Journal of Computer Science 7 (5): 629-637, 2011
ISSN 1549-3636
© 2011 Science Publications

Situation, Team and Role based Access Control

Kyoji Kawagoe and Keisuke Kasai
Department of Information and Communication Science
College of Information Science and Engineering,
Ritsumeikan University, Kusatsu-City, Japan

Abstract: Problem statement: An emergency system of sharing and exchanging user's personal information is demanded in medical treatment and disaster situations. **Approach:** In such a system, personal information access control depending on user situations is greatly necessary. However, it is complicated to manage personal information access control directly, because the existing access control methods only support regular access control, not for an emergency case. **Results:** In this study, we propose a new access control model, called STRAC, which stands for Situation, Team and Role based Access Control. The STRAC enables access control of user personal information with consideration of context changes. **Conclusion/Recommendations:** In our proposed model, a concept of situations is introduced. Moreover, the proposed model is based on a concept of TMAC, which is an extension of a conventional RBAC model.

Key words: Access control, personal information, permission rights, crucial emergency, Emergency Operation Unit (EOU), object context, RBAC and TMAC models

INTRODUCTION

Personal information has been getting more important due to introductions of information systems using personal information such as location data from GPS. Examples of such information systems include patient information management in hospital information systems and information delivery in disaster information systems. For example, a medical doctor needs to view electronic medical records of an emergency patient on his or her diagnosis. However, it is necessary to manage the personal information strictly, so as not to permit it to be viewed by other doctors. After emergency medical treatment, the doctor's permission privileges should be deleted and another doctor receives new permission privileges on this patient when the next medical treatment on him occurs. The management is more complex in disaster information systems because it is important to open some kinds of personal information in some urgent situation but the target user should be limited. Therefore, it is important that such emergency systems should control personal information access privileges in an adaptable and flexible way.

Many access control methods have been proposed so far. The most basic access control method is RBAC (Feiner *et al*., 1995), which stands for Role Based Access Control. In the RBAC, based on roles, a role is defined as an abstract set of access privileges. Concretely, permission privileges are assigned to roles and roles are assigned to users. Therefore, it is unnecessary to assign a set of permissions to an individual user as well as to assign a particular permission to many users. Because of simplicity of RBAC method, RBAC has been used in many fields such as information management, resource management and access management. RBAC is currently a standard of ANSI/INCITS (American National Standard 359-2004; Ferraiolo *et al*., 2001; Sandhu *et al*., 2000). There have been many successors based on RBAC (Thomas, 1997; Joshi *et al*., 2005; Kulkarni and Tripathi, 2008; Bertino *et al*., 2005; Bertino *et al*., 2001; Covington *et al*., 2000; Park *et al*., 2006; Moyer and Ahamad, 2001; Motta and Furuie, 2001).

TMAC (Thomas, 1997), team based access control, was proposed to extend RBAC so as to introduce the concept of teams. A user is engaged in one or many groups. The content of his/her access control privilege needs to be changed depending on his/her group which he/she is supposed to be in. TMAC has several successors (Christos *et al*., 2001; Alotaiby and Chen 2004).

In this study, we introduce a concept of situations as an extension of the existing RBAC and TMAC in order to enable systems to handle a dynamic change of permission privileges to access to personal information more easily.

Corresponding Author: Kyoji Kawagoe, College of Information Science and Engineering, Ritsumeikan University Japan

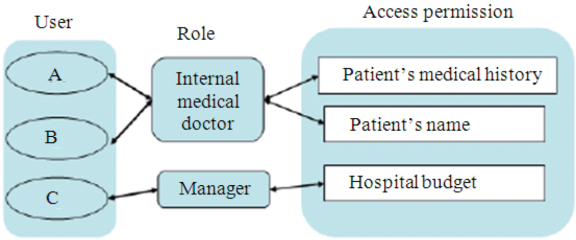


Fig. 1: Example of RBAC

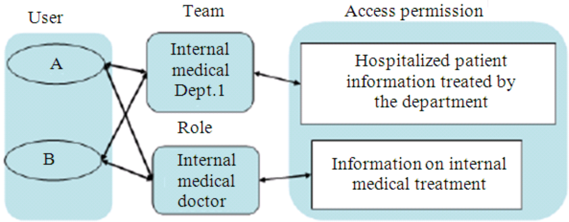

User
Team
Access permission
A
B
Internal medical Dept.1
Role
Internal medical doctor
Hospitalized patient information treated by the department
Information on internal medical treatment

Fig. 2: Example of TMAC

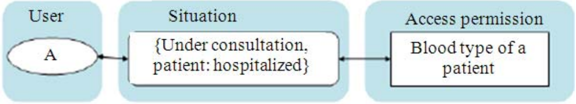


Fig. 3: Example of situation

It is important to manage access control of personal information dynamically when some crucial emergency event, such as urgent medical treatment and disaster relief, happens. We propose a new model, called STRAC, Situation/Team/Role based Access Control, of personal information access control for supporting dynamic permission privilege changes which is difficult to manage in not only RBAC but also TMAC.

