
Que Nguyet Tran Thi
Faculty of Computer Science & Technology
HCMC University of Technology, VNUHCM
Ho Chi Minh City, Vietnam
ttnquyet@cse.hcmut.edu.vn

Tran Khanh Dang
Faculty of Computer Science & Technology
HCMC University of Technology, VNUHCM
Ho Chi Minh City, Vietnam
khanh@cse.hcmut.edu.vn

Abstract— The rapid growth of location-based applications, geographic or large scale information systems has resulted in the demand of strictly controlling data access that requires specifying and enforcing fine grained policies with the variety of context-aware spatial and temporal restrictions. Besides, the interoperable use of distributed and heterogeneous data such as data sharing, data integration or data exchanging between different organizations has caused the formation and development of access control mechanisms using XML for enforcing security rules and policies in accordance with the international standards. In this paper, we propose an extension of XACML called the X- STROWL model for a generalized context-aware role-based access control (RBAC) model with the support of spatio-temporal restrictions and in conformity with the NIST standard for RBAC. In doing this, the XACML architecture is augmented with new functions and data types. In addition, policies are reorganized to adopt with the NIST standard. Besides, a set of conditions aimed to a certain circumstance can be generalized into a context profile and specified in the access control policies. The model also integrates the OWL ontology for semantic reasoning on hierarchical roles to simplify the specification of access control policies and increase the intelligence of the authorization decision engine.

Keywords- XACML, access control model, RBAC, RBAC with OWL, spatial temporal data access control model, context aware access control model

I. INTRODUCTION

Extensible Access Control Markup Language (XACML) [1][2] is an international industrial standard by OASIS community designed to express diversified security policies, access rights as well as authorization decision requests/responses in XML. XACML provides a general and flexible model describing the way to evaluate an access request. Besides, XACML supports many specialized features such as the extensibility of new data types, attributes and user-defined functions in the policy specification, the ability of combining multiple rules and policies into a decision based on combining algorithms, and especially the allowance of including contextual conditions into the policies. Moreover, OASIS has also defined a RBAC profile of XACML [3] that can build the role-based access control model under this language. However, the profile still obtains limitations and has not complied with the NIST standard for RBAC yet [4][5]. Namely, it has not effectively supported hierarchical roles as well as static and dynamic separation of duties constraints (S/DSOD constraints) in user-role and permission-role assignments. Additionally, in order to reduce the cost of administration and the storage space of policies, access control systems should require the capability of reasoning on the role hierarchy to inherit positive permissions from junior roles or negative permissions from senior roles.

In this paper, we propose a generalized access control model, X- STROWL (eXtended XACML for Spatial Temporal Role based access control model with OWL), which is extended from XACML but supports the RBAC model according to the NIST standard and addresses the spatio-temporal access restrictions. Besides, a new kind of policy – the context policy – which defines a set of conditions generalized into a context profile and shared between access control policies is also indicated in this extension. In addition, semantic reasoning on hierarchical roles and resources are also taken into account in the X- STROWL model. To such an extent that one of our major contributions is suggesting an innovative generalized framework which satisfies the diversified access control requirements in the real world.

The rest of paper is organized as follows. Section II analyzes the limitations of the OASIS RBAC profile of XACML which is the foundation for extensions and presents the approaches of relevant access control models. The details of the X- STROWL model are illustrated in section III. We give the comparison between the X- STROWL model and other related models to conclude the outstanding features of the proposed access control model in section IV. Finally, the conclusion and further work are summarized and discussed in section V.

II. RELATED WORK

Before elaborating on the X- STROWL model, we introduce briefly the most relevant models extended from XACML [1][2]. In these extensions, the RBAC profile of XACML approved by OASIS committee [3] is considered as the main reference of approaches for modeling role-based access control by XACML. For this reason, we present its limitations in details first and then discuss the other models.
A. Limitations of OASIS RBAC profile of XACML

XACML 3.0 has released the profile for core and hierarchical role-based access control model to express the RBAC model under XACML. In the scope of the profile, it can express policies with multiple users per role, multiple roles per user, multiple permissions per role, and multiple roles per permission as well as support the hierarchy of roles. However, it still has several shortcomings as follows:

- Role Enablement Authority (REA) that takes the responsibility for role assignments to subjects is not described in the XACML workflow. It is assumed that the role attributes are assigned beforehand to users and available in the request context produced by Context Handler before sending it to PDP [3].

