Dynamic Collaborations for Information Sharing Within and Across Virtual Teams

Ahmad Kamran Malik, Schahram Dustdar
Distributed Systems Group, Vienna University of Technology, Austria
{kamran, dustdar}@infosys.tuwien.ac.at

Abstract

Dynamic teams created by different enterprises collaborate to achieve their mutual goals. User-based collaborations for a specific task, within and across teams, result in a new team represented as dynamic collaboration (c). These inter-team and intra-team collaborations are mostly temporary which are based on specific task or goal. Collaborations among different teams, within a team, or across teams can take different forms when some or all of the users belonging to one or more teams collaborate for a common goal having diverse access control and sharing requirements imposed by themselves, their teams, or their parent enterprises. In such a dynamic environment, information sharing and privacy are critical issues for users, their teams and enterprises. We propose a dynamic sharing and privacy-aware model that supports enhanced sharing and preserves the privacy of users, their teams, and enterprises. We extend the widely used Role-Based Access Control (RBAC) model with team and task entities in addition to sharing and privacy data elements. In this model, using context constraints and hybrid access control policy, a user can control what to share with whom in which context conditions. We present a Dynamic Sharing and Privacy-aware Role-Based Access Control (DySP-RBAC) model that extends the RBAC model for enhanced sharing and privacy of information among collaborating users within and across virtual teams.

I. Introduction

Collaborations are the heart of Collaborative Working Environments (CWE) [1] and are an area of active research. Dynamic collaborations in a CWE involving multiple entities, frequently changing requirements, and context-based conditions are considered in this paper. Enterprise-based virtual teams are distributed and dynamic entities which involve users from all over the world. Users can be experts of different fields which collaborate with each other in a virtual team and can be employees of different enterprises. Entities involved in this scenario are the users, roles, tasks, teams, and enterprises. Users, collaborating with each other, need to share their context information, for example, location, current task, devices, network, etc. Privacy of user's information being shared within or across different teams, tasks, and enterprises is a basic requirement. In presence of dynamic and temporary collaborations this need becomes essential. Need for dynamic collaboration can emerge within a team or across multiple teams, which result in inter-team and intra-team collaboration. These collaborations are target-oriented and their target is mostly limited to the completion of a task or part of a task. Within a team, dynamic collaborations can be considered as sub-team and across the teams they can be in the shape of union or intersection of teams. As dynamic collaborations are mostly created for a temporary purpose, for example, for the lifetime of a meeting which spans few hours or few days, the question is whether create a new team for them or just provide them with collaborative sharing privacy within their existing team.

We propose a Dynamic Sharing and Privacy-aware Role-Based Access Control (DySP-RBAC) model which is based on the Role-Based Access Control (RBAC) model [2]. Our DySP-RBAC model extends RBAC using team and task entities as well as sharing and privacy related data elements. Sharing elements include collaborative relationships among users and access levels. Privacy elements include purpose, obligations, and context conditions. Owner-defined context conditions are used to preserve the privacy of an owner's information in such a dynamic environment. Context constraints [3] are used to cope with the dynamic nature of environment. Using context constraints, dynamic policy adaptation can be performed at runtime. In this way, by extending the RBAC model, DySP-RBAC model becomes a hybrid policy model which contains hybrid permission assignments defined by the administrator as well as owner of the information. Moreover, collaboration and contact history is used in this system that helps evaluating access requests. Information being shared among collaborating users is arranged in a hierarchical order so that user can specify as much information she wants to share with others.

The remainder of the paper is organized as follows. Section 2 describes a motivation scenario. Section 3 describes our dynamic access control policy. Section 4 presents the Dynamic Sharing and Privacy-aware Role-Based Access Control (DySP-RBAC) model. Section 5 presents a discussion about the handling of final user permissions and dynamic collaborations in DySP-RBAC model. Section 6 describes background and related work. Finally, in Section 7, we draw our conclusions and describe our future work.

