

Context-based Role Engineering for Ubiquitous Computing Environment

Seon-Ho Park, Jung-Ho Eom, and Tai-Myoung Chung
Internet Management Technology Laboratory,
School of Information and Communication Engineering,
Sungkyunkwan University,
300 Cheoncheon-dong, Jangan-gu,
Suwon-si, Gyeonggi-do 440-746, Republic of Korea
{shpark, yjhan}@imtl.skku.ac.kr, tmchung@imtl.skku.ac.kr

Abstract

Existing context-based access control mechanisms have a problem of tradeoffs between the expressive power and the management efficiency of authorization policy. In this paper we introduce the context-aware role engineering which is a new concept for context-based access control and context-aware service scenario driven role engineering approach. The context-aware role engineering enables that context-based access control can be implemented using basic RBAC₃ model with context constraints.

Keywords— RBAC, Role Engineering, Context-Awareness

1. Introduction

With the growing progress of ubiquitous computing technologies, a considerable number of studies have been conducted on context-based access control (CBAC) models for the ubiquitous computing environment[1, 2, 4, 10, 11, 15, 17, 18, 22]. The CBAC models enable using context information for a configuration of authorization policies and an enforcement of the policies. Most studies on the CBAC models are based on the role based access control (RBAC) model[9], and focus the role enabling/disabling or the role activation/deactivation based on the context such as temporal information, location information, and so on. Such context-dependent role enabling or activation mechanisms focus context-aware role management but does not consider the context-based permission management. GTRBAC model[11] addressed the context-based permission management using context-based assignment event such as “assign/deassign p to r ”, but it provides too complex policy rules compared to traditional RBAC model.

In this paper, we analyze the existing CBAC models based on RBAC model, in perspective of an expressive

power of authorization model and efficiency of authorization management. From the result of analysis, we address a problem of tradeoff between expressive power and efficiency of authorization model occurring due to expression of context-related component or constraints. We introduce the concept of context-based role engineering, so called CONRE, which conducts role and permission modeling through role engineering at the level of context-aware service and application. Role engineering is one of the essential phases in implementation of RBAC-based security architecture. The general function of the role engineering is to provide methods and rules for definition of roles and permission-role assignment. The motive of CONRE is to enable implementing CBAC, just using a RBAC₃ model. The scope of this paper is motivating and introducing of CONRE, and providing the first method of CONRE which is the Context-Aware Service Scenario-Driven Role Engineering, called CASDRE. The CASDRE method is based on the scenario-driven role engineering method[16]. In CASDRE, a set of context-aware service scenarios is generated, then using these scenarios, several role engineering sources such as a permission catalog, a task profile, a SoD constraints catalog, and a context-aware constraints catalog are generated for modeling RBAC system.

We proceed as follows. For starter, we describe briefly RBAC model, role engineering and CBAC models. We analyze problems of existing CBAC and introduce the concept of CONRE. Subsequently, we address detailed process of CASDRE and conclude this paper.

2 Preliminaries

2.1 Role-Based Access Control

RBAC is the most popular access control model which is used broadly in many application domains. In RBAC

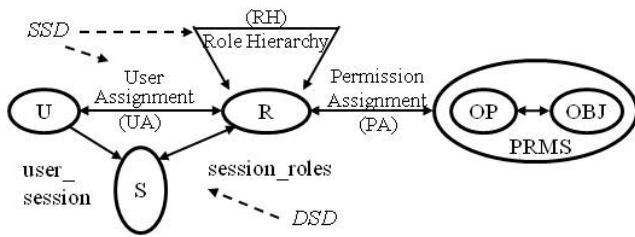


Figure 1. Components and their relations in RBAC Model

model, permissions are associated to roles (PA) and users are assigned to the roles (UA)[9]. The role concept is similar to the notion of grouping of users or permissions, but the role concept includes responsibility and job functions of users in the level of applications or enterprise therefore, the RBAC model is suitable to administrate complex policies for large, structured organizations. RBAC model consists of RBAC₀ model which is core model and includes users, roles, permissions, sessions, UA, and PA, RBAC₁ model which is an hierarchical RBAC and includes elements of RBAC₀ and the role-role assignment, RBAC₂ model which is a constraint RBAC provides SoD principles, and RBAC₃ model which consolidates RBAC₁ and RBAC₂. Figure 1 illustrates RBAC model[9].

