J Intell Manuf (2010) 21:607–612
DOI 10.1007/s10845-008-0208-z

New role-based access control in ubiquitous e-business environment

Sejong Oh

Received: 25 May 2008 / Accepted: 17 November 2008 / Published online: 30 November 2008
© Springer Science+Business Media, LLC 2008

Abstract Ubiquitous e-business is one of major topics in intelligent manufacturing systems. Ubiquitous e-business environment requires security features including access control. Traditional access control models such as access control list (ACL), mandatory access control (MAC), and role-based access control (RBAC) are unsuitable for a ubiquitous e-business environment because they cannot satisfy its requirements. In this study, we propose a new access control model termed the Ubi-RBAC model. It is based on the RBAC model and adds new components such as space, space hierarchy, and context constraints. Ubi-RBAC covers the context awareness and mobility of subjects (human users), which are the key issues of access control in the ubiquitous e-business environment.

Keywords IMS · Access control · RBAC · Ubiquitous e-Business · Context awareness

Introduction

The Intelligent Manufacturing Systems (IMS) program is an industry-led, global, collaborative research and development program established to develop the next generation of manufacturing and processing technologies (IMS 2008). One of the IMS technical themes is mobile and ubiquitous e-business and e-work. In spite of the importance of the theme, security issues were not considered enough. Access control is an important security issue. Large organizations or information systems require an access control mechanism. The basic purpose of access control is to offer a methodology by which only authorized users (subjects) can access information resources (objects). Access is the ability to perform tasks such as reading, writing, and the execution of system resources. Access control is a means to control the ability to perform the above mentioned tasks. The access control of computer systems describes whether or not specific users or processes can access specific system resources, and their allowed access type. Some of the well-known access control models are access control list (ACL), discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC).

Developing an access control model involves the process of modeling the security features of the real world. Therefore, access control models reflect the real world. For example, the "role" concept of the RBAC model reflects the job function or job position of a company and the "subject" reflects a human user. If the requirements or environment of the real world change, a new access control model will be required. We expect that ubiquitous computing including ubiquitous e-business will be a new environment in the near future. In the ubiquitous computing environment (UBE), traditional access control models would be unable to cover new security requirements.

Weiser (1991) has revealed that ubiquitous computing implies enabling the availability of many computers throughout the physical environment, while effectively making them invisible to the user. Ubiquitous computing is considered by some to be the third wave of computing. The first wave was many individuals per computer, and the second wave was one individual per computer. The third wave will be many computers per individual. Three key technical issues are as follows: power consumption, user interface, and wireless connectivity. In the traditional environment, access control is implemented inside a computer system and connected network. The subject is a user who logs on to the system, and

S. Oh (✉)
Department of Computer Science, Dankook University, Cheonan, Korea
e-mail: sejongoh@dankook.ac.kr

www.manaraa.com

the object is the file, program, or tables in a database. In the UBE, access control is implemented in a special "space" such as an office or workshop. In the UBE, subjects do not log on explicitly. Instead, a sensing device specifies the subjects and maintains a dynamic session. A telephone, fax, milling machine, press, and bar code scanner are examples of objects in the UBE. It is clear that the UBE is different from the traditional environment. As a result, well-known access control models cannot be directly applied to the UBE. Figure 1 shows the difference between the traditional environment and UBE from the viewpoint of access control.

In this study, we propose an access control model for the UBE termed the Ubi-RBAC model. We consider the RBAC model as the base model and add extra components for the UBE (Sandhu et al. 1996; Ferraio et al. 1995; Sandhu 1995; Gavrila and Barkley 1998; Sandhu et al. 1999). The fundamental concept of the RBAC model is to prevent users from accessing company information at their discretion. Instead, access rights are associated with roles, and users are assigned to suitable roles. The notion of a role is an enterprise or organizational concept. Therefore, RBAC enables us to model security from an enterprise viewpoint since we can align security modeling with the roles and responsibilities in the company. This greatly simplifies the management of access rights. Figure 2 shows the basic RBAC model.

