, or from the relationship between user and object. Based on above requirements generated from mediation system, the basic RBAC model is extended to increase the flexibility. As indicated in Figure 2, two fields are added to permissions component from basic model and context information is included in the constraints applied on permission assignment.

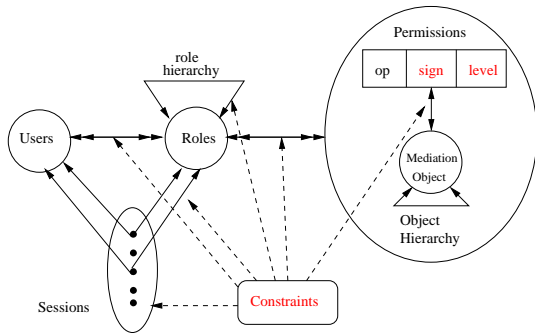


Figure 2: An extended RBAC model

We use XML, a standard for semi-structured data representation and transmission, as the exchange model to provide interoperability among heterogeneous data sources. Wrapper techniques are employed to transform the heterogeneous data models to homogeneous XML model. A W3C proposal

is used to identify the internal components of an XML document, namely, the XPath language [3].

DEFINITION 3.1 (MEDIATION OBJECT (MO)). A mediation object is an XML schema or schema component(s), patterned by an XPath expression based on global schema at global level; and patterned by XPath based on source schema at source level, under the assumption that XML object (instance) is defined by XML schema.

DEFINITION 3.2 (AUTHORIZATION REQUEST). An authorization request $\langle subj, op, obj, cxt \rangle$ refers to a query as to whether or not the subject $subj$ is permitted to perform the operation op on the object obj under the context cxt that is any data captured at the time of the access request for the specific action.

DEFINITION 3.3 (ACCESS POLICY). An access policy is a conditional permission assignment defined as $\langle role \rangle, \langle op \rangle, \langle obj \rangle, \langle level \rangle, \langle sign \rangle, \langle context constraints \rangle$, where $role \in ROLES$, $op \in OPERATION$, $obj \in OBJECT$, $level \in \{global, source\}$ and $sign \in \{+, -\}$ denotes grant and deny respectively. Access is granted if and only if the context constraints are evaluated to be "true".

Note the context attribute entry cxt in authorization request is different from the *context constraints* in access policy. The former is either a certain property of the *user* or a specific property of the environment, i.e, time. The latter specifies that certain context attributes must meet certain conditions in order to permit a s specific operation [11], i.e., access is permitted on the condition that the access is in the range of [9:00am - 1:00pm].

The following is the example of an access policy:

1. (doctor, files/record/*, global, +, work_time & at_hospital & joint_dept)
2. (nurse, files/record/research_value, global, -, joint_dept)
3. (family-member, files/record/research_value, global, -)
4. (patient, files/record/ research_value, global, -)
5. (nurse, files/record/case/name, global, +, joint_dept)
6. (nurse, files/record./case/diagnosis, global, +, joint_dept)
7. (nurse, files/recocrd/case/test, global, +, joint_department)
8. (family-member, files/record/casename, global, +)
9. (family-member, files/record/case/diagnosis, global, +)
10. (family-member, files/record/case/test, global, +)
11. (patient, files/record/case/name, global, +)
12. (patient, files/record/case/diagnosis, global, +)
13. (patient, files/record/case/test, global, +)
14. (doctor, patients/patient/*, source1, +, work_time & at_hospital & joint_dept)
15. (nurse, patients//disease, source1, -)
16. (patient, patients//disease, source1, -)
17. (patient, patiens//xray, source1, -)

Rules 1 - 13 are at the global level, whereas rules 14 - 17 are at the source level, which provide the flexibility of policy specification. Rule 1 stipulates that at global level *doctor* can see all the patient records in both his department and his joint department in the hospital at work time. Rule 2 stipulates that at global level *nurses* are not allowed to view the *research_value* node in patient record from the joint department. Rule 3 stipulates that at global level *family members* are not allowed to view the *research_value* node in patient record. Rule 4 stipulates that at global level *patients* are not allowed to view the *research_value* node in patient

record. Rule 5 - 7 stipulate that at global level *nurses* are allowed to view the *name, diagnosis, test* node in patient record from the joint department. Rule 8 - 10 stipulate that at global level *family members* are allowed to view the *name, diagnosis, test* node in patient record. Rule 11 - 13 stipulate that at global level *patients* are allowed to view the *name, diagnosis, test* node in patient record. At source 1, Rule 14 stipulates that at source 1 *doctor* can see all the patient record. Rule 15 stipulates that at source 1, the *nurses* from the joint department are not allowed to view the *diagnosis* node in the patient record. Rule 16, 17 stipulate that at source 1, the *patients* are not allowed to view the *diagnosis* and *test* node in the patient record.

