Using usage control to access XML databases

Lili Sun, Yan Li

Abstract

XML documents usually contain private information that cannot be shared by all user communities. It has been widely used in web environment. XML database is becoming increasingly important since it consists of XML documents. Several applications for supporting a selective access to data are available over the web. Usage control has been considered as the next generation access control model with distinguishing properties of decision continuity. It has been proven efficient to improve security administration with flexible authorization management. Objects-oriented database systems represent complex data structure and XML databases may be stored in the objects-oriented database system. Therefore authorization models for XML databases could be used the same the models as object-oriented databases. In this paper, we propose usage control models to access XML databases and compare with a methodology designed for object-oriented databases. We have analysed the characteristics of various access authorizations and presented detailed models for different kinds of authorizations. Finally, comparisons with related works are analysed.

Key words: *XML, XML database, XML Schema, Usage access, Authorization*

1. Introduction

The extensible markup language (XML) is a standard for describing the structure of information and content on the Internet over the past several years. XML has recently [1] emerged as the most relevant standardization in the area of document representation through markup language [1]. XML is used to store and exchange data in the Internet environment that may include private messages of customers. It overcomes the complexity of Standard Generalized Markup Language (SGML) and the user can define document structures, removing the limit of the fixed tags in Hypertext Markup Language (HTML). XML documents support the information which is at different degrees of sensitivity and varying granularity levels.

We identify two levels of instance and the Document Type Definition (DTD) at which authorizations on XML documents can be defined [3, 6]. A DTD is a file which contains a formal definition of a particular type of XML documents. A DTD consists of two parts: the element declarations and the attributes declarations. Instance level authorizations denote privileges that only to a specific document. DTD level authorizations specify the privileges of all documents following a given DTD. XML Schema is an XML-based alternative to DTD. It supports complex constraints for XML components, such as elements, attributes, datatypes and groups. A well-validated XML document must follow the format specified by one or several schemas. In the access control model the central authority uses XML schemas to specify the format of information to be changed. With the features of XML Schema, a

[1] This is the authors' final corrected version of the paper published as: Sun, Lili and Li, Yan (2009) *Using usage control to access XML databases.* International Journal of Information Systems in the Service Sector, 1 (3). pp. 32-44. ISSN 1935-5688

flexible and easy-customized access control model can be achieved.

Access control has been considered as a major issue in information security community since the beginning of the information security discipline [14]. Through access control, the system can restrict unauthorized users access to the resources in the system and guarantees the confidentiality and integrity of the resources. Manage access control to database or other collections of structured data, the traditional access control models, the discretionary and mandatory access control [7, 8, 12] have been augmented by various research groups. Usage control is a new access control model which extends traditional access control models and other access control models in many aspects. The term "usage" means usage of rights on digital objects. The main different properties of usage control with traditional access control models are continuity of access decision and mutability of subject attributes and object attributes [18].

A recent development in the database field has been the introduction of semi-structured and self-describing data, collecting the data which are an XML format calling XML Databases [20]. Some work [17, 20] discussing the relationship between securing XML documents and object oriented databases (OODB) has been done. From literature review, however, we have not found a detailed discussion of how the usage access model can be applied to XML databases. In this paper, we propose authorization models which adopt usage control to manage access XML based databases. Traditional access control, such as the discretionary and mandatory access control, focused on the control of access to server-side objects. They give an access request and an algorithm which computes a view of the target XML document based on the user's requirement's right. They have analyzed authorization decisions on a subject's access to target resources before access. However, usage access control authorization decisions are not only checked and made before access, but also are repeatly checked during the access period. Meanwhile obligations and conditions become decision factors for the management of XML documents.

The remainder of this paper is organized as follows: Section 2 illustrates the background of XML, XML databases. The usage control model and continuity properties are introduced in this section. Section 3 has a view of the OODB authorization model. Section 4 shows our proposed authorization models for usage control using XML databases. It includes six models of *pre-Authorizations, ongoing-Authorizations, pre-Obligations, ongoing-Obligations, pre-Conditions and ongoing-Conditions*. Section 5 reviews the differences between this works from others. Finally, Section 6 concludes the paper.

