Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp566-572)

An approach for Implementation of RBAC Models with Context Constraint to Business Process Systems

Yi Guo, Myeonggil Choi, Sangmun Shin, Guibam Bae, Yongsun Choi
Department of System management & engineering,
Inje University,
South Korea

http://bpm.inje.ac.kr

Abstract: Business Process Management (BPM) System has recently been paid much attention because they can support dynamic business processes over heterogeneous computing systems. However, most BPM systems merely support fundamental security services at during run time, such as authentication of users and network security. Apparently, to satisfy the real-time systems security requirement, it is more effective and secure to consider security issues during the processes' build time. In this paper, we describe an approach to implement the RBAC models with context constraint for business process system. Specifically, we utilize the RBAC models with context constraint mechanism to meet our needs and describe the security architecture to be applied to a BPM system. The intention of this paper is to extend RBAC models with context constraints to fulfill the requirements of BPM systems with respect to security, flexibility and expansibility.

Key-Words: RBAC, Context constraint, UML, Business Process Systems, Scenario-driven Role Engineering Process

1 Introduction

BPM systems are becoming more and more important for enterprises and organizations for facilitate their business process. Security and flexibility are the two important issues on electronic business system [11]. In the security issues aspect, Role-Based Access Control (RBAC) model [15] is a promising alternative to traditional discretionary and mandatory access control (DAC and MAC) model, which is regarded as a neutral policy and has been used in a variety of forms for computer system security [18]. According to RBAC, access authorization to specified objects, which is called permission or privilege, is assigned to roles rather than to individual users. Role is an abstract description of behavior and collaborative relation with others in the organization, which is comparatively steady and effective. Role also has hierarchical structures, which are designed according to both an organization's business activities and permission interpretation. Therefore, it can be easily adjusted and may reflect the dynamic adjustment for business as well. As a result, RBAC models provide intuitive support for expressing organizational access control policies, especially for addressing security issues in the environments of collaborative BPM systems. However, conventional RBAC lacks the ability to specify a fine-grained control on individual users in certain roles and on individual object instances. For collaborative environments, it is insufficient to have role permissions base on object types. Rather, it is often the case that a user in an instance of a role might need a specific permission on an instance of an object [19]. Moreover, constraints are an important aspect of RBAC and a powerful mechanism for laying out higher-level organization policy. However, the specification of such constraints has not been discussed in the conventional RBAC model [19]. Strembeck, M. and Neumann, G. introduced a framework for a special kind of RBAC constraints, called context constraints [17], which are defined as dynamic exogenous authorization constraints. In this paper, we provide an approach to implement such context constraints on RBAC model to build a process to implement the proposed security model to BPM systems. Our objective in this paper is to utilize the RBAC with context constraints to the existing business process model and have minimal changes to the existing models.

The organization of this paper is as follows: Next Section presents the preliminary knowledge of

RBAC and context constraints on RBAC models. In section 3, we provide a set of representation using object-oriented and UML of the RBAC with context constraints from static, dynamic and functional perspectives. In section 4, we propose the requirement of a BPM system and raise the security issue on the BPM systems. In section 5, we describe the whole architecture of a security model using RBAC with context constraints suitable for BPM systems. At last, we draw our conclusion and prospect the future work in these areas.

2 Literature review

Given the undeniable importance of RBAC models [15], researchers have done a considerable amount of research on modify this model to implement to different systems.

Bertino, Ferrari, and Atluri et al. [6] [7] [10] proposed an interesting and powerful constraint-based security model on logic predicates, that allows for different expressivity than the one presented here. Predicates in constraint expressions include predicates over a role graph and predicates over history (user and role that executed some task), task activation and task outcome (success, abort).

Bertino et al. [7] also describes optimizations of the basic constraint verification mechanism. However, at this point, there is not enough experience in this domain to further evaluate which method or combination of methods would reduce the overall cost of authorization related predicates, or even if any of those optimizations are necessary for the "average case".

Castano et al. [8] propose an active-rule based workflow security model which is implemented on top of the Wide workflow system. Event-Condition-Action rules are employed to specify instance, history and event constraints. Selection of agents to which tasks are assigned is discussed, e.g. first trying to assign tasks to authorized users places in lover positions of the role/level hierarchies. Actual support for policies is not presented in their paper. Castano et al. also include the notion of temporal constraints. Each of the relations of the meta-model can be annotated with a time specifier so that the validity of a relation, say can-play, between user and role, only holds during the specified time period.