- To support the core RBAC model [4], each Role <PolicySet> policy that represents a role in the system contains only a single <PolicySetIdReference> tag element referencing to a Permission <PolicySet> policy including the set of permissions for that role. With this approach, the individual permissions cannot be shared between roles and there is also no feature supporting many-to-many role-permission assignment. Besides, it is infeasible to define constraints explicitly on role-permission assignments. A user after being assigned to a role by Role Enablement Authority, s/he will have all permissions stated in the given Permission <PolicySet> policy associated with her/his role including inherited permissions if any. In fact, the assignment should require further conditions, e.g., the permission “operate on patients” is granted to a doctor role only if s/he is standing in the operating room. Moreover, without supporting conditions specified in the policies, it cannot be applied for partial delegations that delegate only some permissions of a certain role to others along with concrete conditions such as “in the summer vacation”, “in the foreign country”, etc.

- The implementation of role hierarchy is in terms of the hierarchical relation between permissions by including a <PolicySetIdReference> tag element in the Permission <PolicySet> policy of senior role to the Permission <PolicySet> policies of junior roles instead of demonstrating the relation between roles explicitly. Thus, it lacks the practicability and visualization in building hierarchical roles in the real systems. Moreover, in some cases, a senior role can inherit permissions of junior roles but not always. Therefore, the access control policies should specify further conditions in the permission assignments to roles.

- The static and dynamic separation of duties constraints on both of user–role assignments and role–permission assignments [4][5] are not taken into consideration in the profile. Consequently, it cannot achieve the third level of the NIST Standard for RBAC model.

It can be seen that the RBAC profile of XACML released by OASIS cannot fully support the features for a RBAC model according to the NIST standard. Some relevant access control models extended from this profile provide more advantages but still retain some disadvantages. The following section discusses about these models in details.

B. Access control models extended from XACML

The GeoXACML model proposed by [6][7] provides the expansion of spatial data types and functions to indicate the spatial data driven restrictions. It can also be extended to support other spatial access restrictions based on user and contextual information. However, the role based access control model is not taken into consideration. In GeoXACML, an authorization is defined as a triple of subject, resource, and action. Therefore, GeoXACML does not support the advantages of RBAC model that is usually deployed in the large systems [4][5].

Following the RBAC profile of XACML defined by OASIS, there are other models which possess some both of advantages and disadvantages.

The model in [8] implements the Or-BAC model [9] in XACML that provides the abstract entities including view as a set of objects with the same properties, activity as a set of actions with the same operation, and context as a set of conditions for a certain circumstance. Nevertheless, conditions about contexts are evaluated independently with role, view, and activity attributes. Moreover, unlike the RBAC model of NIST standard which represents an authorization state as the relation between roles and permissions, an access control policy in this model is a triple combining role, view, and activity. As a result, the constraints on permission assignments to roles are not be addressed. S/DSOD constraints are not also taken into account in the model as well.

One another model proposed by [10] describes the routine of Role Enablement Authority that are out of the scope of the OASIS RBAC profile of XACML. Moreover, the major contribution of authors is that the S/DSOD constraints are modeled by XACML policies. Therefore, it is flexible to define additional conditions in the policies such as role r1 and role r2 are mutually exclusive only in the morning. This idea is also brought into the X-STROWL model. In another aspect, the model does not distinguish between two concepts of enabled and active roles. A role which is enabled by Role Assignment <PolicySet> policies through REA is considered as an active role for the subject and then its state is updated before PDP evaluates access control policies. Therefore, it will impact on the S/DSOD constraint evaluation for later processes if no permission associated with the role is authorized.