2. Background

2.1 XML

XML [5] is a markup language for describing semi-structured information. Semi-structured data is just data that does not fit neatly into the relational model. In XML, data can have an elaborate and intricate structure that is significantly richer and more complex than a table of rows and columns. An XML makes possible capturing and expressing the structure of the data as we understand it, without forcing it into a too-simple structure. XML documents can be classified into two categories: well-formed and valid. A document is said to be well-formed if it follows the grammar rules of XML, for example, there is exactly one element that completely contains all other elements or elements may nest but not overlapped, *etc*. A well-formed document is valid only if it contains a proper DTD in the source and if the document obeys the constraints of that declaration. On the other hand, since XML is a structural transformation, it can transform one structure to another structure. An

example of XML document containing information on a company staff is shown in Table 1.

XML documents not only show the contents of data but also the constraints and relationships between data in Table 1. The element *GeneralInfo* includes *Name, address,* and *email* elements, and *GeneralInfo* element is a sub-element of *staff*. Since an XML document can express complex relationship between data, it can satisfy with varying security requirements. XML is used to store and exchange data in the Internet including private messages. Some users may like to access some particular parts of an XML document. In the above example in Table 1, for *staff* objects everyone can read general information such as *name*, *address*, *email* and so on. However, the staff financial information will be restricted. Therefore the user access permission has to be limited according to security policies. This example shows that securing XML document forms a significant topic for research.

```
<?xml version= "1.0" encoding= "UTF8"?>
   <StaffInfo xmlns=
      "http://www.company.com/StaffInfo">
     <company name= "computer company">
      <staff StaffId="12345">
       <GeneralInfo>
        <name> Tony Mahanee </name>
        <address> 1 Smart Street </address>
        <email> Tony@hotmail.com</email>
       </GeneralInfo>
       <WorkInfo>
          <workarea>implement</workarea>
          <developarea>research
          </developarea>
       </WorkInfo>
       <FinancialInfo>$3800</FinancialInfo>
      </staff>
    </staffInfo>
```

Table 1: XML Document Example

Document Type Definition (DTD) and XML Schema are two main validation specification mechanisms [2, 19]. They can be attached to XML documents, specifying the rules that XML documents may follow. An XML Schema is an XML-based alternative to DTD [23]. XML Schemas provide a means for defining the structures, contents and semantics of XML documents. XML Schemas are extensible to future additions. XML Schemas are richer and more powerful than DTDs. The example below in Table 2 displays an XML Schema for a corresponding valid XML instance in Table 1.

```
<?xml version= "1.0"  encoding= "UTF-8"? >
   <xs:schema
        targetNamespace="http://www.company.com/StaffInfo"
        xmlns:xs= "http://www.w3.org/2001/XMLSchema"
        elementFormDefault= "qualified">
     <xs:annotation>
        <xs:documentation>
           Staff Information Instance
        </xs:documentation>
     </xs:annotation>
   <xs:element name= "StaffInfo">
    <xs:sequence>
      <xs:element name= "staff"  type= "xs:string"/>
        <xs:complexType name= "GeneralInfo"/>
            <xs:sequence>
              <xs:element name= "name"
                          type= "xs:string"/>
               <xs:element name= "address"
                           type= "xs:string"/>
              <xs:element name= "email"
                           type= "xs:string"/>
            </xs:sequence>
          </xs:complexType>
       <xs:complexType name= "WorkInfo">
        <xs:sequence>
          <xs:element name="workarea" type= "xs:string"/>
         <xs:element name= "developarea"  type= "xs:string"/>
        </xs:sequence>
       </xs:complexType>
       <xs:element name= "FinancialInfo"
                          type= "xs:string"/>
       <xs:attribute name= "StaffId"  type= "xs:string"/>
</xs:schema>
```

Table 2: XML Schema Example

2.2 XML and Databases

An XML document is a collection of data. It is a self-describable, exchangeable and a tree graphic structure description data set. XML documents fall

into two categories: *data-centric* and *document-centric* [9]. Data-centric documents are those where XML is used as a data transport. For example, dynamic Web pages are a special case of data-centric documents. Document-centric documents are documents that are designed for human read. Examples are books, emails and advertisements. They are characterized as irregular structures and mixed contents.