The basic RBAC model is designed for the traditional environment; authorization and the access control process are static. The UBE requires a dynamic adjustment of sessions, user–role assignment, and permission–role assignment

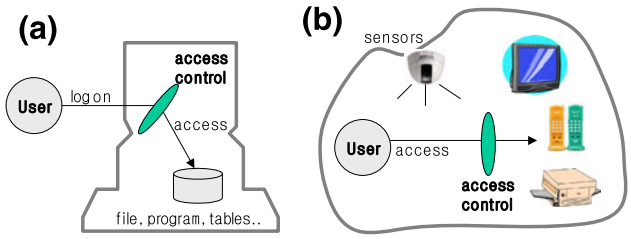

Fig. 1 Traditional environment vs. UBE. **a** Traditional environment, **b** Ubiquitous Computing Environment (UBE)

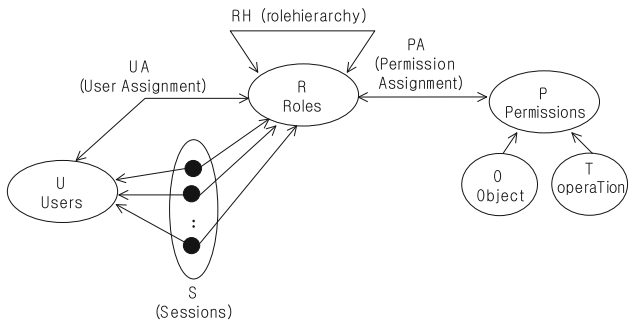

Fig. 2 Role-based access control model

according to the context information. It also requires the management of space as a component of access control. Users can navigate among spaces, and their authority is dynamically changed depending upon their location. The Ubi-RBAC model accepts these requirements and proposes the concept of a "space hierarchy."

This paper is organized as follows. Section "Motivations and related works" introduces our motivations and related works. It also discusses the characteristics of access control in the UBE. Section "Ubi-RBAC model" introduces the Ubi-RBAC model and provides a description of the model, space hierarchy, and access control principle of Ubi-RBAC. In section "Discussion" we discuss the Ubi-RBAC model and compare it with other models. Finally, the paper is concluded in section "Conclusion".

Motivations and related works

From the viewpoint of access control, the requirements of the UBE are as follows (Oh and Park 2004):

(1) A user (subject) does not explicitly log on to a system or space. Instead, sensing devices specify the user and maintain an implicit session. For example, an authorized worker switches on a machine without logging on. For simplicity, we assume that the user can be distinguished by his/her ID card.
(2) In the UBE, the object of access control is not a file or tables in a computer. Instead, the object is an intelligent device such as a copy machine, conveyor belt, and telephone. In general, an intelligent device in the UBE is distinguished by an IP address, and it contains an embedded micro-controller with memory. These devices can make a decision following the input data.
(3) Read, write, and execute are the basic access types in the traditional access control. In the UBE, there are various access types such as on/off, touch, push, connect-to-internet, and so on. The access type is similar to the functions of intelligent devices.
(4) As described above, a user's authority is changed dynamically by context information such as the location, time, computing resources, production plan, and so on. A context agent gathers the context information from the sensing devices in the UBE.
(5) Space (location) is an important context of access control. A user can move from one location to another. The user's authority is restricted by his/her location. For example, a user John can use the milling machine in his workplace; however, he may not have permission to use the milling machines in other workplaces.

www.manaraa.com

There exist some research works that are related to the access control in the UBE. Zhang and Parashar (2004) have proposed a context-based access control model for pervasive applications through the dynamic role-based access control (DRBAC) model. DRBAC has the following two characteristics of dynamic access control: (1) A user's access privileges can be changed when the user's context changes and (2) a resource can adjust its access permission when its information changes. In spite of these characteristics are suitable for UBE, DRBAC does not consider space as an independent component and the context is roughly merged into the RBAC model.

Sampemane et al. (2002) have introduced an access control model for an active space. Active spaces are physical spaces augmented with heterogeneous computing and communication devices along with supporting software infrastructure. An active space can be configured for different types of applications at different times. The active space model comprises an access control system that automates the creation and enforcement of access control policies for different configurations of an active space. The model also defines three types of user roles—system roles, space roles, and application roles. System roles are assigned when user accounts are created, and they define users' generic permissions for certain classes of objects (or resources) within the entire system. Each active space has its own space roles, and if a user enters an active space, his/her system roles are automatically mapped to space roles. The active space model is a dynamic access control model; however, it does not clearly describe the mapping of system roles to space roles.

Wedde and Lischka (2004) have proposed the RBAC model in ambient and remote space. This model focuses on presenting a distributed and location-dependent RBAC approach that is multi-layered. It assumes that the objects of access control are stored in distributed system storage.