For reasoning on hierarchical roles, there are two relevant models [11][12] for the representation of role hierarchy and S/DSOD constraints using XACML and OWL Ontology [13][14] that provides semantic concepts and reasoning engine.

In [11], the authors have proposed ideas using semantic functions in the <target> matching of access control policies to maintain all possible roles in the role hierarchy if the corresponding permissions given in the policies want to be propagated to junior or senior roles. Besides, S/DSOD constraints and role assignments are specified by axioms of OWL ontology for OWL reasoner. However, access control policies are specified on the relation between role, action, and
resource. Therefore, the authorization is not based on the relation between roles and permissions as described in the NIST standard for RBAC model. As a result, it does not support many to many permission-role assignment [4][5] and specify constraints on these assignments. Moreover, the specification of S/DSOD constraints using OWL is more inconvenient than using XACML policies in the cases they require additional contextual conditions.

Another model [12] integrates OWL into XACML to support semantic concepts on both of hierarchical roles and resources. Nevertheless, the model does not use Role Assignment <PolicySet> policies to assign roles to users but creates axioms in OWL knowledge base. Therefore, it cannot take advantage of the feature of XACML policies that allows specifying further conditions.

Through the analysis of above models, the X-STROWL model inherits several of their strong points and regulates the data flow of XACML for enforcing access control policies in conformity with the NIST standard for RBAC model and supporting spatial and temporal access restrictions.

III. EXTENDED XACML PROFILE FOR X-STROWL MODEL

As mentioned above, the foundation of the proposed model is derived from XACML and based on the NIST standard for RBAC model. We provide the extension of the OASIS RBAC profile of XACML and overcome the disadvantages of models discussed in section II by introducing the new kinds of XACML policies and proposing a comprehensive framework for enforcing these access control policies.

A. Types of Policies

In general, there are five kinds of policies used in the X-STROWL model. They are evaluated by the specialized components (see section III.B for details). According to the RBAC model, the logical relation between them is demonstrated in Fig. 1.

1) Role Assignment <PolicySet>: defines which role can be enabled to which subjects under which conditions. This kind of policy is evaluated by the RoleEA component to retrieve role attributes for subjects.

2) Permission <PolicySet>: describes a valid combination between resources and actions along with any further conditions for permissions in the organization. The conditions are applied for all roles associated with these permissions. Such policies are used in the PermissionFinder component to obtain the authorized permissions corresponding to the requested action and resource.

3) Permission Assignment <PolicySet>: specifies which permissions are assigned to which roles and under which conditions. Besides, the policy can allow the propagation of positive permissions to senior roles or negative permissions to junior roles through semantic functions with the same approach of [11] that communicates with OWL Ontology Reasoner to obtain the hierarchy of roles. By this way for specifying the hierarchy of resources, it also allows objects in the low level to be valid for a permission if existing objects in the higher level associated with that permission. Such policies are evaluated by the main PDP component to determine whether a data access is permitted or not.

4) Separation of Duty <PolicySet>: is used to model S/DSOD constraints on the assignments of role-user and role-permission as suggested in the NIST RBAC model. Conditions are also established for the constraints through the <Condition> element. Additionally, it can include semantic functions to state that the constraints can impact on the senior or junior roles according to the predefined hierarchy. They are assessed in the final stage by the S/DSODEvaluation component to verify that the activation of enabled roles for subjects is not violated by the policies of DSOD constraints. Meanwhile, the SSOD constraints are normally specified in role assignments by administrators through the administrative policies. This process is the same as the evaluation engine of policies for data access but its details are not mentioned in this paper.

5) Context <PolicySet>: contains a set of conditions for defining context profiles. For example, a general condition for all permissions of users in the organization is that they are only permitted to request data in the working hour and at the office. In such case, a context, e.g., OfficeWorkingHours is declared in a Context <PolicySet> policy and established as a condition in the policies. Such context attribute values are maintained by the ContextEA component through Context <PolicySet> policies.