To store and retrieve data in data-centric documents, you need to know how well structured your data is. For highly structured data, you will use an XML-enabled database for data storage, such as a relational or object-oriented database, and some sort of data transfer software such as middleware [4]. If your data is semi-structured, you may have two choices. You can fit your data into a well-structured database, such as a relational database, or you can store it in a native XML database. The native XML database is specialized for storing XML data and stores all components of the XML model intact [6]. To store and retrieve document-centric documents, you will need a native XML database. Some native XML database models are stored in the relational and object-oriented databases. For example, in the relational database storage Document Object Model (DOM), there will be elements, attributes, PCDATA, entity, and other entities cited forms. As traditional databases add native XML capabilities and native XML databases support the storage of document fragments in external (usually relational) databases, the access control models for traditional databases, such as a relational or object-oriented databases and native XML databases could be used the same way.

2.3 Usage control

The usage control is a generalization of access control. It enriches and refines the access control areas in its definition and covers obligations, conditions, continuity (ongoing controls) and mutability [14, 22]. There are eight core components in the usage control model: subjects, subject attributes, objects, object attributes, rights, authorizations, obligations, and conditions (see Figure 1). Subjects and objects are familiar concepts with the tradition access control. A right is used for accessing a subject to an object in a mode, such as read or write. Subject and object attributes can be used during the access decision process. Subject attributes are identities, group names, roles, memberships, security clearance, and so on. Objects are entities that subjects hold rights on, whereby the subjects can access or use objects. For instance, in an on-line shopping store, a customer can be subject. A price could be an object attribute. For example, a story book with DVD is priced at $38 and with delivery the price is required at $48. Rights are privileges that subjects can hold on objects. The authorizations of rights require associations with subjects and objects. A right represents the access of a subject to an object, such as read or write.

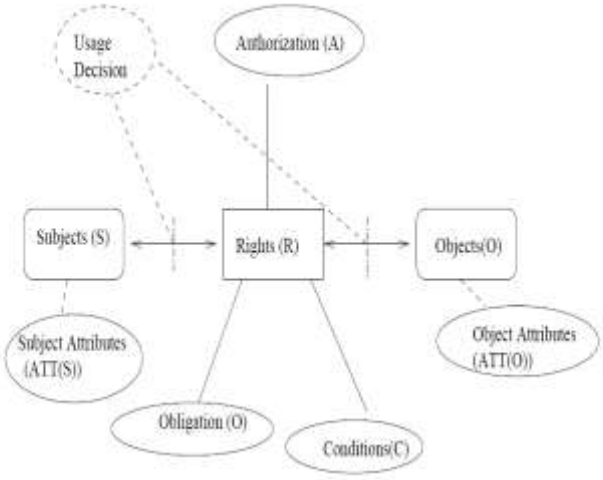

Figure 1: Components of Usage Control Model

The authorization, obligations and conditions are main components of usage control decisions. In the usage control model, the authorization rule permits or denies the access of a subject to an object based on

subject and object attributes. Obligations are performed by subjects or by the system. Conditions are not related to subject or object attributes. They are system environment restrictions.

Authorizations, obligations and conditions are decision factors used to check and determine whether a subject should be allowed to access an object. Obligations and conditions are new concepts that can resolve certain shortcomings that have been in traditional access controls. In general, the authorization of most traditional access controls, such as mandatory, discretionary and role based access control are assumed to be done before access is allowed [18]. However in the usage control model it extends this for continuous enforcement. Authorizations may require updates on subject and object attributes. The process of continuity properties in usage control model consists of three phrases: before usage, ongoing usage and after usage. To enforce control decisions, we have two different types: pre-decision and ongoing-decision. For mutability, there are three kinds of updates: pre-update, ongoing-update, and post-update. Therefore, Authorizations can be either pre-authorization (preA) or ongoing-authorization (onA). Pre-authorization is performed before authorization is required to the access. But ongoing authorization may be performed during the access, such as when a book stocking list in a bookstore is periodically checked while the access is in progress. Obligations are requirements that a subject must perform before (pre) or during (ongoing) accesses. Conditions are decision factors that depend on environmental and system-oriented requirements. Subject and object attributes can be used to select which condition requirements have to be used for a request.