Park and Sandhu have proposed the concept of usage control (UCON). UCON encompasses traditional access control, trust management, and digital rights management, and it extends beyond these features in its definition and scope (Park and Sandhu 2002). It comprises six components—subject, objects, rights, authorization rules, conditions, and obligations. The concept of subjects, objects, and rights is similar to those in access control. Authorization rules are a set of requirements that should be satisfied before allowing subjects' access to objects or use of objects. Conditions are a set of decision factors that the system should verify during the authorization process along with the authorization rules before allowing the usage of rights for an object. Obligations are mandatory requirements that a subject has to satisfy after obtaining or exercising rights on an object. UCON is more advantageous than the traditional access control model in the UBE. Wang et al. (2006) have applied the UCON to the UBE. They borrowed the features of access control from the UCON and applied it to the special space of the UBE. They also introduced space objects with associated space rights, which are permissions for resources within the space. A space object is similar to space role of the active space model. The access control policies for the space are described by space objects and rights.

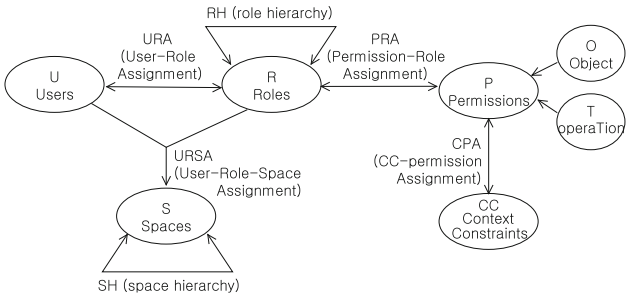

Fig. 3 Ubi-RBAC model

Despite the proposal of several access control models, the issues of users' mobility and context awareness of the UBE have not been addressed properly. We present a new access control model to overcome these problems.

Ubi-RBAC model

Formal description of Ubi-RBAC

In this section, we provide the formal description of the Ubi-RBAC model. Figure 3 shows the basic components of the Ubi-RBAC model. It is based on the RBAC model and follows the same definitions of user (U), role (R), role hierarchy, role–role assignment (RH), and permission (P). In addition, the Ubi-RBAC model has the following new components: space (S), space hierarchy (SH), and context constraint (CC).

Definition 1 (*space*) A space S is a unit location where user's access occurs. Examples of such a space are a desk, office, seminar room, workplace, and company. A nonobjective location such as "production department" can also be a space.

Definition 2 (*space hierarchy*) SH is a partially ordered space hierarchy on spaces subject to $SH \subseteq S \times S$. Figure 4 shows an example of a space hierarchy. In a space hierarchy, a higher space is a location in a lower space.

Definition 3 (*context constraint*) A context constraint CC is a special condition that decides whether a permission is allowed or disallowed. For example, the permission (*milling_machine, OPERATE*) is allowed if some materials are on the milling machine. The permission (*door-of-room216, OPEN*) is disallowed after office hours.

www.manaraa.com

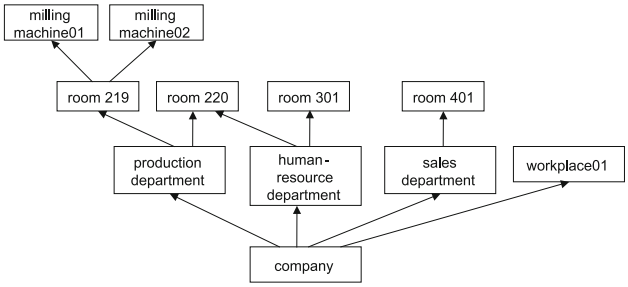

Fig. 4 Example of space hierarchy

Ubi-RBAC also follows the definitions of user–role assignment (*URA*) and permission–role assignment (*PRA*) of the RBAC model. Ubi-RBAC defines new relations such as user–role–space assignment (*URSA*) and context-constraint–permission assignment (*CPA*).

Definition 4 (*URSA: user–role–space assignment*) *URSA* is a ternary relation (*user, role, space*). A *URSA* tuple $(u1, r1, s1)$ should satisfy the condition that $(u1, r1)$ is an element of the user–role assignment information.

Definition 5 (*CPA: context-constraint–permission assignment*) *CPA* is a ternary relation (*context constraint, permission, allowance*) where the allowance is "true" or "false." For example, a *CPA* tuple $(c1, p1, true)$ implies that the permission $p1$ is allowed under the context constraint $c1$, whereas the tuple $(c1, p1, false)$ implies that $p1$ is disallowed under $c1$.