In the X-STROWL model, all kinds of policies are enforced for the whole process of data access control. To such an extent, the mechanism of evaluating these policies possesses some common characteristics. The evaluation components in the model are implemented with the similar architecture extended from the XACML framework. Consequently, the model can provide the generalization and consistence for all components in design and implementation.

B. Workflow of X-STROWL model

As illustrated in Fig. 2, except for the basic components of the OASIS XACML framework such as PEP, PDP, PIP, PAP and Context Handler, the X-STROWL model includes four additional main blocks, RoleEA (Role Enablement Authority), ContextEA (Context Enablement Authority), PermissionFinder and S/DSOD Evaluation. Note that in this figure, it starts with the step 2 instead of step 1 which indicates the process of writing and retrieving policies from PAP as described in the OASIS XACML specification.
Each block in the X-STROWL model consists of two private components, ContextHandler and PDP, with the same architecture of RoleEA as shown in Fig. 3. RoleEA takes the responsibility to maintain the list of enabled roles for subjects. The mechanism for RoleEA works similarly to XACML. RoleEA-ContextHandler will receive the request from the main ContextHandler to find authorized roles. For each role stored in the database, RoleEA-ContextHandler issues a request to RoleEA-PDP to examine that whether the role can be assigned to the subject through Role Assignment <PolicySet> policies. Whenever RoleEA requires additional information to evaluate conditions, it will work with PIP to retrieve values from the appropriate sources such as the resource content, database, external devices e.g. Global Position System (GPS) tools, etc. In the case that the conditions involve a context attribute value, RoleEA will interact with ContextEA to validate the value of context stated in the Role Assignment <PolicySet> policies. Other parties if they desire to obtain context attribute values will communicate with ContextEA as well. The architecture of ContextEA is also similar to the original XACML and RoleEA. ContextEA-ContextHandler receives requests from external blocks to validate a context status and then makes a request to ContextPDP to evaluate it based on Context <PolicySet> policies according to the data flow of XACML. Meanwhile, the PermissionFinder component discovers permissions which can match with the action in the data access request based on the Permission <PolicySet> policies. The other component, S/DSOD Evaluation, performs the assessment of static and dynamic separation of duties constraints for detecting mutual exclusive roles and permissions or limitations about the number of roles and permissions being assigned to users or activated at the same time [4][5].

The whole architecture operates by the steps shown in Fig. 2 and summarized as follows:

1) Firstly, PEP sends a data access request to ContextHandler in its native request format. ContextHandler then constructs another requests to RoleEA and PermissionFinder to maintain the list of enabled roles and permissions corresponding to the request of subject. These processes can be performed simultaneously.

2) After receiving the lists of available roles and permissions from RoleEA and PermissionFinder, ContextHandler creates requests for each pair of possible roles and permissions and sends to PDP for evaluating through Permission Assignment <PolicySet> policies. The steps from 6 to 20 are demonstrated for the evaluation process of each candidate request (for example, permission p1 is assigned to role r1). If PDP requires more information to verify conditions, it will inquire of ContextHandler and then ContextHandler queries about the attributes from PIP as described in the OASIS XACML specification. With regard to semantic functions on the hierarchy of roles defined in the policies, PDP will communicate with OWL Ontology Reasoner to obtain the possible roles. Permission <PolicySet> policies in the step 1 can also include semantic functions to state that which permissions are applied to the supper and sub sets of resources. The reasoning is based on the hierarchical relation between roles and resources represented by OWL ontology axioms [11][12][13][14]. Besides, the conditions specified in Permission Assignment <PolicySet> policies can include constraints on context attributes. In this case, it must perform additional steps to validate the context status for the request evaluation. PDP sends ContextHandler a request to confirm values for context attributes stated as a condition for enforcing access control policies. Subsequently, ContextHandler interacts with ContextEA to fulfill this work.