Hung and Karlapalem [13] present the security features of the CapBasED-AMS workflow system and discuss the trade-off between security and risk of a system and present a metric to evaluate such trade-off. The risk factor is equated to the number of tasks any user executes in a given instance (case). The rationale is that users that perform more tasks are more knowledgeable about the task being performed and thus pose a higher security risk. Failure resilience, the ability to complete a task, on the other hand, would depend on more than one user being able to execute each task. A greedy algorithm is proposed, that determines task assignments that would achieve high failure resilience and low security risk factor according to these definitions. The need for overriding a workflow system is also discussed in Miller et al., in the context of a health care workflow application. The mechanism, implemented as part of the METEOR workflow system, is called "Break-Glass-Procedure", and allows certain authenticated users to temporarily assume greater privileges or higher roles. System response in this case is to employ maximal auditing/tracking and inform a workflow administrator.

A few commercial workflow systems offer some functionality related to role authorization constraints too, such as IBM WebSphere MQ and Staffware2000. [8] [11] [12].These commercial systems are not so expressive as the ones discussed in the present work. Mechanisms for overriding constraints such as the ones discussed here are also not offered by any of these commercial workflow systems [20] .

3 Context constraints on RBAC models

3.1 RBAC Model

Role-based access control (RBAC) [15] has rapidly emerged in the 1990s as a technology for managing and enforcing security in large-scale systems. It has recently received considerable attention as a promising alternative to traditional discretionary access control (DAC) and mandatory access controls (MAC). Intuitively, the basic notion of RBAC is that permissions are associated with roles, and users are assigned to appropriate roles. RBAC ensures that only authorized users are given access to certain data or resources. Therefore, RBAC policy is based on the roles of the subjects and can specify security policy by the way it maps the roles of the subjects to an organization's structure.

A general family of RBAC models called RBAC96 [15]. Figure 1 illustrates the most general model in this family. Fig. 1 shows roles and permissions that regulate access to data and resources. Intuitively, a user is a human being or an autonomous agent, a role is a job function or a job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of the role, and a permission is an approval of a particular mode of access to one or more objects in the system or some privilege to carry out specified actions. Roles are organized in a partial order ≥, so that if $x \geq y$ then role x inherits the permissions of role y. Members of x are also implicitly members of y. Each session relates one user to possibly many roles. The idea is that a user establishes a session and activates some subset of roles that he or she is a member of (directly or indirectly by means of the role hierarchy).

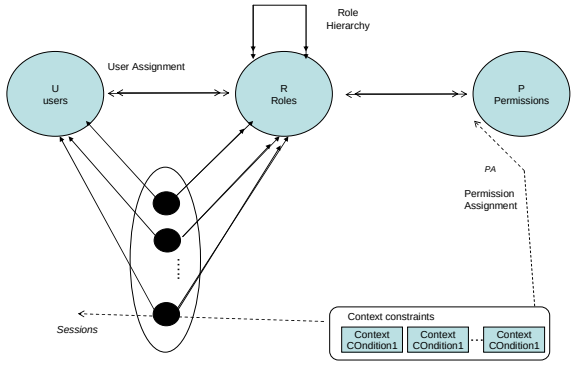

Fig. 1 Context constraints RBAC Models

From a policy perspective, the capability with RBAC to impose constraints on user membership by assigning users to roles provides a powerful means of enforcing conflict of interest and cardinality rules for roles as they uniquely apply to a collaborative environment [2]. Users can be easily reassigned from one role to another without modifying the underlying access structure. RBAC is thus more scaleable than user-based security specifications and greatly reduces the cost and administrative overhead associated with fine-grained security administration at the level of individual users, objects, or permissions.

Although traditional RBAC are very effective and popular for traditional and collaborative systems, RBAC has several weaknesses. First, tradition RBAC lacks the ability to specify a fine-grained control on individual users in certain roles and on individual object instances. For collaborative environments, it is insufficient to have role permissions based on object types. Rather, it is often the case that a user in an instance of a role might need a specific permission on an instance of an object. Another important issue in the RBAC model implementation is the power of constraints specification. Constraints are an important aspect of role-based access control and a powerful mechanism for laying out higher-level organizational policy. The importance of flexible constraints to support emerging applications has been recently discussed by researchers [2][19], However, the specification of such constraints have not been discussed in the RBAC model. In this paper, we will discuss the context constraints on RBAC models.