The study is structured as follows: The next section discusses the concept of situations in our model to solve the problem in RBAC and TMAC, followed by its formal definitions. Then, we describe a prototype framework of our proposed situation based access control model with introduction of the situation, based on both RBAC and TMAC. After the related work and comparison with our STRAC are described, we conclude our study.

MATERIALS AND METHODS

RBAC and TMAC models: Throughout the study, we focus on an application of Hospital Information System, in which patients, medical doctors, nurses and medical technologists are related. In Fig. 1, a simple example of RBAC model is described. As shown in the figure, a role is an abstract of several access permission privileges. For example, Internal Medicine Doctor is a role and is assigned to both a user A and a user B. With this assignment, both users can access to the access permissions assign to this role, such as Patient's medical history and Patient's name in this example.

Figure 2 shows an example of TMAC model description in a schematic way. A user A is a member of some team, such as Internal medicine department I. The user is also a member of another team, Internal Medicine Doctor Research. For each membership, personal information which the user can access to is different. For example, when he does as a member of the first team, then he can access to the hospitalized patient information treated by the department.

As these examples show, a system can control accessibility of users with both RBAC and TMAC efficiently, rather than the novice access control with neither roles nor teams. However, suppose that we have a case where some unexpected event occurs suddenly, such as when some emergency medical treatment is necessary due to a traffic accident, or when some urgent survivor checking is needed after a strong earthquake. In such a case, each user is dynamically assigned to appropriate access permission privileges for individual personal information with either RBAC or TMAC. However, such a control is difficult to do because a system or its manager needs to manipulate all the necessary temporal assignment in a short time. Otherwise all the information is open to all the users.

Situation based access control: In Fig. 3, a concept of situations is shown. In order to realize flexible access control management in emergency cases, a concept of situations is introduced in the study. A situation is an abstract of conditions which is composed of user contexts and related objects contexts. For example, in Fig. 3, a user A can obtain access permission privileges on the blood type of a patient of user A if the user A is under consultation and the patient is also hospitalized. The first condition is related to user contexts and the second condition is related to object contexts. Here any object, which is defined in an application field differently, can be considered so far. The point on this situation concept lies in the composition of two kinds of context information: user contexts and object contexts.

Table 1: operation types

Component	operation types	Remarks
User	Create/Delete/Update	These operations can be done on registration, profile change and quitting of a user
Role	Create/Delete/Update	A role can be created, deleted and updated on request.
Team	Create/Delete/Update	A team can be created, deleted and updated on request.
Permission	Create/Delete/Update	A permission on personal information can be manipulated with these operations
User contexts	Create/Delete/Update	A user context, a type of user’s condition can be manipulated.
Object contexts	Create/Delete/Update	An object context, a type of a condition on an object can be manipulated.
Situation	Create/Delete/Update	A situation, a pair of the above two types of contexts can be manipulated.
Session	Define/ Detect Permission Privilege/ Obtain Personal Information	A session is defined beforehand, permission privileges are detected for a session and finally personal information can be obtained and returned finally.

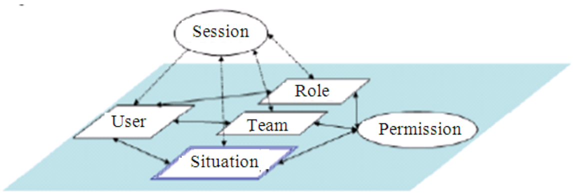


Fig. 4: Basic structure of the proposed method

With the introduction of the situation concept, the basic structure of our proposed model is shown in Fig. 4. As this figure shows, the situation concept is introduced in the same level as the role concept of RBAC and also as the team concept of TMAC. A user can be assigned to roles, teams as well as situations. Depending on the assigned role, team and situation, access permissions can be assigned to the user as a result for each session. We will view these assignments in more detail. When a patient is in a severe condition due to a sudden traffic accident, he/she is moved to a hospital where our proposed access control model is incorporated into a hospital information system. Then, he/she needs to be medically treated by some doctors and nurses in an Emergency Operation Unit (EOU). That is, a situation of {under treatment, (Patient: in the EOU)} is detected here and some appropriate and pre-defined access permission is assigned to the users. It goes without saying that both user’s team information and user’s role information are also taken into account for access permission assignment. After the urgent operation is finished and the patient is in a more stable state, hr/she will next be physically taken care of. He/she is hospitalized in a physical department of the hospital and made some physical treatment or operations for a while. From the physical doctor viewpoint, the situation is described as {under treatment, (Patient: Hospitalized in a physical department)}. Some access permissions for the doctor are assigned to the doctor for this situation description. After physical treatment, finally the patient is moved to the internal medical consultation and necessary personal information can be accessed by a limited number of users by our model based assignment as it is before.