---

![Figure 2. Workflow of X-STROWL model](image-url)
3) After evaluation, in step 18, if PDP produces a permit response, it means that the subject can perform the request with the role specified in the corresponding Permission Assignment <PolicySet> policy evaluated by PDP from the step 6 to 18 [Fig. 2]. However, the dynamic separation of duties constraints can prevent enabled roles from being activated for subjects. In the X-STROWL model, such constraints are verified by the S/DSOD Evaluation component through Separation of Duty <PolicySet> policies. The mechanism for this function is the same as RoleEA and PermissionFinder. However, S/DSODEvaluation-PDP can interact with OWL Ontology Reasoner through semantic functions in the case that the S/DSOD constraints require to be applied on the hierarchy of roles. For example, a Separation of Duty <PolicySet> policy implies that role r1 along with all senior roles of r1 and role r2 are mutually exclusive. When evaluation, PDP executes the semantic functions for retrieving the senior roles of r1 through the hierarchy of roles predefined by OWL ontology axioms as shown in the steps 12 and 13 of the main PDP described above.

4) There are two options considered in this model after verifying S/DSOD constraints. The first one is that all candidate pairs of roles and permissions will be evaluated by PDP with the same procedure from step 6 to step 20 [Fig. 2] and then ContextHandler component holds the valid pairs in a list. Subsequently, in the step 21, ContextHandler will return a permit response to PEP together with the permitted roles and special obligations that allow the requester to choose one role in the list returned and then update the chosen role’s state for evaluating S/DSOD constraints of other processes afterward. In the case that no enabled role cannot be activated due to the violation of S/DSOD constraints, ContextHandler will return a deny response to PEP accompanied with the list of enabled roles and obligations that suggest the subject to disable a certain role being active but having conflicts with the enabled roles stated in the S/DSOD constraint policies. After that, if the user disables a suggested role and resends the request of data access, the whole process will be reevaluated. In the second option, if the role and permission being checked does not violate the policies of S/DSOD constraints, ContextHandler will return a permit response to PEP with the obligation which updates the state of the role into “active” and accomplishes the process of evaluating access control policies. Otherwise, ContextHandler continues to examine other candidates with the same process (step 6 to 20) for acquiring the first authorized role and permission.

In summary, the X-STROWL model is an extension of XACML and works in accordance with the mechanism of XACML. The extra components including RoleEA, PermissionFinder, ContextEA, S/DSOD Evaluation and the main process have the same operating method. Moreover, due to the specification of S/DSOD constraints by XACML policies, contextual restrictions are also supported in these constraints. In addition, the evaluation of S/DSOD constraints is performed after obtaining the legal roles and permissions matching with the request of subjects. As a result, the X-STROWL model can support S/DSOD constraints for both of user-role assignments and role-permission assignments. Therefore, the X-STROWL model can achieve the third level – Constrained RBAC of NIST Standard RBAC model [4][5].

C. Spatial and temporal access restrictions

Access restrictions enforced by the X-STROWL model can include spatio-temporal conditions. For example, an administrator can define a policy that users are permitted to read the information about the boundaries of citizen houses within 5 years only if the requested objects are located in the area of a certain city and valid during that period. Thank to the extensibility of XACML, it is feasible to declare new data types such as several temporal and spatial data types listed in Table I, by defining new tags within the <AttributeValue> element as described in GeoXACML for spatial restrictions [6][7]. Additionally, spatial and temporal functions [15], for instance listed in Table II, are accompanied with spatial and temporal data types to establish conditions stated in the <Condition> element of XACML policies.

<table>
<thead>
<tr>
<th>TABLE I. SPATIAL AND TEMPORAL DATA TYPES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial data types</td>
</tr>
<tr>
<td>Point</td>
</tr>
<tr>
<td>LineString</td>
</tr>
<tr>
<td>LinearRing</td>
</tr>
<tr>
<td>Polygon</td>
</tr>
<tr>
<td>Box</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II. SPATIAL AND TEMPORAL FUNCTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial functions</td>
</tr>
<tr>
<td>Disjoint</td>
</tr>
<tr>
<td>Touches</td>
</tr>
<tr>
<td>Crosses</td>
</tr>
<tr>
<td>Within</td>
</tr>
</tbody>
</table>