2.2 Role Engineering

Role engineering (RE) is one of the important processes to implement and operate RBAC system, which of the concept is introduced by Coyne[5]. RE is an approach to defining roles, permissions, constraints and role-hierarchies and assigning permissions to the roles. RE provides functions of defining roles and naming them, defining permissions and assigning them to roles, and defining constraints, which are required for implement RBAC system in accordance with business rules, security rules, and an organization's structure. Ultimately, the purpose of RE is designing every elements of RBAC₃ model except for the user-role assignment.

The approach to conduct RE process divides into a top-down approach and a bottom-up approach. The top-down role engineering (TDRE) approach is based on analysis of structures of organization and system and business policy and rules[7, 8, 12, 16, 19, 21]. The bottom-up role engineering (BURE) deduces roles, role hierarchy, permission, and user-role assignment through analyzing existing assignment relations of users and permissions which were already defined and used past[6, 13, 14, 20, 24, 23]. Basic principle of BURE approach is to get a user-role assignment matrix and a role-permission assignment matrix through decomposing an existing user-permission assignment matrix.

This method enables to reduce the cost of RE process because of using the existing user-permission assignment relations, while the result of RE process can be insufficient for implementing complete RBAC system because it does not consider organization's structure and business rules. On the other hand, the approach to defining roles in TDRE is based on various data such as business process and strategies, organization's structure, human-resource specifications, and so on; therefore it is a method which can reflect organization's features well. TDRE approach enables that defining role is near completion, while it has drawbacks that most process depend on manual mechanism and the cost of RE process is high.

2.3 CBAC Mechanisms

Existing most researches on CBAC mechanism or model are based on RBAC model. That is, they are the form of extended RBAC model, such as defining context dependent constraints or adding new context-processable element into RBAC model. To address temporal context based access control requirements, Bertino et al. propose a Temporal Role Based Access Control (TRBAC) model [1]. Covington et al. tried to adapt environmental information to role based access control and introduced the notion of environment role [4]. Moyer and Abamad proposed Generalized Role Based Access Control (GRBAC) model[15]. GRBAC model enhanced traditional RBAC by incorporating the notion of object roles, and environment roles, with the traditional notion of subject roles. Park et al. presented Context-Role Based Access Control (CRBAC) model[17]. They introduced the notion of the context-role which facilitates configuration and management of context-related policies for context-based authorization and access control.

The CBAC mechanisms described above are extended RBAC models which control activation/deactivation of roles assigned to users using context-related rules. Such context based role activation and deactivation approaches does not consider context dependency of permissions assigned to roles. That is, they cannot enforce fine-grained context-dependent authorization control because of context-based permission assignment. Generalized Temporal Role Based Access Control (GTRBAC) model[11], which generalizes the TRBAC model to incorporate a set of language constructs for the specification of various temporal constraints on roles, provides temporal constraint on permission assignment. GTRBAC model has, however, the policy management overhead problem which is occurred due to defining temporal constraints on permission assignment, user assignment, and roles.