It should be noted from the above example that it would be more difficult and tedious to manage the change of access permission, caused either by users context change or by object context change. However, by predefining user contexts and object contexts and by detecting an appropriate situation on time, permission control can be automatically done with the proposed model.

STRAC operations: In our proposed model, there are several operation types on our components. Some of the important operation types are described below in the Table 1.

Formal definition: The proposed model has the following components:

- U stands for users and shows a domain of users
- R stands for roles and shows a domain of roles
- T stands for teams and shows a domain of teams
- P stands for Permissions and shows a domain of permissions
- Se stands for Sessions and shows a domain of sessions
- UC stands for user contexts and shows a domain of user contexts
- OC stands for object contexts and shows a domain of object contexts
- Si stands for situations and shows a domain of sessions. A situation is a pair of a user context and an object context. That is $Si \subset UC \times OC$

In the method, there are three components added to the original RBAC and TMAC. The first two components, UC and OC are introduced to define a situation, as well as the last component, “situation”, is added to the original TMAC and RBAC. It is because our model is an extension of both RBAC and TMAC models, so as to include dynamic change in access control.

With these components, several assignment relations are defined as follows:

- URA $\subseteq U \times R$, is a many-to-many user-to-role assignment relation
- TUA $\subseteq T \times U$, is a many-to-many team-to-user assignment relation
- PRA $\subseteq P\times R$, is a many-to-many permission-to-role assignment relation
- TPA$\subseteq T \times P$, is a many-to-many team-to-permission assignment relation
- SUA $\subseteq Si \times U$, is a many-to-many situation-to-user assignment relation
- SPA $\subseteq Si \times P$, is a many-to-many situation-to permission assignment relation
- UUCA $\subseteq U \times UC$, is a many-to-many user-to-user context assignment relation
- POCA$\subseteq P \times OC$, is a many-to-many permission- to object context assignment relation

As described before, the proposed method is based on both RBAC and TMAC. Therefore the differences of the definition of the above are additions of the situation related assignment relations. That is, SUA, SPA, UUCA and POCA are added to the original TMAC and RBAC.

As for functions, the proposed method has the following functions:

- Session-user: Se $\rightarrow$U is a function mapping each session s $\in$ Se to the single user. That is, user(s) is the user of the session and is the constant during the session
- Session-role: $S_e \rightarrow 2^R$ is a function mapping each session s $\in$ Se to a set of roles. That is, Session-role(s) $\subseteq \{r \mid (user(s), r) \in URA\}$
- Session-team: Se $\rightarrow 2^T$ is a function mapping each session s $\in$ Se to a set of teams. That is, Session-team(s) $\subseteq \{t \mid (t, user(s)) \in TUA\}$
- Team-user: T $\rightarrow 2^U$ is a function mapping each team t$\in$T to a set of users. That is, Team-user(t) $\subseteq \{u \mid (t, user(s)) \in TUA \wedge \exists s$ such that $user(s)=u\}$
- Session-situation: Se $\rightarrow 2^{Si}$ is a function mapping each session s$\in$Se to a set of situations. That is, Session-situation (s) $\subseteq\{s_i \mid (s_i , user (s)) \in SUA$
- User-UserContext: U $\rightarrow 2^{UC}$ is a function mapping each user u$\in$U to a set of object contexts. That is, User-UserContext(u) $\subseteq\{uc \mid (u,uc)\in UUCA\}$
- Permission-ObjectContext: P $\rightarrow 2^{OC}$ is a function mapping each user u $\in$U to a set of object contexts. That is, Permission-ObjectContext (p) $\subseteq$ {oc| (p,oc)$\in$POCA}

Among the first five functions, the Session-situation function is only added to the original TMAC and RBAC models. The Session-situation function is used to express a set of situations for a certain session. The last two functions are introduced to express the concrete description of each situation. As described before, a situation is a pair of an object context and a user context. Therefore, these functions are used to express the relationship the relationship between a user and a user context and the relationship between permission and an object context and, respectively.

Description example: Using the above definition of the proposed model, a description example is presented below in order to make model understanding easier.

Personal information: Patient (Name, Age, Blood-type).