We summarize the complete description of the Ubi-RBAC model as follows:

$U = \{u \mid u \text{ is a user}\}$
$P = \{p \mid p \text{ is a permission}\}$
$R = \{r \mid r \text{ is a role}\}$
$S = \{s \mid s \text{ is a space}\}$
$CC = \{cc \mid cc \text{ is a context constraint}\}$

$URA \subseteq U \times R$, user–role assignment
$PRA \subseteq P \times R$, permission–role assignment
$CPA \subseteq CC \times P$, context-constraint–permission assignment
$URSA \subseteq U \times R \times S$, user–role–space assignment

$RH \subseteq R \times R$, partially ordered role hierarchy
$SH \subseteq S \times S$, partially ordered space hierarchy

Space hierarchy in Ubi-RBAC

Space hierarchy is one of special features of the Ubi-RBAC model. It reflects the hierarchy of the authority in the real world. Let us assume that *URSA* relations exist on the space hierarchy shown in Fig. 4, as follows:

(*TOM, CLERK, company*)
(*TOM, PRODUCTION_DEPT, production_department*)
(*TOM, MILLING_WORKER, milling_machine01*).

If *TOM* enters the space *production_department*, he can perform some tasks using the authority assigned to the *PRODUCTION_DEPT* role. If *TOM* enters the space *room219* or *room220*, he can assume the role of *PRODUCTION_DEPT* because he can inherit the default role from *production_department*. If *TOM* enters the space, *milling_machine01*, he can assume the role of *MILLING_WORKER*, whereas in the space *milling_machine02*, he can assume the role of *PRODUCTION_DEPT*. If *TOM* enters the space *room401*, he can assume the role of *CLERK*, which is inherited from the *company* through *sales_department*.

Property 1 *If a user U1 moves to space S1, he/she can assume the default role assigned to U1 in S1. If U1 has no default role in S1, the user inherits the default role from the nearest child space of S1.*

Space hierarchy should be consistent with role hierarchy. When a security administrator group assigns the default role to (*user, space*), users should follow Integrity principle 1.

Integrity principle 1 Assume that the symbol ">" expresses the partial order of role hierarchy and space hierarchy, and $A > B$ implies that role A is an ancestor of role B in a role hierarchy or space hierarchy. If a user has default roles A and B in the *space hierarchy* and $A > B$, then $A > B$ on the *role hierarchy* as well.

Access control principle in Ubi-RBAC

The access control principle is the main rule of an access controller. An access controller is a software module that makes decisions regarding users' access requests. It allows or denies the access request on the basis of authorization data. Therefore, the core of the access controller is a decision-making algorithm. The input of the algorithm is the access request "(*user_id, device, operation, space*)." The output of the algorithm is "True (allow access)" or "False (deny access)." Now, we describe the decision-making algorithm of the Ubi-RBAC controller. We use the notations described in Sect. "Formal description of Ubi-RBAC".

Let us suppose that

$$request\,(user_i, object\_id_i, operation_i, space_i)$$

is an access request of user $user_i$; then, the decision-making algorithm is as follows:

www.manaraa.com

```
/* check a user */
If (user_i ∉ U)
  return False;                    // deny access
/* check a permission */
If ((object_id_i, action_i) is valid permission) {
  take the permission id of (object_id_j, action_i) from
  P into perm_k;
  take all context-constraints of perm_k into context-
  constraints;
} else
  return False;                    // deny access
/* check context constraints */
while (CC! = φ) {
    take the next context-constraint cc_m from CC;
    If (cc_m is False)
  return False;                    // deny access
}
/* check access rights of a user */
take the activated role for user_i from URA into
  set role_n;
take the child roles of role_n from RH into set
  assigned_roles;
add role_n into assigned_roles;
while (assigned_roles! = φ) {
    take the next role role_n from assigned_roles;
    If (((role_n, perm_i) ∈ PRA) & ((user_i, role_n,
    space_i) ∈ URSA))
  return True;                     // allow access
}
  return False;                    // deny access
```

Note: ϕ means empty set

Discussion

The proposed Ubi-RBAC model intends to support the context awareness and mobility of subjects (human users) in the UBE. In the Ubi-RBAC environment, human users' authority depends on their assigned role, and their role is dynamically changed when they navigate among spaces. A user's default role for a specific space is defined by URSA information. If a user has no default role for a space, he/she inherits the default role from the nearest child space. A user's pre-assigned permission can be limited by context constraints. When a user requests access to a device, the access controller may verify both his/her permission and context information. A valid permission can be invalid depending upon the time, number of people, status of device, and so on.

Ubi-RBAC does not adopt the concept of "session" from RBAC because there is no explicit login process in the UBE. When a user enters a space, the sensor system recognizes him/her and the implicit session is started automatically.

The method for implementing the context constraint is a difficult issue, and there are several methods to achieve it. For example, context information

- number of people over x (x: variable number)
- end of working hours
- room 215 has no meeting schedule

has various forms and sometimes includes variables. Therefore, it is difficult to develop a unique data model for the context information. It is a topic for further research and includes the design of an access controller (reference monitor).