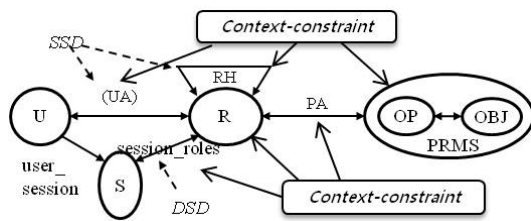


Figure 2. CRC-CBAC model

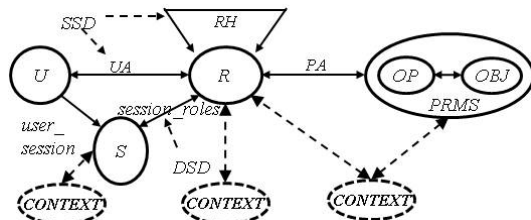


Figure 3. CRE-CBAC model

3 Towards Context-Aware Role Engineering

3.1 Problem of Existing CBAC models

In problems of existing CBAC models we focus on a tradeoff problem which can occur due to expressive power and management efficiency of policy. When extending RBAC to context-awareness, it is required defining context-related policy rules and its method is divided into a method adding context-related element (CRE) and a method applying context-related constraint (CRC) to the role or assignment relations of elements in RBAC. Both methods using CRE and CRC cause a problem increasing policy management overhead. That is, if the expressive power of context-related policy increase, then the efficiency of policy management decrease.

Figure 2 and figure 3 show generalized concepts of CBAC models based on application and extension of RBAC model. Figure 2 represents CBAC models based on the CRC approach and we call these CBAC models CRC-CBAC. Generally CRC-CBAC models[1, 3, 11] have six CRC Applying Points (CAPs) as you can see in figure 2. Because various contextual constraints such as temporal, spatial, environmental constraints and system resources conditions can be defined, if every six CAPs are applied the authorization policy management becomes very complex. In addition, because the constraint is non-abstracted information specifying concrete conditions, reuse of constraint policy is difficult unlike concept of role or permission. So, there is a problem that as security rules to be defined increase the number of CRC increase together.

Figure 3 represents CBAC models based on the CRE approach, so called CRE-CBAC models. CRE-CBAC mod-

els use extended role concept which is made for abstraction of various context information[2, 15, 17], or add new elements applying context awareness into assignment relations to RBAC model[2, 10, 18]. GEO-RBAC model includes a new element which enables activating user-role assignment relations only if certain context conditions are satisfied[2]. Similar to the concept of extending UA, there have been studies on adding new elements extending permission-role assignment relation to RBAC model[2, 10, 18]. There are several advantage of CRE-CBAC model. Firstly, even if context-related access control rules very increase, the number of polity rules are added in overall policy rules becomes to be small relatively. Second, even if values of attributes in defined CRE change, policy rules configured with that CRE will not change, because of the abstraction feature of CRE. There are, however, several weaknesses of CRE-CBAC. CRE-CBAC models are inflexible as compared with CRC-CBAC models and the expressive power of CRE-CBAC is also inferior to that of CRC-CBAC, because the number of CAPs in CRE-CBAC models is generally smaller than that in CRC-CBAC models.

RBAC is the most popular access control mechanism which is already applied into many enterprise IT solutions, database products, and so on; therefore it is inefficient to modify features or functions of RBAC systems which are already implemented. In addition, to use the modified RBAC mechanism to applying CRE or CRC cannot be considered as the best solution for context-based access control because many users have used solutions based on RBAC standard such as NIST RBAC which is verified as stable model for a long time.

3.2 Context-Aware Role Engineering

In the previous section, we discussed problems of researches on existing CBAC models. As a solution of that problem, we introduce Context based Role Engineering, so called CONRE approach. The purpose of CONRE is that context-aware access control can be implemented only using RBAC₃ model with context constraints. For this, CONRE creates roles, permissions, context constraints based on security and management considerations which are deduced through analyzing requirements of context-aware services. The effects of CONRE are as following:

- The number of CAPs can be reduced when configuring RBAC policy using CRC
- It is possible to deduce context-aware fine-grained roles and optimize role hierarchy in accordance with context constraints.
- CRE approach does not be required any more, because roles are created considering context-aware service,