Based on the involvement of three decision factors: authorizations, obligations, and conditions, we focus on developing usage control models for XML databases. This involves XML Schemas and XML documents. We assume that a usage request exists on an XML target object. Decision-making can be done before (pre), during (ongoing) or after the exercise of the requested right. Based on the requirements we have six possible cases as a model for usage control: pre-Authorizations, ongoing-Authorizations, pre-Obligations, ongoing-Obligations, pre-Conditions and ongoing-Conditions.

3. The OODB authorization model

Object-oriented database systems (OODB) [16] are an important emerging technology for applications in business, industry, and many other areas. OODB is the most popular data model to represent complex data structure [10]. XML database is a standard for representing semi-structured data. Schemas of XML databases can represent dynamic data structure, such as list, tree and graph. Since OODB and XML databases are suitable to represent complex objects. They can have the same authorization models. Therefore, the OODB authorization model can be applied to XML Schema and documents [23]. The OODB authorization model presented by Rabitti et al [16] is a discretionary access control model for object-oriented database. It models an authorization as a triple:

$$f: S \times O \times A \rightarrow \text{(True, False)}$$

Where S represents the set of subjects, O represents the set of objects and A is the possible authorization types (access modes) in a system. The models of authorization supported in existing database systems are all designed for relational, hierarchical, or network models of data.

The basic idea of access control model is to group subjects into access control groups and to grant authorizations in terms of access types, such as read, write, and delete [17]. These access types are usually ordered such that the authorization for one right may include others. Thus authorization for a delete may imply authorization for a write, which in turn may imply authorization for a read. Database systems usually define authorizations for the schema entities,

such as classes, attributes, and indexes [17]. In the database object part of the authorization, Rabitti et al., discusses two graphs: the authorization object schema (AOS) and the authorization object graph (AOG). The following Figures 2 and 3 show the examples of AOS and AOG. The edges in the AOG represent relationships between objects. The nodes in both the AOS and AOG deal with collections of objects of a given type depending on how OODB handles sets of objects. AOS looks at the possible granules defined by the schema for OODB; AOG considers actual object instances on the database. All access control problems eventually seek an answer to a fundamental question typically posed as follows: subjects allowed to access of type on object *o*. The answer to any access control request can now be obtained by utilizing a function *f* that determines if the corresponding authorization *(s, o, a)* is true or false.

XML schema defines XML documents with a hierarchical structure, containing attributes and elements. Elements can have sub-elements nested to any levels. In this paper we assume that the objects using access control consist of documents which conform to an XML Schema. For Rabitti model [16] the XML Schema can be used to construct the AOS, the XML documents make up the AOG. An AOS and AOG, for our example in Section 2, are shown in Figures 2 and 3, respectively. In the AOS in Figure 2, we have indicated elements by rectangles, attributes by ovals. So *Staff, GeneralInfo, WorkInfo,* and *FinancialInfo* are elements, *StaffId* is attribute of element *Staff*. In the AOG in Figure 3, the element values are shown as their string value and attribute values are in quotes. *Tony, Tony@hotmail.com* and *3800* are string value of elements, "*12345*" is the value of attribute.