Suppose we are in the hospital system, the participating users can play the doctor, nurse, family member and patient roles. User-to-role assignment denoted by $\{ \langle Martha, doctor \rangle, \langle Mary, nurse \rangle \}$. The following example intuitively illustrates the access decision in the context-aware, multiple policy environment.

Example If a user *Martha* from department 1 requests to view the *diagnosis* node of the patient record in source 1. According to role assignment, *Martha* is assigned the *doctor* role. The applicable rules are rule 1 and rule 14 at global level and source 1 respectively. Based on the rules the access could be granted on the condition that context constraints are satisfied, that is during work_time, at hospital, the doctor's department is the patient's joint department. The constraint functions *time()*, *location()* and *joint()* are called. The functions *time()* and *location()* are environment functions obtained by system built-in functions. From the user's credential, the user attribute *department 1* is returned, and from the medical record object, the object attribute *department 2* is returned. If *department 1* and *department 2* are joint department, the *joint()* function returns true and *Martha's* access is granted; and *Martha's* access is denied otherwise.

4. ACCESS CONTROL ENFORCEMENT IN MEDIATION SYSTEM

In this section we show how to enforce security for heterogeneous data source, which distinguishes us from the related work and makes our work more relevant to the real-world situation. Also different from [4, 1, 7] where the context information was not considered, our work discusses how to enforce security at different level on one data sources under certain context.

To allow security policy specification at different points, we have both global level and source level access control. This "hybrid" approach, in contrast to global-only or source-only controls, ensures the autonomy and security of the sources while avoiding unnecessary processing and forwarding of queries. When end users pose their query against the global schema which has the form of a global schema tree. Such global query are processed by the access decision algorithm, with unauthorized nodes removed. Then the secured global query is translated into source queries based on the mapping relations specified in mediator specification.

In order to protect the source data at the global level, we need to transform source data into a secure representation as shown by M. Kudo in [10]. A more general treatment of source-to-target data transformation was *valid interpretation* in [15]. Assume that S_i be a source schema in the

mediator M . A *secure interpretation* D_{S_i} for S_i is a reformation of source data in S_i such that each datum in D_{S_i} satisfies the security requirement of S_i . We use $D_{S_i, G}$ to denote the data representation in global level w.r.t. D_{S_i} , which means: (1) it is a secure interpretation of S_i , and (2) it satisfies the mapping M_i between S_i and G . We call the collection of all data representation in global level in a mediator M as *domain of the mediator*, denoted by $dom(M)$. Intuitively, $dom(M)$ contains data allowed in a predefined global schema M retrieved from relevant heterogeneous information sources.

At each source, access control may be enforced at finer level that is currently supported by the application, depending on the capabilities of the particular source. The query process algorithm is illustrated in Figure 3. In the algorithm, any source query generated by the mediator from an application query is issued to the (wrapper of the) source together with the role and context information. Each mediator that takes care of the sub-query to the local source, will check against the source policy base and enforce access control at source-level and answers the sub-query in the similar with as those in global level. This means that the source security policies, instead of global security policies, are enforced, thus providing autonomy of security policy specification at each data source.

```

input: <usr, op, obj.xml, cxt>
available sources: data sources, mediator specification M[i],
                  global and source policy bases, i.e., ROLES, UA, PA
output: answer tree
1. globalQuery.xml = AccessDecision(usr, op, obj.xml, cxt)
2. ArrayList sourceQueryList = new ArrayList()
3. ArrayList result = new ArrayList()
4. for each mediator M_i
5.   sourceQuery.xml = reformulate(globalQuery.xml, M[i])
6.   (sourceQuery.xml).DBNumber = i
7.   sourceQueryList.add(sourceQuery.xml)
8. endfor
9. for (int j = 0, j < sourceQueryList.size(), j++)
10.  localQuery.xml = sourceQueryList.get(j)
11.  int number = (localQuery.xml).getDBNumber()
12.  securedLocalQuery.xml = AccessDecision(usr, op, localQuery.xml, cxt)
13.  connect source[number]
14.  localResult = executeQuery (securedLocalQuery.xml, source[number])
15.  result.add(localResult)
16. endfor
17. return result:

```

Figure 3: Query process algorithm

At both levels, the access decision algorithm (as in Figure 4) is called to remove the unauthorized node from the query tree. The algorithm shares common idea with [4], but we use RBAC as the access control model and takes the context information into consideration, making the access decision adaptive to the environment.

We are implementing the mediation security system based on the access control algorithm. Both the security policies and mediator specification are defined in XML files. Security policies include the authorization related information, i.e., the users, roles and permissions. XML data model is employed as the *exchange model* among heterogeneous data bases. So XML parsers (DOM and SAX) are applied to help build up and process the trees that represent the parsed documents.