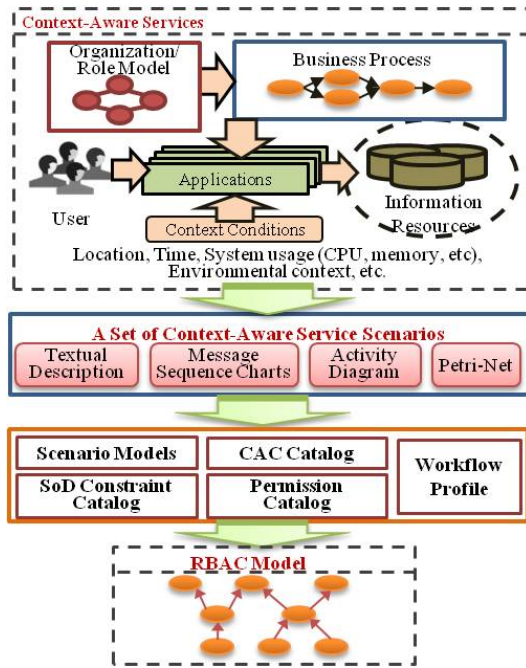


Figure 4. Overview of Context-Aware Service Scenario-Driven Role Engineering

therefore it is sufficient representing CBAC policy using context constraints and the fine-grained context-aware user-role assignment.

- It is possible to deduce previously context constraints to activate or deactivate roles, as well as defining context-aware fine-grained roles.

4 Context-Aware Service Scenario-Driven Role Engineering

4.1 CASDRE Overview

In this section we introduce Context Aware Service Scenario-Driven Role Engineering (CASDRE) approach as the first method for CONRE. CASDRE is engineering approach providing RBAC modeling functions suitable for context-aware service using scenario-driven role engineering proposed by Neumann's work[16]. Basic principle of CASDRE is to make scenarios for context-aware service, to extract roles, permissions, and CACs from those scenarios, finally to define relations and assignment among roles, permissions, and CACs.

Figure 4 shows general operations and functions of CASDRE. First step of CASDRE is to deduce draft scenarios of context-aware services considering various elements and

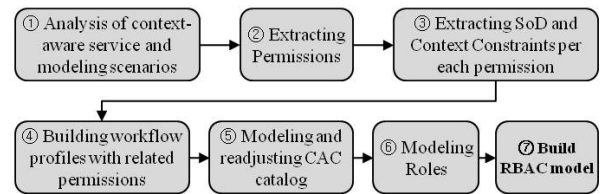


Figure 5. CASDRE Process

conditions such as users, resources, applications, an organization's structure, a business process, and context aware restriction conditions of context-aware service. To represent context-aware service, it can be used, various scenario definition languages or tools such as message sequence chart, activity diagram, petri-net, and UML. After draft scenarios are created, those scenarios are completed through verification step. With the scenarios, we can start role engineering processes which consists of extracting a permission catalog, a SoD constraints catalog, and a CAC catalog from the scenarios, and creating work-profile required for a role hierarchy, UA, and PA.

4.2 CASDRE Process

In this section we address CASDRE process. CASDRE operates in accordance with the process in figure 5.

4.2.1 Modeling Context Aware Service Scenario

In first step modeling context-aware service scenario is conducted through the analysis of context-aware services and applications. Firstly, every service scenarios are written in a method of text-based description. Then basic elements of service, such as user(subject), service objects, operation types, and types and identifiers of applications, are identified from the text-based scenario description. Next, the MSC specifying scenario is written using the elements. MSC is used to represent easily relations of subject and object, operations and application types, and context conditions using flowed and visual notations. Draft scenario needs to be verified manually by expert. Finally completed scenarios are inserted into the set of scenario models.

4.2.2 Permission Derivation

In this step we identify the object set and the operation set from every scenarios and extract a set of pairs $\langle obj, op \rangle$ ($obj \in OBJ, op \in OP$) from MSC of scenarios. These $\langle obj, op \rangle$ pairs require to be abstracted for the RBAC modeling. The abstraction means naming and identifying to permissions.

4.2.3 Creation of SoD constraints and CC

In this step we define constraints for SoD principles and context-aware access control. The constraints for context-aware access control is called the context condition (CC). CCs are statements to represent context conditions and constraints and written using the context attribute (CA) and the context process function (CPF). The CA is required to specify types of context conditions. That is, it means attributes specify type and scope of data to represent context information such as time, date, location, and so on. CPF is a tool to provide mechanism for processing context. For instance, a predicate *location()* is used as a CPF to process a location information. CC provides functions to represent contextual constraint conditions which are related every permissions using the CPF.