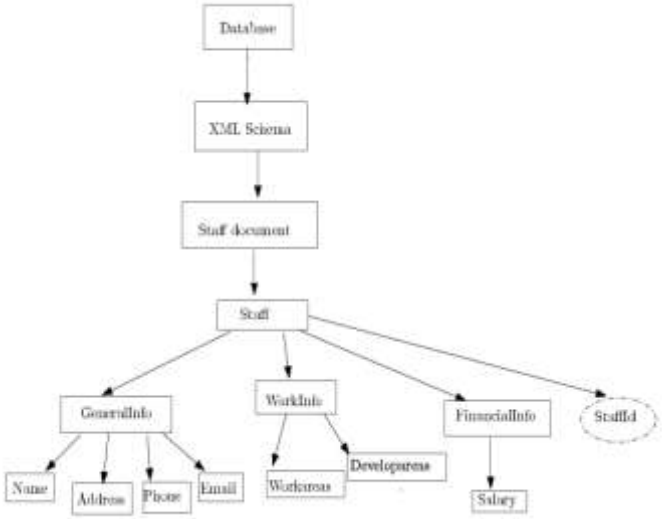

Figure 2: Authorization Object Schema

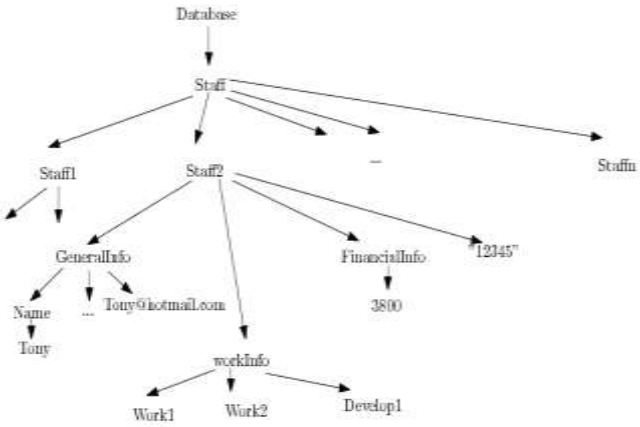

Figure 3: Authorization Object Graph

4. Usage control models with XML databases (UCMXMLDB)

For the XML databases with the usage control model, subjects and objects are similar concepts as in the OODB authorization models. A right represents the access of a subject to an object, such as read or write. Several possible operations on parts or all of an XML document: reading, changing the contents (update), adding sub-elements or attributes (extend) and possibly other operations like following a pointer, applying XLS transformations, etc. This is also

similar as component authorization types in discretionary access control model for object-oriented database. In a real environment, the resource of XML objects based on different XML Schemas from various servers and organizations. Therefore there will be vast number of schema components. On the other hand, a number of users will make subjects complex. With all these features, the security administration will be very complex in both centralized and decentralized deployments.

We now discuss authorization models for XML database adopting usage control in this section. XML documents provide the information structure and semantics in a web environment and XML documents will be stored in a relational, object-oriented database or XML Native database. The authorization models for XML Native database model are the same as the OODB model. Based on the three usage control components *Authorization*, *Obligations* and *Conditions*, we develop six possible cases as core models: pre-Authorization, ongoing-Authorization, pre-Obligations, ongoing-Obligations, pre-Conditions, ongoing-Conditions using XML database. Meanwhile, to apply the modularity, extensibility information stored in the XML databases, all the components in our models, such as subjects, subject attributes, objects, object attributes, rights, authorizations, obligations, and conditions are specified in an XML format.

1. pre-Authorization Model (UCMpreA) with XML databases

In *UCMpreA* model, the decision process is performed before the access is allowed. The *UCMpreA* model provides an authorization method on whether a subject can access the XML database. It consists of the following components: *S, XO, R, ATT(S), ATT(XO)* and usage decision functions *preA*, respectively. Where *S, XO, R,* represent Subject, XML object, Rights required on XML object access modes, respectively. *ATT(S), ATT(XO),* represent attributes of subjects, XML object, respectively. In usage control, authorization decision is made based on subject attributes and object attributes. During the *UCMpreA* model it includes following process:

$allowed\ (s, xo, r) \Rightarrow preA(ATT(s), ATT(xo), r),$

This predicate indicates that if subject *s* is allowed to access XML object *xo* with right *r,* then the indicated condition *preA* must be true. The *allowed(s, xo, r)* predicate shows that subject *s* can access the XML database object. At this process, *UCMpreA* corresponds roughly to the discretionary access control model. *preA* is very similar as the triple relationship *f* in OODB authorization model in previous section. The three components *ATT(s), ATT(xo)* and *r* also can replace *S, O, A* in OODB authorization model. For example, applying *UCMpreA* to Figures 2 and 3, for the element staff *Tony*, his financial records can only be accessed by himself and financial administrators. Before they want to access this information, they have to give a username (subject) and a password (subject attribute). In the meantime they need to know *Tony*'s (object) staffId (object attribute). Then they can to read (right) Tony's financial records.