3.2 Context Constraint on RBAC Models

Commonly, RBAC services deployed in interactive environments often need to consider context information to enforce complex access control policies that rely on information like time, location, process-state, or access history. In principle, a central idea of a RBAC model is to support constraints on almost all parts of the RBAC model. However, it is often required to consider different context information in authorization decisions, especially in highly interactive environments.. In this paper, we propose context constraint as a means to consider context information in access control decisions. Strembeck, M. describes the categories of constraints on RBAC in terms of different perspectives. In the perspective of evaluate time, they differentiate between static and dynamic constraints: Static constraints can be evaluated at "administration time" of an RBAC model, while Dynamic constraints can only be checked at runtime according to the actual values of specific attributes or with respect to characteristics of the current session [16].

In the perspective of factors, RBAC constraints can be classified to the endogamous (model intrinsic) and exogenous (environmental) factors: Endogenous constraints relate to intrinsic properties of an RBAC model and inherently affect the structure and construction of a concrete instance of an RBAC model. While exogenous constraints either exclusively involve attributes that do not belong to the core elements of an RBAC model (e.g., time constraints that restrict role activation to a specific time interval or allow access operations for a particular resource only

on a specific weekday), or refer to external attributes or properties of a specific RBAC model element. In general, exogenous constraints are defined as conditions that take external data into account for certain operations or decisions of an access control service.

Based on the functions of RBAC constraints, they also can be subdivided into authorization constraints and assignment constraints: Authorization constraints place additional controls on access control decisions. Thus, even if a subject is in possession of a permission that grants a certain access request, the access can only be allowed if the corresponding authorization constraints are fulfilled at the same time. While assignment constraints control the assignment or activation of permissions and roles (e.g. maximum and minimum cardinalities or separation of duty constraints).

Intuitively, a context constraint specifies that certain context attributes must meet certain conditions to permit a specific operation. According to the above classification, context constraints should be described as dynamic exogenous authorization constraints [16]. Furthermore, A context constraint is defined through the terms context attribute, context function, and context condition [16].

3.3 UML Representation of RBAC on Context Constraint

Since the RBAC has become widely accepted as the proven technology, many security researchers and secure system developers have dedicated to develop role-based systems and inject role-base models to secure existing systems. In order to give a sound blueprint to system analyzers and developers, it is necessary to represent RBAC models with a general-purpose visual modeling language. In this paper, we utilize the Unified Modeling Language (UML) to represent the RBAC model in terms of context constraint on the model.

The Unified Modeling Language (UML) is a general-purpose visual modeling language in which we can specify, visualize, and document the components of a software system [5] **Error! Reference source not found.**. It captures decisions and understanding about systems that must be constructed. The UML family consists of use case modeling, static modeling, and dynamic modeling. According the components of RBAC model and the context constraints on RBAC models, we modify class diagrams conventional representation for a static view of the RBAC models [16] with context constraints.

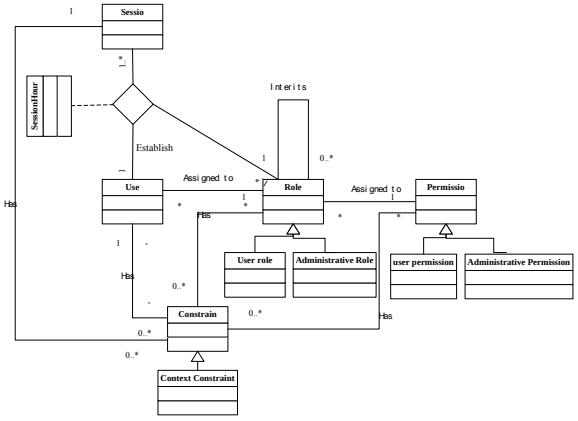

Fig.2 Class Diagram: for Context Constraint on RBAC models

4 Implementation of Context constraint RBAC to Business Process Systems

4.1 Security requirement of business process systems

A business process is a business procedure involving the coordinated execution of multiple tasks performed by different processing entities. With the great popularity of IT technology, especially web technology, web-based business process management systems are being deployed over the enterprise distribute computing environment [9]. In this case, we need to consider several security issues for web-based workflow systems such as authentication of the user, net work security for data transport, and access control [3]. Many processes are security critical in the sense that security requirement are a central part of process requirements and security mechanisms are required for their realization. Examples range from the authorization of a military engagement, to an enterprise purchase process, to even the coordination of the sequence of user interface masks displayed to a user. In a decentralized workflow management model (DWFMS) [3] [4], where the inter-mask dependencies are enforced without having to have a centralized WFMS, the partition of workflow and building workflow stub on each agent need to consider the security issue.