4.2.4 Building Task Profiles

In this step, task profiles are created using related permissions. Permissions in scenarios can have functional connections or be related mutually for same purpose. We can make a set of related task and the workflow by collecting and abstracting these related permissions. Task is a set of related permissions which of executions are not interrupted and is conducted generally by one role. Workflow is an ordered set of related tasks which have same business purpose and dependencies.

That is, the main function of this step is to create a set of tasks and task profiles by analyzing and abstracting related permissions.

4.2.5 CAC Modeling and Adjusting CAC Catalog

In this process CAC modeling is conducted based on CCs and Task Profiles which are created in step 3 and step 4. CCs deduced in step 3 are specifying and identifying contextual conditions which are mutually exclusive in aspect of executing permissions. That is, these CCs become basic CACs for conducting CBAC in the level of permission.

If aggregating CCs is possible and also required in accordance with task profiles which are created in the step 4, we can create high level CACs by collecting CCs. Note that PA must be considered as an applying point of CACs. The relation between CAC and PA must be a relation of N:1. If that is 1:N or M:N, defining additional CAE is required, and it becomes difficult to express sufficiently context-based authorization policy using RBAC.

When aggregating CACs is completed, we should define the CAC applying point, assign identifiers to CACs, create and store CAC catalog.

4.2.6 Context-based Role Mining

Role mining is the most important step of the role engineering process. At this step, roles are created using constraints catalog, CAC catalog, and task profiles. All permissions which compose one task profile can be assigned one user role and this method is enough for modeling basic RBAC. For modeling context-aware RBAC, if there are two or more relations between task profile and CAC, permissions in the task profile must be separated by CAC.

```
for each task-profiles in profile-set{
    create cac-set;
    insert all CACs related with task-profile into cac-set;
    record the cardinality of cac-set;
} for each task-profiles in profile-set{
    for each cac's in cac-set {
        create perm_set[i];
        i = 0;
        for each permissions in task-profile {
            if (permission is associated with cac) {
                insert permission into perm_set[i];
            }
        }
    }
} for each perm_set in task-profiles {
    create role and assign permissions in perm_set;
}
```

Figure 6. Context-based role mining algorithm

4.2.7 RBAC Model Definition

This step builds RBAC policy using temporary role model, permission catalog, constraints catalog, and CAC catalog. In this step, unnecessary elements which are generated in the process of modeling RBAC policy should be removed and relationships of assignment among RBAC elements are defined based on task-profile and scenario model.

5 Concluding Remarks and Future Works

In this paper, we discussed existing researches on context-based access control model and pointed out that there is a problem of tradeoff between expressive power and management efficiency of policy in existing context-based access control models. As a solution, we introduce new role engineering approach, "Context-based Role Engineering (CONRE)" and "Context-Aware Service Scenario Driven Role Engineering (CASDRE)" which is the first method realizing the CONRE. Although the CONRE is at

initial level and CASDRE mechanism is just based on existing scenario-driven role engineering approach, the CONRE can offer a promising approach for the next generation of role engineering for context-based access control.

In future works, we will apply several context aware service scenarios to CASDRE and study on bottom-up approach of CONRE using previously defined contextual constraints. We briefly discuss about the problem of tradeoff between expressive power and management efficiency of policy in existing context-based access control models, because of limited paper space. In future works, we will provide mathematical and computational analysis to prove the problem.