2. ongoing-Authorizations Model (UCMonA) with XML databases

A usage control model for ongoing-Authorizations is used to check ongoing authorizations during access processes. In this model, usage requests are allowed without any 'pre' decision making. With XML database, *UCMonA* model has the following components: *S, XO, XD, R, ATT(S), ATT(XO)* as before in UCMpreA and ongoing usage decision functions *onA. onA* is used to check whether *S* can continue to access or not. It includes two processes:

$allowed\ (s, xo, r) \Rightarrow true,$

$stopped\ (s, xo, r) \Leftarrow \neg\ onA(ATT(s), ATT(xo), r),$

In this model usage decision function is *onA.* The *allowed (s, xo, r)* is a prerequisite for ongoing authorization on XML object *xo*. Compare with OODB authorization model, *onA* is similar as the

triple relationship *f*. *ATT(s), ATT(xo), r* components also replace *S, O, A* in OODB authorization model. The access of subject *s* to *xo* is terminated if the ongoing authorization *onA* fails. During this process the requested access is always allowed as there is no pre-authorization all the time. *allowed (s, xo, r)* is required to be *true*, otherwise ongoing authorization should not be initiated. Ongoing authorizations are active throughout the usage of the requested right, and some requirements are repeatedly checked for a continued access. These checks are performed periodically based on time or event. During the process *stopped* procedures are performed when attributes are changed and requirements are no longer satisfied. *Stopped (s, xo, r)* indicates that rights *r* of subject *s* on object XML database are revoked and the ongoing access terminated. For example, in a limited number of simultaneous usages, suppose there are only two financial staffs and Tony himself able to access the information about his salary. If a third financial staff requests access and pass the pre-authorization, the staff with the earliest time access would be terminated. *onA* monitors the number of current usages on *xo(ATT(xo))*, records which access is the earliest start, and terminates it. While this is a case of ongoing authorization, it is important that the certificate should be evaluated in a *pre* decision.

3. pre-Obligations Model (UCMpreB) with XML databases

UCMpreB introduces pre-obligations that have to be fulfilled before access is permitted. It will return *true* or *false* for usage decision depending on whether obligation actions have been fulfilled or not. For example, an administrator is required to register by filling forms before accessing one staff financial information. Also pre-obligation action may be done by some other subject. When using *UCMpreB* model to access XML database documents, the *UCMpreB* model has the following components: *S, XO, R, ATT(S), ATT (XO)* as before with UCMpreA, *OBS, OBO* and *OB* represent obligation subjects, objects, and actions, respectively; decision function *preObfilled*: $OBS \times OBO \times OB \rightarrow \{true, false\}$. The function *preObfilled* is used to check if obligations are obeyed or not before the subject(s) accesses the object(xo). *preObfilled* : $OBS \times OBO \times OB \rightarrow \{true, false\}$ is the same as the triple relationship $f : S \times O \times A \rightarrow (true, false)$ in OODB authorization model. The preObfilled function must be true if subject(*s*) is allowed to access XML object *xo* with right *r*.

4. ongoing-Obligations Model (UCMonB) with XML databases

Different from the pre-Obligations model, Ongoing-obligations model may have to be fulfilled periodically or continuously. For example, when an administrator accesses the financial information through the Internet within every 15 days, she/he may have to repeatly input a password. The model concerns whether obligations have to be fulfilled. Using the *UCMonB* model with XML databases it has following components: *S, XO, R, ATT(S)* and *ATT(XO)* as before and an ongoing decision function *onObfilled* : $OBS \times OBO \times OB \rightarrow \{true, false\}$. *OBS, OBO,* and *OB* represent obligation subjects, objects, and actions, respectively; The ongoing function *onObfilled* is used to check if obligations are continually obeyed or not during subject(s) access object (xo) in XML database document. *onObfilled :* $OBS \times OBO \times OB \rightarrow \{true, false\}$ is the same as the triple relationship $f : S \times O \times A \rightarrow (true, false)$ in. *allowed*(s, xo, r) is a prerequisite for *UCMonB*. It means that *s* can access XML database documents. Where *stopped (s, xo, r)* indicates that the access of *s* on *xo* with *r* is revoked if the ongoing obligations fail.