4.2 The proposed RBAC secure architecture of business process systems

Here, we modify the RBAC model for workflow system architecture [18] and proposed our new layer to this architecture. The integrated architecture is shown as Figure 3. There are five levels. From bottom to top are context, permission, workflow, role, and user denoted with level 1-5. context-constraints is elicited and packed and adds to permission layer, while permission is not directly assigned to role, but capsulated in activity definition corresponding to current business showing as thin line from level1-3, and activity has both separability and correlation so that workflow may be dynamically assembled with them to satisfy the flexibility [18].

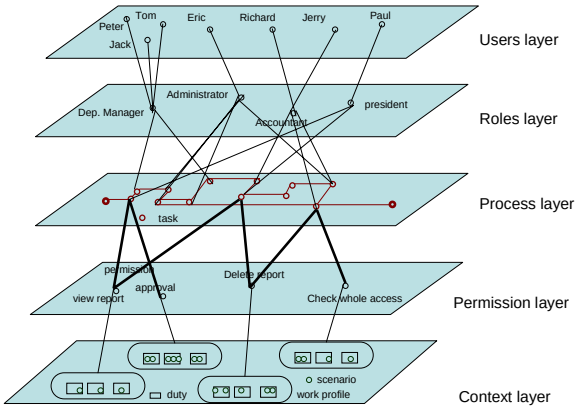

Fig. 3 Architecture of Context- constraints RBAC BPM Model

4.3 Elicitation and specification of context constraints

With respect to access control, one has to ask first which parts of these unmanageable quantities of context information are relevant for a specific authorization decision, and how the corresponding information may be elicited and defined on the modeling level. In this section, we therefore suggest a process for the elicitation and specification of context constraints. Prior to describing the engineering of context constraints in detail, we give some background information concerning the scenario-driven role engineering process. In the scenario-driven role engineering process usage, scenarios of an information system are used to derive permissions and to define tasks. In general, a *scenario* describes an action and event sequence, for example, to register a new patient in a hospital information system. Thus, each scenario consists of several steps, and a subject performing a scenario must possess all permissions that are needed to complete the different steps of this scenario. In turn, a task consists of one or more scenarios, and tasks are combined to form work profiles. A work profile comprises all tasks that a certain type of subject is allowed to perform. In a hospital environment different work profiles for physicians, nurses, and clerks are needed, for instance. In the role engineering process, work profiles are then used together with the permission catalog and the constraint catalog to define a concrete RBAC model. However, the scenario-driven approach presented in [16] only provides general guidance for the sub process of defining (exogenous) constraints. This fact and our aim to specify and enforce context constraints in an RBAC environment led us to the definition of the process extension proposed in this section.

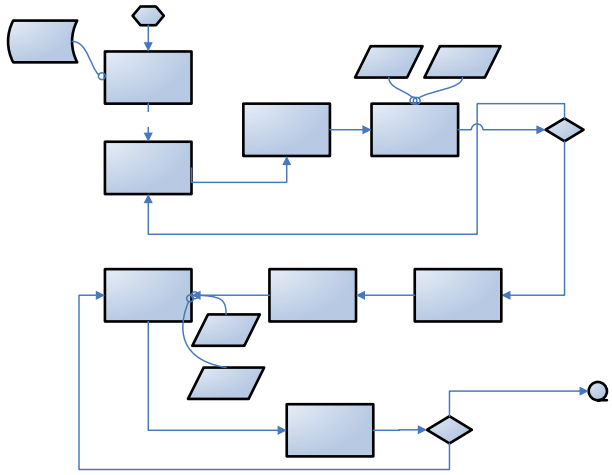

Fig. 4 An Activity Diagram for the elicitation and specification of context constraints

Figure 4 shows an activity diagram for the process of elicitation and specification of context constraints. The engineering of context constraints is in essence a requirements engineering process. To elicit context constraints we especially use goals, which are a familiar concept in the area of requirements engineering [16].Goals are suitable to be applied in combination with scenarios to elicit and define requirements and to drive a requirements engineering process. Intuitively, a goal is an objective that the system should or must achieve [16]. An obstacle is an undesired condition, which obstructs the fulfillment of one or more goals. Therefore, obstacles can be regarded the opposite of goals. Scenarios and the scenario model serve as the basis for the scenario-driven role engineering process [16]. The first step of the constraint engineering process shown in

Figure 4 is to fetch the current scenario model. The following activities are described in more detail in [16].