Component definition:

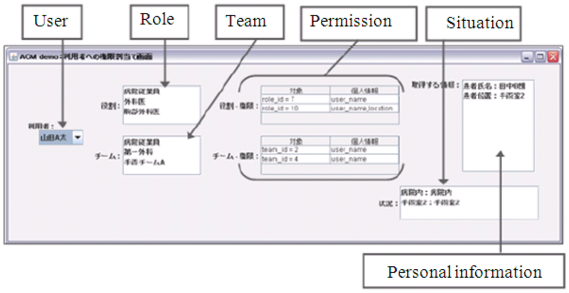
- U = {Taro, Hanako}
- R = {Surgeon, Nurse}
- T = {OperationTeam}
- P = {read-Name, read-Age, read-Bloodtype}
- Se={s1}
- UC = {working, operating}
- OC = {Patient:operating room, Patient: in hospital}
- Si = {(operating, Patient: operating room), (working, Patient: in hospital)}

Session assignment functions:

- user (s1) = {Taro}
- Session-role (s1) = {Surgeon}
- Session-team (s1) = {OperationTeam}
- Session-situation(s1)={(operating, Patient:operating room)}

Other Assignment:

- User-UserContext(Taro) ={operating)
- User-UserContext(Hanako)={operating}
- Permission-ObjectContext(read-Name)={operating room}
- Permission-ObjectContext(read-Age)={operating room}
- Permission-ObjectContext(read-Bloodtype) ={operating room}



ACM demo：利用者への権限割当て画面

利用者：山田A太

役割：病院従業員
外科医
胸部外科医

チーム：病院従業員
第一外科
手術チームA

役割・権限：

対象	個人情報
role_id = 7	user_name
role_id = 10	user_name,location

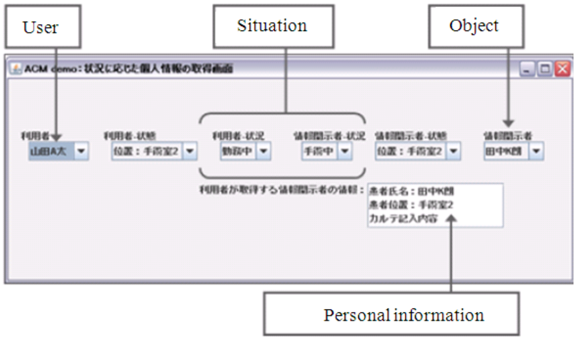
チーム・権限：

対象	個人情報
team_id = 2	user_name
team_id = 4	user_name

取得する情報：患者氏名：田中K郎
患者位置：手術室2

状況：病院内：病院内
手術室2：手術室2

Fig. 5: Sample of display on permission assignment function



ACM demo：状況に応じた個人情報の取得画面

利用者：山田A太
利用者-状態：位置：手術室2
利用者-状況：勤務中
情報開示者-状況：手術中
情報開示者-状態：位置：手術室2
情報開示者：田中K郎

利用者が取得する情報開示者の情報：患者氏名：田中K郎
患者位置：手術室2
カルテ記入内容

Fig. 6: Sample of display on situation assignment

- URA = {(Taro, Surgeon), (Hanako, Nurse)}
- TUA = {(OperationTeam,Taro), (OperationTeam, Hanako)}
- PRA = {(read-Bloodtype, Surgeon), (read-Name, Nurse), (read-Age, Nurse)}
- TPA={(OperationTeam,read-Name), (OperationTeam, read-Age)}
- SUA = {((operating, Patient: operating room), Taro), ((operating,Patient: operating room), Hanako)}
- SPA={((operating,Patient:operating room), read-Name),((operating, Patient: operating room), read-Age), ((operating, Patient: operating room), read-Bloodtype)}

Assignment process:

Step1: Detecting Role and Team: From Taro's role, Surgeon and Taro's team, Operation Team, each set of permission privileges are detected.

Step2: Detecting Situation: Taro's situation where Taro is on operating and patient is in operating room is detected.

Step3: Access Privilege Concatenation: The final access permission privileges are concatenated from each set of the privileges obtained in the Step1 and Step2. Then the concatenated access permission privileges are assigned to this user.

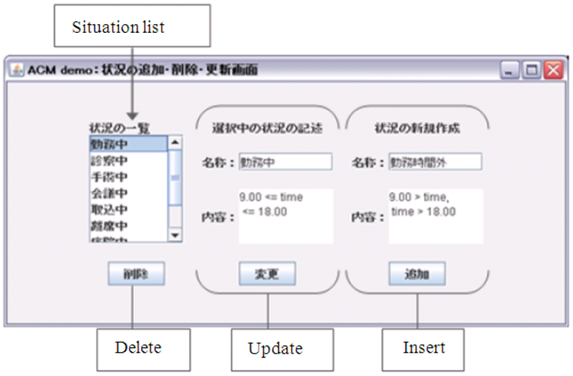
RESULTS

Realization of the proposed model: In this section, a framework, which we developed to show how the proposed model is used in a system, is presented. Due to space limitations, we only describe the three functions of the framework, which are permission assignment function, situation assignment function and situation management function.