5. pre-Conditions Model (UCMpreC) with XML databases

Condition is a very important component in the usage control model. Conditions define that certain restrictions have to be satisfied for usage. By using conditions in usage decision process, it can provide finer-grained controls on usage. Usually the pre-

conditions model has to be used before requested rights are used. For example, suppose there are some requirements to restrict times for accessing information. You should check them before a usage allowed. The *UCMpreC* model with XML databases has following components: *S, XO, R, ATT(S),* and *ATT(XO)* as before and *preCON* (a set of pre-conditions) is for verifying conditions, $preCON \rightarrow \{true, false\}$. The function *preConSatisfied*: $S \times O \times R \rightarrow 2^{preCON}$ is used to check whether the pre-conditions are satisfied or not. It has following process during the access:

$allowed(s, xo, r) \Rightarrow preC(s, xo, r)$

preC is very similar as the triple relationship *f* in OODB authorization model. *s, xo, r* also replace *S, O, A* in OODB authorization model. *allowed*(s, xo, r) expresses that all conditions have to be satisfied before access is approved. Unlike other models, condition models cannot have update procedures. All pre-conditions have to be checked if there are more than two conditions.

6. ongoing-Conditions Model (*UCMonC*) with XML databases

UCMonC model requires conditions to be satisfied while rights are in active use. If violating any of the restrictions, the allowed right is revoked and the exercised is stopped. For example, if the staff information system status changes to "special mode", the access by some users may be terminated.

For the usage access control with XML databases above the six models all include some functions. These functions, such as *preA, onA, preObfilled, onObfilled, preC, etc* are very similar as *f*: $(S \times O \times A)$ *in* the OODB authorization model. But usage authorization method for XML database focuses on checking users' (subjects') authorizations, obligations and conditions with continuity properties. It also can be used for different processes. In practice, the six models of pre-Authorizations, ongoing-Authorizations, pre-Obligations, ongoing-Obligations, pre-Conditions and ongoing-Conditions may need to be combined for an access control.

5. Discussion

Jingzhu and Sylvia[20] introduced a role based approach to access control for XML database. In their model, they provide a general access control methodology for parts of XML documents, combining role based access control as found in the Role Graph Model, with a methodology originally designed for object-oriented databases. Several constraints are included in the model. Their protocol is based on RBAC and hence it focuses on permissions-role assignment, objects hierarchies and constrains. Our approach is based on usage access control; we have analysed the characteristics of various access authorizations and presented detailed models for different kinds of authorizations. It is an important state for XML documents in the Internet since users always alter their conditions or obligations. By contrast, users in our scheme have to pass pre-Authorizations, ongoing-Authorizations, pre-Obligations, pre-Conditions and ongoing-Obligations as well as ongoing-Conditions. This indicates that our method is much more secure and powerful in dynamic environments.

Alban, et al [9] presented an access control model for regulating access to XML documents. In their papers, they use the XPath language to address XML fragments and the XSLT language to compute the view. The model offers the possibility of defining content-based authorization rule. By contrast, our work provides a rich variety of options that can deal with XML database documents. Users can access XML documents with their keys at any time, even when their properties are updated. In our scheme, users have to satisfy pre-Authorizations, pre-Obligations, pre-Conditions ongoing-Authorizations, ongoing-Obligations and ongoing-Conditions.

6. Conclusions

In this paper we introduce XML, XML databases, usage control and OODB authorization model. We discuss access models for XML databases by using usage control. OODB authorization control is used to database system access control. It only has authorization component, it doesn't include obligation and condition components. Usage control encompasses traditional access control, trust management and beyond them in its scope. Usage control model provide an approach for the next generation of access control. It covers both security and privacy issues of current business and information systems. Comparing with the OODB authorization model usage control model for accessing XML databases has richer uses. We developed six possible cases as core models for XML databases using usage access control. In these models we analyse not only decision factors, such as authorizations, obligations and conditions, but also the continuity properties. In this paper we have provided a foundation for further research and development on usage control model with XML databases. It is a new application with usage control. However, much work is still to be done before these models can be used in practice.