AccessDecision(usr, op, obj.xml, cxt)

input: (usr, op, obj.xml, cxt), where obj.xml is the request tree
available resources: global and source policy bases for ROLES, UA, PA

output: authorized tree

```
1. cred = getCredential(usr);    //from the user profile
   //find match in user-to-role assignment
2. roles = getRoles(cred, ROLES, UA)
   //parse and generate instance tree t rooted in root of obj.xml
3. root = parse(obj.xml)
   //check the object's nodes permission based on policy base
4. function checkPermission(roles, op, root, cxt)
5. {
6.   for each r in roles
7.     /*
8.      * find the matched policies in access policies,
9.      * e.g. <r, op, obj, cond1&cond2>
10.     * and the cond1 and cond2 are the context constraints
11.     */
12.     constraints = getAccessPolicy(r, op, root, PA)
13.     /*
14.      * check whether request context cxt conforms
15.      * to the context constraints in access policy
16.      */
17.     if (getContextValue(cxt).satisfies(constraints) )
18.       Access to root is granted
19.     endif
20.   endfor
21.   if (root is not leaf)
22.     for each sub-tree rooted in root of obj.xml
23.       checkPermission(roles, op, sub, cxt)
24.     end for
25.   end if
26. } // end function checkPermission
27. remove the not granted nodes
28. return the authorized query tree
```

Figure 4: Access decision algorithm

5. CONCLUSION

The paper has proposed a mediation security system that provides secure interoperability among heterogeneous data sources, and allows the autonomy of heterogeneous data sources. The basic RBAC model is extended to model the flexible context-aware access control in mediation system. What distinguishes our work from the related work is the security enforcement to heterogeneous databases at different level.

We restricted ourselves to the *read* privilege, that is retrieving data from heterogeneous databases. Indeed, the read privilege is the most important privilege to consider data retrieval from heterogeneous sources. However, we are extending our model with the *write* privilege and model the related collaborative issues, i.e., information flow, task flow.

Acknowledgments: The authors are grateful to the anonymous reviewers for helpful comments. This work was partially supported by the NSF of the USA under grant HRD-0317692, and by the NSF of China under grant No. 60473055.

6. REFERENCES

- [1] A.Gabillon and E.Bruno. Regulating access to XML documents. In *Proceedings of the fifteenth annual working conference on Database and application security*, pages 299–314. Kluwer Academic Publishers, 2002.
- [2] J. Barkley, K. Beznosov, and J. Uppal. Supporting relationships in access control using role based access control. In *Proceedings of the fourth ACM workshop on Role-based access control*, pages 55–65. ACM Press, 1999.
- [3] J. Clark. XML path language (XPath) version 1.0. In *World Wide Web Consortium (W3C)*, 1999.
- [4] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. A fine-grained access control system for XML documents. *ACM Transaction Information System Security*, 5(2):169–202, 2002.
- [5] S. Dawson, S. Qian, and P. Samarati. Providing security and interoperation of heterogeneous systems. *Distributed and Parallel Databases*, 8(1):119–145, 2000.
- [6] C. Delobel and M. C. Rousset. A uniform approach for querying large tree-structured data through a mediated schema. In *Foundations of Models For Information Integration Workshop (FMII)*, 2001.
- [7] E. Bertino, S. Castano, and E. Ferrari. Securing XML documents with Author-X. *IEEE Internet Computing*, 5(3):21–31, 2001.
- [8] R. K. Ege, L. Yang, Q. Kharm, and X. Ni. Three-layered mediator architecture based on DHT. In *International Symposium on Parallel Architecture, Algorithm and Networks (I-SPAN)*, Hong Kong, China, May 2004. IEEE press.
- [9] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. Proposed NIST standard for role-based access control. *ACM Transaction Information System Security*, 4(3):224–274, 2001.
- [10] M. Kudo and S. Hada. XML document security based on provisional authorization. In *Proceedings of the 7th ACM conference on Computer and communications security*, pages 87–96. ACM Press, 2000.
- [11] G. Neumann and M. Strembeck. An approach to engineer and enforce context constraints in an RBAC environment. In *Proceedings of 8th ACM Symposium on Access Control Models and Technologies (SACMAT)*, June 2003.
- [12] R. Sandhu, D. Ferraiolo, and R. Kuhn. The NIST model for role-based access control: Towards a unified standard. pages 47–64.
- [13] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.
- [14] S. Osborn, R. Sandhu, and Q. Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transaction. Information System Security*, 3(2):85–106, 2000.
- [15] L. Xu and D. W. Embley. Using schema mapping to facilitate data integration. 2003.
- [16] L. Yang and R. K. Ege. A role-based access control model for information mediation. In *2004 IEEE International Conference on Information Reuse and Integration (IEEE IRI-2004)*, Las Vegas, Nevada, USA, November 2004. IEEE press.
- [17] W. Yao, K. Moody, and J. Bacon. A model of OASIS role-based access control and its support for active security. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 171–181. ACM Press, 2001.