Permission assignment function: In Fig. 5, the permission assignment function display is shown. The function is a function to assign some permission privileges to a user by selecting his/her current role, team and situation. A manager can select a user from the most left selection field and his/her role and team from the next two selection fields of roles and teams. His/her current situation is output on the privilege lower text field. The final access permission privileges are displayed in the permission output area, which is left of the related personal information display.

Situation assignment function: Figure 6 shows the situation assignment function of the framework. This function especially focuses on permission privilege assignment from the situation. As in Fig. 5, a manager first selects a user to assign permissions. On the other hand, an object, which is a patient in this case, can be selected on the most privilege selection field. In the middle, possible situation conditions are displayed. After selecting the current his/her condition, user contexts and the current patient condition, object contexts, the permission privileges for the selected situation is searched and displayed in the middle.

Situation management function: Among the components of the proposed model, the situation is a new concept, which is added to the base RBAC and TMAC models. Therefore, we show how the situation can be managed. Figure 7 shows a sample display of the framework we developed for situation management function. On the left selections, some situations which have been registered are listed. By selecting one of the listed situations, the selected situation can be either updated or deleted. When a new situation is to be inserted, some situation information just needs to be input in the text fields on the privilege and click the insert button on the bottom of the fields.



ACM demo：状況の追加･削除･更新画面

状況の一覧

勤務中
診察中
手術中
会議中
取込中
離席中

選択中の状況の記述

名称：勤務中

内容：9.00 <= time <= 18.00

状況の新規作成

名称：勤務時間外

内容：9.00 > time, time > 18.00

Fig. 7: Sample of display on situation management

By using these functions of the framework, a system with the proposed model can control an appropriate personal information access privileges in a flexible way owing to our situation concept.

DISCUSSION

After the original RBAC model was proposed many years ago, lots of RBAC extensions have been proposed. The RBAC standard specification contains several representation levels, of Flat RBAC, Hierarchical RBAC, Constrained RBAC and Symmetric RBAC (American National Standard 359-2004; Ferraiolo *et al*., 2001). For these cumulative RBAC levels, various access control aspects can be defined. For example, Hierarchical RBAC enables to define a hierarchical role and assignment of conflict roles can be avoided with Separation-Of-Duty (SSD) in Constrained RBAC. Regarding the RBAC successors, the following models among them are briefly described below: GRBAC (Covington *et al*., 2000; Moyer and Ahamad, 2001), TRBAC (Bertino *et al*., 2001), GTRBAC (Joshi *et al*., 2005), Geo-RBAC (Bertino *et al*., 2005), CA-RBAC (Kulkarni and Tripathi, 2008) and CRBAC (Park *et al*., 2006).

GRBAC (Joshi, *et al*., 2005; Moyer and Ahamad, 2001) is a generalization of the original RBAC by incorporating the concepts of object roles and environment roles in order to apply RBAC for the-home-of–the-future application. The object role is used to capture many commonalities among objects in a system and to classify the objects, while an environment role is used to specify system state. An example of an environment role is a role corresponding to the first Monday of each month.

There are several models to extend the RBAC to include temporal aspects. Some typical models include TRBAC (Bertino *et al*., 2001) and GTRBAC (Joshi *et al*., 2005). In TRBAC, Temporal Role-Based Access Control model, temporal constraints on enabling or disabling roles are supported by incorporating periodic enabling or disabling of roles, individual exceptions and temporal dependency specifications. GTRBAC, Generalized Temporal Role-Based Access Control, was mainly introduced in order to specify temporal constraints for the user-role and role permission assignments and constraint enabling or disabling which cannot be specified with TRBAC. GTRBAC can handle the constraint by introducing new several types of temporal constraints, such as activation constraints, runtime events and constraint enabling expressions.

For extending RBAC to deal with spatial and location information, Geo-RBAC was introduced (Bertino *et al*., 2005). In order to do so, Geo-RBAC introduced special entities to model spatial objects, user positions and geographically bounded roles. In Geo-RBAC, a concept of spatial role was also introduced.