References:

[1] Bertion E. Protecting XML documents. In *Proceedings of Computer Software and Applications Conference*, Page: 132-133, 2000, Taiwan.

[2] Bertion E. and Ferrari E. Secure and selective dissemination of xml documents. *ACM trans. Inf. Syst. Secur.*, 5(3):290-331, 2002.

[3] Bertion E., Castano S., Ferrari E. and Mesiti E. Controlled access and dissemination of xml documents. In *Processings of the second international workshop on Web information and data management*, pages 22-27. ACM Press, 1999.

[4] Bourret R. XML Database Products, 7 August, 2003.

[5] Bray T., Paoli J., Sperberg M and Maler E. *Extensible Markup Language (XML) 1.1 (Second Edition)*. World Wide Web Consortium (W3C), Cambridge, MA, USA, 2000.

[6] Damiani E., Capitani S. and Samarati P. Towards securing xml web services. In *Proc .of the 2002 ACM Workshop on XML Security,* Washington, DC, USA, November 2002.

[7] Damiani E., Paraboschi S. and Samarati P. A fine-grained access control system for aml documents. *ACM Trans. Inf. Syst. Secur.*, 5(2):169-202, 2002.

[8] Damiani E., Samarati S., Vimercati di. and Paraboschi S. Controlling access to XML documents. *IEEE Internet Computing*, 5(6):18-28, 2001.

[9] Gabillon A. An authorization model for xml databases. *In Proceedings of the 11th ACM conference on Computer Security*, 2004.

[10] Kakeshita T. and Murata M. A declarative manipulation for OODB and XML DB having cyclic schema, *IEEE,* 2003.

[11] Kudo M. and Hada S. Xml access control, http://www.tri.ibm.com/projects/xml/xacl/xmlac-proposal.html

[12] Kudo M. and Hada S. XML document security based on provisional authorization. *In Proc. 7tth ACM Conference on Computer and Communications Security*, pages87-96, 2000.

[13] Kuper G., Massacci F. and Rassadko N. Generalized xml security views. In *Proceedings of the 10th ACM sysposium on Access control models and technologies*, pages 77-84. ACM Press, 2005.

[14] Park J. and Sandhu R. Towards usage control models: beyond traditional access control. In *Proceedings of the seventh ACM symposium on Access control models and technologies*, page 57-64. ACM Press, 2002.

[15] Park J., Sandhu R., and Schifalacqua J. Security architectures for controlled digital information dissemination. *In Proceedings of 16th Annual Computer Security Application Conference,* December 2003.

[16] Rabitti F., Bertino E., Kim W. and Woelk D. A model of authorization for next-generation database

systems. *ACM Trans Database Syst*, 16(1):88-131, 1991.

[17] Roshan K.T. and Ravi S.S. Discretionary access control in Object-Oriented databases issues and research directions. *Proc. Of the 16th NIST_NCSC National Computer Security conference*, pages: 63-74, 1993.

[18] Sandhu R. and Park J. Usage control: A vision for next generation access control. In *MMM-ACNS 2003*, pages 17-31, Springer-Verlag Berlin Heideberg, 2003.

[19] Sun L. and Li Y. DTD level authorization in xml documents with usage control. In *International Journal of Science Network Security*, volumn 6, pages 244-250, November 2006.

[20] Wang J. and Osborn S. L. A role-based approach to access control for XML datbases. *SACMAT'04*, June 2-4, 2004.

[21] Wang Y. and Tan K. A scalable xml access control system. In *Proceedings of the 10th international WWW conference*. Poster, 2001.

[22] Zhang X., Park J. and Parisi-Presicce F. A logical specification for usage control. In *SACMAT'4*. ACM Press, 2004.

[23] Zhang X., Park J. and Sandhu R. Schema based xml security: Rbac approach. In *Proceedings of the IFIP WG*. ACM Press, 2003.