5 Conclusions

In this paper, we developed models, mechanisms and architectures for implementing the RBAC with context constraints models for BPM systems. These works minimize the changes on existing BPM systems and then propose a principle to evaluate the permission through elicit the context constraints from the context layer of BPM system. Even though this work is applied to an existing BPM system, we believe that this architecture can be deployed into several application domains such as large-scale collaboratory environments and electronic commerce systems.

References:

[1] Ahn, G.J. and Sandhu, R. "Role-based authorization constraints specification", *ACM Tran. Inf. Syst. Secur. 3, 4 (Nov.)* 2005.

[2] Ahn, G.J., Sandhu, R., Kang, M., and Rark, J. "Injection RBAC to secure a Web-based workflow system." *In Proceedings of 5th ACM Workshop on Role-Based Access Control. Berlin, Germany,* 2000, pp. 1-10.

[3] Atluri V. "Security for Workflow Systems", *Technical Report, Elsevier Science*, 2001.

[4] Atluri, V., Chun, S.A., and Mazzoleni, P., "A Chinese wall security model for decentralized workflow systems", *Proceedings of the 8th ACM conference on Computer and Communications Security*, 2001, pp. 48-57.

[5] Basin, D. Jurgen Doser and Torsten Lodderstedt, "Model Driven Security for Process-Oriented Systems", *Proceedings of the eighth ACM symposium on Access control models and technologies*, 2003, pp. 100-109.

[6] Bertino, E., Ferrari, E. and Atluri, V., "A flexible model supporting the specification and enforcement of role-based authorization in workflow management systems", *Proceeding of the Second ACM workshop on Role-Based Access Control*, 1997, pp. 1-12

[7] Bertino, E., Ferrari, E. and Atluri, V., "The specification and enforcement of authorization constraints in workflow management systems", *ACM Transactions of Information and System Security 2, 1* ,1999, pp. 65-104.

[8] Castano, S., Casati, F. and Fugini, M., "Managing workflow authorizaiton constraints through active database technology", *Information Systems Frontiers 3, 3*, 2001.

[9] Choi, Y. and J.Leon Zhao, "Decomposition-Based Verification of Cyclic Workflows", *Lecture Notes in Computer Science*, Vol. 3707, 2005, pp. 84-98.

[10] Ferraiolo, D., Cugini, J. and Kuhn, R., "Role-based access control (RBAC): Features and motivations" *Proceedings of 11th Annual Computer Security Application Conference, New Orleans, LA, December* 1995, pp241-248.

[11] Holbein, R. and Teufel, S. "A context Authentication Service for Role-based Access Control in Distributed System --- CARDS", *In Proceeding of IFIP 11th International Conference on Information Security*, 1995.

[12] Hristo Koshutanski and Fabio Massacci, "An Access Control Framework for Business Processes for Web Services", *ACM Workshop on XML Security,* 2003, pp. 15-24.

[13] Hung, P.C.K. Karlapalem, K. and Gray J.W., "A study of least privilege in CapBasED-ams", *Proceeding of the 3rd IFCIS International Conference on Cooperative Information Systems,* 1998, pp. 208-217.

[14] Sandhu, R.S., "Transaction control expressions for separation of duties", *Proceeding of the Fourth Computer Security Applications Conference*. 1988, pp. 282-286.

[15] Sandhu, R.S., Coyee, E.J., Ferinstein, H.L. and Youman, C.E. (1996), "Role-based access control model [J]", *IEEE Computer,* Feb.1996, 29(2), pp. 38-47.

[16] Michael E. and Gail-Joon A., "UML-Based Representation of Role-Based Access Control", *Proceedings of 5th IEEE International Workshop on Enterprise Security, NIST,* 2000, pp. 195-200.

[17] Strembeck, M., and Neumann, G., "An Integrated Approach to Engineer and Enforce Context Constraints in RBAC Environments", *ACM Transactions on Information and System Security (TISSEC) Volume 7 , Issue 3,*2004, pp. 392-427.

[18] Sun, Y. and Pan, P., "PRES- A Practical Flexible RBAC Workflow System", *Proceedings of the 7th international conference on Electronic commerce*, 2005, pp. 653-658.

[19] Tolone, W., Ahn, G.J. and Pai, T. "Access Control in Collaborative Systems" *ACM*

Computing Surveys, Vlo.37, No.1, 2005, pp. 29-41.

[20] Wainer, J. Barthelmess P. and Kumar, A., "W-RBAC — A Workflow Security Model Incorporating Controlled Overriding of Constraints", *International Journal of Cooperative Information Systems, Vol. 12, No. 4*, 2003, pp. 455-485.