Table 1 shows a comparison of the Ubi-RBAC model with other models. As observed from this table, the Ubi-RBAC model overcomes the limitations of the other models for the UBE.

Active Space is most similar with Ubi-RBAC, and we discuss more about the two models. Active Space suggests three kinds of user roles: system roles, space roles, and application roles that Ubi-RBAC does not support. System roles are assigned when user account are created, and define users permissions. Space roles express access control policies in specific space. Application roles allow an application to specify a customizable access control policies. Active Space requires *space administrator* for each space who manages the mapping between system and application roles and space roles. In the IMS environment, there are many spaces and not many security workers. Therefore, Active space is not suitable for IMS environment. Furthermore the consistency of security policies between spaces is very difficult because of many *space administrators.* In the Ubi-RBAC, security policies including roles and spaces are managed by centralized security worker or group. One of weak point of Active Space is that it does not support space hierarchy. It is very useful concept for security management such like role hierarchy. Spaces in IMS environment are composed of workplaces and offices. In many cases workplaces has a chain or hierar-

Table 1 Comparison of Ubi-RBAC model with other models

	RBAC	DRBAC	Active space	Wedde and Lischka	Ubi-RBAC
Supports *role*	○	○	○	○	○
Supports *space*	×	×	○	○	○
Supports *space hierarchy*	×	×	×	○	○
Pre-assigned authority can be restricted by context-information	×	△	△	×	○
Users' role is dynamically changed by their location	×	×	○	×	○

www.manaraa.com

chical structure following manufacturing processes. Therefore, space hierarchy is very useful for IMS environment. Ubi-RBAC supports simple security policies management through space hierarchy.

Conclusion

The UBE is a new challenge for access control areas. It has new requirements for access control, which are different from traditional information. In this study, we have proposed a new access control model termed the Ubi-RBAC model; it is based on the RBAC model, and it adds new components such as space, space hierarchy, and context constraints. The purpose of the Ubi-RBAC model is to achieve the context awareness and mobility of subjects (human users) in the UBE. When a user enters a space, his/her role is dynamically changed, and a pre-assigned authority (permission) can be restricted by context constraints. The method for developing a model and the implementation of context constraints are future research topics.

Acknowledgments This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2007-521-D00470).

References

Ferraio, D., Cugini, J., & Kuhn, R. (1995). Role-based access control (RBAC): Features and motivations. *Proceedings of the 11th Annual Computer Security Application Conference*. Anaheim, California, USA.

Gavrila, S. I., & Barkley, J. F. (1998). Formal specification for role based access control user/role and role/role relationship management. *Proceedings of the 3rd ACM workshop on Role-Based Access Control*. Fairfax, Virginia, USA.

IMS. http://www.ims.org. Accessed April 20, 2008.

Oh, S., & Park, J. (2004). Requirement analysis for access control model on ubiquitous computing environment. *Journal of Korea Information Processing Society, 11-A*(7), 563–570.

Park, J., & Sandhu, R. (2002). Towards usage control models: Beyond traditional access control. *Proceedings of 7th ACM Symposium on Access Control Models and Technologies (SACMAT 2002)*. Monterey, California, USA, pp. 57–64.

Sampemane, G., Naldrug, P., & Cambell, R. (2002). Access control for active space. *Proceedings of the 18th Annual Computer Security Application Conference*. Washington, DC, USA, pp. 343–352.

Sandhu, R. (1995). Rationale for the RBAC96 family of access control models. *Proceedings of ACM Workshop on Role-Based Access Control*. Gaithersburg, Maryland, USA.

Sandhu, R., Bhamidipati, V., & Munawer, Q. (1999). The ARBAC97 model for role-based administration of roles. *ACM Transactions on Information and System Security, 1*, 2 (TISSEC).

Sandhu, R., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control method. *IEEE Computer, 1.29*, 38–47.

Wang, H., Zhang, Y., & Cao, J. (2006). Ubiquitous computing environments and its usage access control. *Proceedings of the First International Conference on Scalable Information Systems*. HongKong, China.

Wedde, H.F., & Lischka, M. (2004). Role-based access control in ambient and remote space. *Proceedings of 9th ACM Symposium on Access Control Models and Technologies (SACMAT 2004)*. Yorktown Heights, New York, USA, pp. 21–30.

Weiser, M. (1991). The computer for the 21st century. *Scientific American, 265*, 94–104.

Zhang, G., & Parashar, M. (2004). Context-aware dynamic access control for pervasive applications. *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference* (pp. 21–30).

www.manaraa.com

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

www.manaraa.com