There are many context-aware applications with advent of various sensing devices and smart mobile equipment. CA-RBAC is a context aware extension of RBAC. In order to handle context awareness with CA-RBAC (Kulkarni and Tripathi, 2008), CA-RBAC manages the access control layer separately from the context aware management layer. On the context management layer, context agents check and monitor the object locations and conditions. They also send a query or an event to access control layer objects. On the layer, appropriate permission privileges can be performed by dynamic object bindings. On the other hands, C-RBAC, context-role based access control model, supports a new concept of context role (Park *et al*., 2006). A context role is used to capture security relevant context information about the environment for use in C-RBAC policies. Examples of context roles include location-related context and time-related context. Because a context role is an abstract concept, any context is possible to can be captured, such as location, speed, temperature, light level and so on. Although these RBAC successors described above have been proposed, no existing models can handle a situation in emergency events.

As described before, with the introduction of TMAC (Thomas, 1997), Team based Access Control, modeling of teams as well as of roles can be realized. C-TMAC (Christos, 2001) and TMAC2004 (Alotaiby and Chen, 2004) are some extensions of the TMAC. C-TMAC, Context based Team Access Control, consists of users, roles, permissions, teams and contexts, as well as a set of sessions.

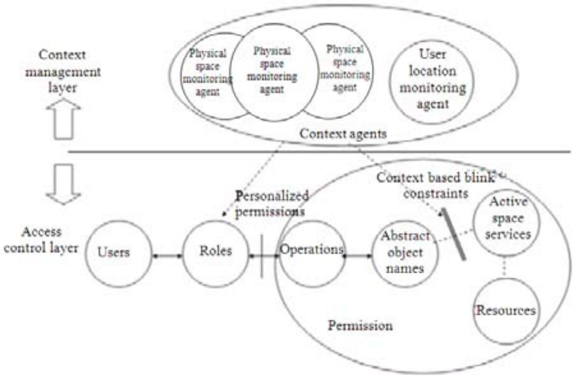


Fig. 8: CA-RBAC (Kulkarni and Tripathi, 2008)

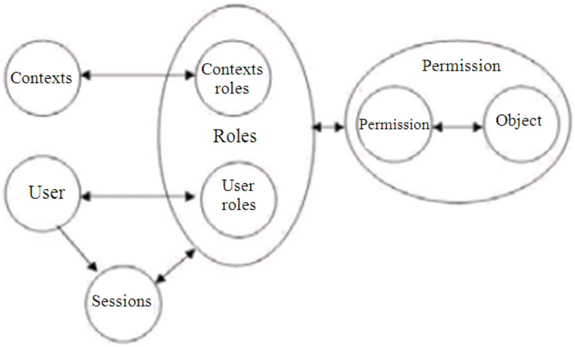


Fig. 9: C-RBAC (Kulkarni and Tripathi, 2008)

The main difference of C-TMAC from TMAC is the introduction of contexts. A context is assigned to teams in C-TMAC. TMAC2004 is also an extension of TMAC. In TMAC2004, team instances are introduced in order to capture an internal team work in a team. While a team is associated with roles, only the team instance is assigned to contexts and used for activating permissions. Because teams and roles can be use cumulatively, Team and role based access control should be appropriate extension to RBAC. However, in order to handle context information, both C-TMAC and TMAC2004 are not enough due to limited abilities in specifying contexts.

Recently, a situation based access control model called SitBAC has been proposed (Pelega *et al*., 2008). SitBAC was mainly developed for patient data access control, similar to one of our applications. SitBAC is a conceptual model with use of Object Process Methodology, in order to define scenarios of situations where patient data access is permission or denied. It is unclear that SitBAC can be used together with the popular RBAC model, because SitBAC is more general than RBAC.

In the rest of this section, some related work and comparison with our method are described.

CA-RBAC (Kulkarni and Tripathi, 2008): CA-RBAC is composed of two layers: Context Management Layer and Access Control Layer, shown in Fig. 8. On the Access Control Layer, the components of the conventional RBAC models are defined in addition to the mechanism for personalized permissions. On the Context Management Layer, Context agents, including some physical space monitoring agents and user location monitoring agents, are described, which are used for monitoring sensor data and detecting user locations. The CA-RBAC also contains two features: Context based binding and personalized role permissions. These are used for access control in context awareness services. The context based binding is to bind a context- based permission to a service provided within some region under some context condition. The context agents on the context management layer bind an object to the underlying service when some user is in a location. A personalized role permission is described as a special operation on an object defined in a role. For role-to-operation assignment the operation is independently assigned to the role like an instance of the operation. Therefore, a user assigned to the role can access to an instance of the special operation only when the user has the role detected by the context agents. The CA-RBAC model is more complex and concrete because the model can be realized with the existing OMG framework rather than RBAC model. The separation of the context management function from the access control has several advantages. However, there may be some argument on the interface between the two layers. The dependency of the interface may cause the complicated design task.