References

- [1] E. Bertino, A. Bonatti, Piero, and E. Ferrai. Trbac: A temporal role-based access control model. *ACM Transactions on Information and System Security*, 4(3):191–233, August 2001.
- [2] E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca. Georbac: A spatially aware rbac. In *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies*, pages 29–37, Stockholm, Sweden, 2005.
- [3] L. Chen and J. Crampton. On spatio-temporal constraints and inheritance in role-based access control. In *proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security (ASIACCS 2008)*, pages 156–167, 2008.
- [4] M. J. Covington, W. Long, S. Srinivasan, A. K. Dev, M. Ahamad, and G. D. Abowd. Securing context-aware applications using environment roles. In *proceedings of the sixth ACM symposium on Access Control models and technologies*, pages 10–20, 2001.
- [5] E. Coyne. Role engineering. In *Proceedings of the first ACM Workshop on Role-based access control*, page 4. ACM, 1995.
- [6] A. Ene, W. Horne, N. Milosavljevic, P. Rao, R. Schreiber, and R. Tarjan. Fast exact and heuristic methods for role minimization problems. In *Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 1–10. ACM, 2008.
- [7] P. Epstein and R. Sandhu. Engineering of role/permission assignments. In *proceedings of the 17th Annual Computer Security Applications Conference.*, pages 127–136. IEEE Computer Society, 2001.
- [8] E. B. Fernandez and J. Hawkins. Determining role rights from use cases. In *proceedings of the second ACM Workshop on Role-based access control*, pages 121–125, 1997.
- [9] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli. *Role-Based Access Control*. Artech House, 2003.
- [10] F. Hansen and V. Oleshchuk. Srbac: A spatial role-based access control model for mobile systems. In *Proceedings of the 7th Nordic Workshop on Secure IT Systems*, pages 129–141, October 2003.
- [11] B. D. Joshi, J., E. Bertino, U. Latif, and A. Ghafoor. Generalized temporal role based access control model (gtrbac) (part i) - specification and modeling. Technical Report CE-RIAS TR 2001-47, Purdue University, USA, 2001.
- [12] A. Kern, M. Kuhlmann, A. Schaad, and J. Moffett. Observations on the role life-cycle in the context of enterprise security management. In *Proceedings of the seventh ACM symposium on Access control models and technologies.*, pages 43–51, 2002.
- [13] M. Kuhlmann, D. Shohat, and G. Schimpf. Role mining - revealing business roles for security administration using data mining technology. In *Proceedings of the eighth ACM symposium on Access control models and technologies*, pages 179–186, 2003.
- [14] I. Molloy, H. Chen, T. Li, Q. Wang, N. Li, E. Bertino, S. Calo, and J. Lobo. Mining roles with semantic meanings. In *Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 21–30. ACM, 2008.
- [15] M. J. Moyer and M. Ahamad. Generalized role based access control. In *proceedings of The 21th International Conference on Distributed Computing Systems (ICDCS)*, pages 391–401. IEEE Computer Society, April 2001.
- [16] G. Neumann and M. Strembeck. A scenario-driven role engineering process for functional rbac roles. In *Proceedings of the seventh ACM symposium on Access control models and technologies*, pages 33–42, 2002.
- [17] S.-H. Park, Y.-J. Han, and T.-M. Chung. Context-role based access control for context-aware application. In *proceedings of International Conference on High Performance Computing and Communications, LNCS4208*, September 2006.
- [18] I. Ray and M. Toahchoodee. A spatio-temporal role-based access control model. In *Proceedings of the 21th Annual IFIP WG 11.3 Working Conference on Data and Application Security, LNCS4602*, pages 211–226, July 2007.
- [19] H. Roeckle, G. Schimpf, and R. Weidinger. Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization. In *Proceedings of the fifth ACM Workshop on Role-based access control*, pages 103–110, 2000.
- [20] J. Schlegelmilch and U. Steffens. Role mining with ORCA. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 168–176. ACM, 2005.
- [21] D. Shin, G.-J. Ahn, and S. Cho. On modeling system-centric information for role engineering. In *proceedings of the eighth ACM symposium on Access control models and technologies*, pages 169–178, 2003.
- [22] M. Strembeck and G. Neumann. An integrated approach to engineer and enforce context constraints in rbac environments. *ACM Transactions on Information and System Security*, 7(3):392–427, 2004.
- [23] J. Vaidya, V. Atluri, Q. Guo, and N. Adam. Migrating to optimal RBAC with minimal perturbation. In *Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 11–20. ACM, 2008.
- [24] J. Vaidya, V. Atluri, and J. Warner. RoleMiner: mining roles using subset enumeration. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 144–153. ACM, 2006.