With this approach, administrators can define complicated policies including spatial and temporal conditions for large
scale systems requiring a fine grained access control model. Moreover, \textit{X-STROWL} model can be also applied in Geographic Information Systems with the same mechanism of GeoXACML \cite{6}\cite{7}. However, the access restrictions in \textit{X-STROWL} model can be designated based on spatio-temporal relations between attributes of roles, subjects, actions and resources and contextual information such as request time, locations of subjects, the proximity of subjects, e.g., an ongoing fire, a running process, the number of users around the subject, etc. Such access restrictions are called \textit{context driven}. In summary, by extending from XACML with the new data types and functions, providing semantic concepts in retrieving summary, by extending from XACML with the new data types and functions in \cite{11}, we give a new kind of policy restrictions by expanding new data types and functions. In this paper, we give a new kind of policy restrictions by expanding new data types and functions. In this respect, we have proposed a generalized access control model, called \textit{X-STROWL}, which integrates XACML and OWL ontology for supporting the NIST standard RBAC model with the variety of spatial and temporal access restrictions by expanding new data types and functions. In which, we give a new kind of policy \textit{Permission Assignment <PolicySet>} for specifying which roles to hold which permissions along with further specific conditions. The context notion is also taken into account in all kinds of policies as a condition for user – role assignments, permission – role assignments, and S/DSOD constraints. The data flow for evaluating a request based on the phases of RBAC model is also described in details in this paper. However, the aspects related to the structure of policies, the mechanism of OWL Ontology Reasoner, as well as concrete examples are not presented in details in this paper. Moreover, how to integrate the \textit{X-STROWL} model into the real systems such as Geographic Information Systems and the spatial and temporal database systems is our further consideration.

IV. EVALUATION

To evaluate the features of the \textit{X-STROWL} model compared to others, we offer the criterions as listed in Table III including the supports of the NIST standard for RBAC model, XACML standard, the kinds of access restrictions, and the reasoning service for hierarchical roles and resources. Note that in this table, the context driven access restrictions not only includes contextual information but also provides the generalization of conditions for particular circumstances.

In table III, we also give the evaluation on other models presented in section II and remark the proposed model. By restructuring policies, remodeling the workflow, and inheriting several advantages of other models such as specifying S/DSOD constraints by XACML policies in \cite{10} and using semantic functions in \cite{11}, \textit{X-STROWL} can surmount the disadvantages of relevant models and provide the comprehensive features except for RBAC3 - Symmetric RBAC model according to the NIST standard \cite{4}\cite{5}. However, with the proposed architecture, it is not too difficult to implement the interface for managing the permission assignments to roles and vice versa as suggested in the Symmetric RBAC model.

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a generalized access control model, called \textit{X-STROWL}, which integrates XACML and OWL ontology for supporting the NIST standard RBAC model with the variety of spatial and temporal access restrictions by expanding new data types and functions. In which, we give a new kind of policy \textit{Permission Assignment <PolicySet>} for specifying which roles to hold which permissions along with further specific conditions. The context notion is also taken into account in all kinds of policies as a condition for user – role assignments, permission – role assignments, and S/DSOD constraints. The data flow for evaluating a request based on the phases of RBAC model is also described in details in this paper. However, the aspects related to the structure of policies, the mechanism of OWL Ontology Reasoner, as well as concrete examples are not presented in details in this paper. Moreover, how to integrate the \textit{X-STROWL} model into the real systems such as Geographic Information Systems and the spatial and temporal database systems is our further consideration.

### Table III. The Checklist of the Supporting Features

<table>
<thead>
<tr>
<th>Features</th>
<th>NIST Standard RBAC (\cite{4}\cite{5})</th>
<th>XACML Standard (\cite{10})</th>
<th>Access Restriction</th>
<th>Reasoner</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>GeoXACML profile</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>XACML profile X</td>
<td>+</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RBAC Constraints</td>
<td>X</td>
<td>+</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Specification and</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Enforcement of</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RBAC model</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>XACML + OWL</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>OWL profile</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X-STROWL</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(X: support totally, +: support partially)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

REFERENCES