In our STRAC model, a user context and an object context are used to handle context awareness. The details of the context description are hided inside these context definitions. Therefore, our model is viewed to be simpler and easier to understand and to implement.

C-RBAC (Park *et al.*, 2006): C-RBAC was proposed as an extension of the RBAC by inclusion of context roles, as shown in Fig. 9. A context role is a special type of roles, representing context information about the related environment. A context role can be assigned to a context which represents some context information in the system such as time, location and temperature. In order to activate the context-to-context role assignment, a C-RBAC transaction specifies the access control with a tuple in the form of <user-role, context-role, permission>.

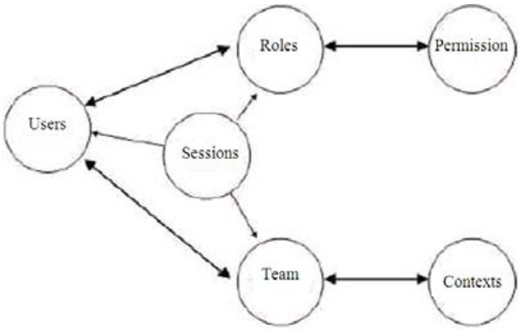


Fig. 10: C-TMAC (Georgiadis *et al*., 2001)

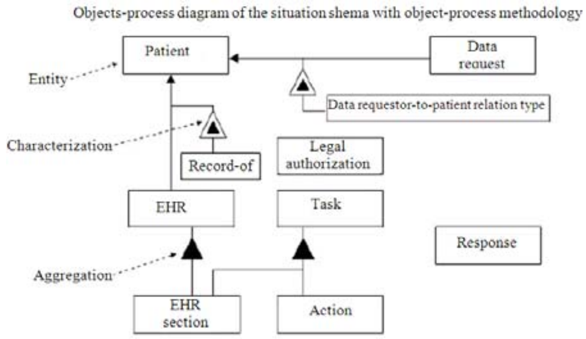

Objects-process diagram of the situation shema with object-process methodology
Patient
Data request
Entity
Data requestor-to-patient relation type
Characterization
Record-of
Legal authorization
EHR
Task
Response
Aggregation
EHR section
Action

Fig. 11: SitBAC (Pelega *et al*., 2008)

In the CRBAC, the only extension on the existing RBAC model for context aware applications is to introduce the context role as a special type of roles. The RBAC role concept, called User roles in C-RBAC, is originally a set of permissions. Because the context role is not a set of permissions, it may be natural that the context role is defined as the different type than the role.

In our STRAC model, the situation is not the special type of roles because the situation is not a set of permissions, as in the same way as the team of TMAC. In order to activate the assignment of user-to-role and situation-to-user, a session description can be used in our model.

C-TMAC (Christos *et al*., 2001): C-TMAC is an extension of TMAC for adapting it to context aware applications. The point of C-TMAC is introduction of the context to which a team can be assigned, as shown in Fig. 10. In C-TMAC, the context represents information on the required data objects for a specific activity, including locations and time intervals. Therefore, the team is used to enable a user to obtain a context as an intermediary between users and contexts.

In the original TMAC, the team concept is used for representing a group of users rather than the context intermediary. In the C-TMAC, it is unclear whether a team as its sense in the original TMAC can be described, representing a group of users independently from context information.

In our STRAC model, the original team concept is used as it is in the TMAC model. In addition to roles and teams, a situation can be described in order to assign a user to a context as the same meaning in C-TMAC as an intermediary between users and context information.

SitBAC (Pelega *et al*., 2008): SitBAC is an access control model, not based on the RBAC model and based on Object-Process methodology, in order to define scenarios for patient data access control, shown in Fig. 11. In the SitBAC model, the situation schema is proposed after analyzing the domain information and studying access requirement on electronic health records in a qualitative way. It goes without saying that the obtained SitBAC is far different from either RBAC or TMAC, because the SitBAC model is a result of information access control modeling for the specific application domain. As a result, some entity types like Patient and EHR, are defined as special types of entities used for situation schema description. The SitBAC is proposed from a complete different way of access control modeling as the other RBAC related models including our STRAC model. Therefore, any comparison between SitBAC and other RBAC models may be meaningless except a comparison only on the approach itself.

CONLCUSION

In this study, we propose a new access control model especially for emergency systems, called STRAC, which stands for Situation, Team and Role based Access Control. In order to realize a flexible permission privilege management, which is required in emergency systems, the proposed model is extended from the existing models, RBAC and TMAC with introduction of the situation components. A situation, which is an abstract of conditions, is represented by a pair of user contexts and object contexts. With this introduction, an emergency system, based on this model, enables to easily change permission privileges with changes in user contexts as well as object contexts. We also describe in the study the basic concept and the framework of the proposed model. From a brief comparison with the base RBAC and TMAC, it is also shown that the proposed model is suitable for the emergency system application area.

The proposed model, STRAC can be easily extended in order to specify hierarchical structures of roles, teams and situations, as is the same way both RBAC and TMAC do. Future work includes STRAC model implementation combined with an emergency system such as a hospital information system.

ACKNOWLEDGMENT

This study is partially supported by MEXT, Japan.

REFERENCES

Alotaiby, F.T. and J.X. Chen, 2004. A model for team-based access control (TMAC2004). Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), Apr. 05-07, Las Vegas, Nevada, pp: 450-454. ISBN: 0-7695-2108-8

American National Standard, 359-2004. Information technology-role based access control.

Bertino, E., B. Catania, M.L. Damiani and P. Perlasca, 2005. GEORBAC: A Spatially Aware RBAC. Proceedings of the 10th ACM Symposium on Access Control Models and Technologies, ACM, New York, USA., pp: 29-37. DOI: 10.1145/1063979.1063985

Bertino, E., P.A. Bonatti and E. Ferrari, 2001. TRBAC: A temporal role based access control model. ACM Trans. Inform. Syst. Security, 4: 191-233. DOI: 10.1145/501978.501979

Covington, M.J., M.J. Moyer and M. Ahamad, 2000. Generalized role based access control for securing future applications. Proceeding of the National Information Systems Security Conference (NISSC).

Feiner, D.F., J.A. Cugini and D.R. Kuhn, 1995. Role-Based Access Control (RBAC): Features and motivations. Proceedings of the Annual Security Applications Conference, IEEE Computer Society Press.

Ferraiolo, D.F., R. Sandhu, S. Gavrila, D.R. Kuhn and R. Chandramouli, 2001. Proposed NIST standard for Role-based Access Control. ACM Trans. Inform. Syst. Security (TISSEC), 4: 224-274. DOI: 10.1145/501978.501980

Georgiadis, C.K., I. Mavridis, G. Pangalos and R.K. Thomas, 2001. Flexible team-based access control using contexts. Proceedings of the sixth ACM symposium on Access control models and technologies, ACM, New York, USA., pp: 21-27. DOI: 10.1145/373256.373259

Joshi, J.B.D., E. Bertino, U. Latif and A. Ghafoor, 2005. A generalized temporal role-based access control model. IEEE Trans. Knowl. Data Eng., 17. 4-23. DOI: 10.1109/TKDE.2005.1

Kawagoe, K. and K. Kasai, 2010. STRAC: Personal information access control for emergency systems. Proceedings of the 2010 6th World Congress on Services, July, 5-10, IEEE Computer Society Washington, DC, USA., pp: 280-286. DOI: 10.1109/SERVICES.2010.103

Kulkarni, D. and A. Tripathi, 2008. Context-aware role-based access control in pervasive computing systems. Proceedings of the 13th ACM Symposium on ACCESS Control Models and Technologies, June 15-17, ACM New York, USA., pp: 113-122. DOI: 10.1145/1377836.1377854

Motta, G. and S. Furuie, 2001. A contextual role-based access control authorization model for electronic patient record. IEEE Trans. Inform. Technol. Biomed., 7: 202-207. ISSN: 1089-7771

Moyer, M.J. and M. Ahamad, 2001. Generalized role-based access control. Proceeding of the 21st IEEE International Conference on Distributed Computing Systems, Apr. 16-19, Mesa, AZ., pp: 391-398. ISBN: 0-7695-1077-9

Park, S.-H., Y.-J. Han and T.-M. Chung, 2006. Context-role based access control for context-aware application. Lecture Notes Comput. Sci., 4208: 572-580. DOI: 10.1007/11847366_59

Pelega, M., D. Beimelb, D. Dorib and Y. Denekamp, 2008. Situation-based access control: Privacy management via modeling of patient data access scenarios. J. Biomed. Inform., 41: 1028-1040. DOI: 10.1016/J.JBI.2008.03.014

Sandhu, R., D.F. Ferraiolo and D.R. Kuhn, 2000. The NIST model for role based access control: Toward a Unified Standard. Proceeding of the 5th ACM Workshop on Role Based Access Control, July 26-28, ACM New York, USA., pp: 47-63. DOI: 10.1145/344287.344301

Thomas, R.K., 1997. Team-based Access Control (TMAC): A primitive for applying role-based access controls in collaborative environments. Proceedings of the 2nd ACM Workshop on Role based Access Control, Nov. 06-07, ACM New York, USA., pp: 13-19. DOI: 10.1145/266741.266748