978-0-7695-4625-4/11 $26.00 © 2011 IEEE
DOI 10.1109/FIT.2011.45

II. MOTIVATING SCENARIO

The dynamic collaborative working environments, nowadays, use virtual teams of experts that are located in different parts of the world and collaborate with each other through communication technologies. In this way, many distributed individuals could become a part of more than one team at a time. Virtual teams are dynamic in nature and members can join and leave their team at any time. There are many types of collaboration among members of the teams. Within a team, there can be temporary sub-teams, which are created for handling a temporary situation/tasks. Our proposed scenario uses different types of temporary collaborations within and across teams that are described in Figure 1. Here *"C"* represents temporary or dynamic collaboration among users or teams. In Figure 1, the dynamic collaboration types (a) and (b) represent user-based collaborations (not team-based) whereas dynamic collaboration types (c) and (d) represent team-based collaborations. In dynamic collaboration type (a), a few users within a team collaborate for some specific purpose and form a sub-team. In collaboration type (b), a few users from different teams can independently collaborate with some external users to form a temporary collaboration. In type (c), each team allows some of its users to collaborate with other users whereas in type (d) whole team is involved in collaboration. These temporary collaborations are handled as task-based collaborations in our DySP-RBAC model.

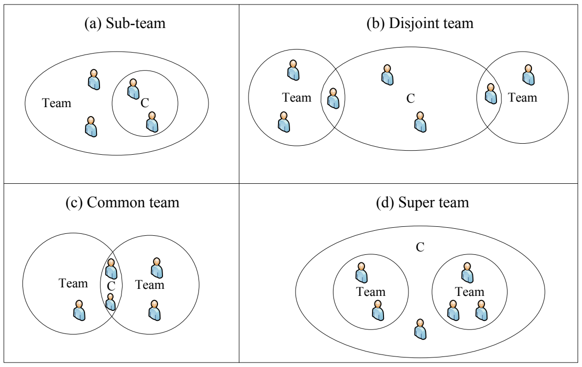

Fig. 1. Types of dynamic collaborations represented by *"C"*

We use an example from the health care domain where different teams of doctors are taking care of patients in the wards and in the emergency situations, for example, Operation Theater (*OT*) and Intensive Care Unit (*ICU*). Team members are working in different geographical areas. In emergency situations, local team members, for example, general doctors, seek help from remote team members, for example, specialist doctors. In this way, dynamic collaborations are formed which take care of specific patients. Access policy is needed to block not only the unauthorized access but also the authorized team members who are not relevant to the task. These target-oriented dynamic collaborations are dynamically formed and are dissolved after the emergency situation. Thus, there are different types of user-based collaborations among team members. User-based collaborations in sub-teams are stronger than in the main team. Members of a sub-teams need to be aware of the detailed information about their specific patient and other members of their sub-team during the emergency situation.

Context and collaborative relationships are used to access the required information. The dynamic nature of the context can add unwanted access restrictions, for example, the doctor moves out of the *OT* and access is denied. Access policy should be dynamic and flexible enough to provide the requested information at a certain level of granularity to the authenticated members by relaxing the access policy rules based on the current context.

III. DYNAMIC ACCESS CONTROL POLICY

The access control policy, in our system, is defined for collaborating teams and their dynamic collaborations. A dynamic collaboration, in our system, is defined as "*a joint action by two or more teams or their members for accomplishing a mutual task*". We extend RBAC model for specifying our access control policy. As our scenario is related to collaborative environments, we need a dynamic and adaptive access control policy. This policy includes roles, context (of subject, object and owner), and sharing/privacy constraints. As the context is a dynamic entity which changes frequently so context-based access can create unwanted access restrictions. For example, a subject is given access to an object based on her role and current context. After some time her context changes, for example, due to change in location or time, the granted access will be revoked as soon as the context is changed. There may be a case that both the owner and the requester of the context want to continue access to the context information, so the context-based access rules need to be relaxed. It can be based on current collaborative situation, i.e., seeing the other related context entities like continuation of a mutual task and the history of information access. This requires continuous monitoring of current tasks and context on both sides; the owner and the requestor.

Context of all the participating entities is used in our system. Context of collaboration, collaborative task, subject, owner and object is used to help evaluate the access control decision. Access control rules become adaptive and change with the changing situation, but not on the cost of privacy of the owner. For example, task context is being shared among team members for a specific period of the time. After the given period access is denied, but the task is still continued. If the subject wants to continue access to the task context, continuation of a task gets priority over the time limit and the system can grant access for the lifetime of the mutual task. According to different situation requirements, some context items get priority over the others. For example, context rule for request of a subject says access is possible only between the office hours say from 9:00 am to 5:00 pm. If an important task is continued before or after this time for which the access of context is required then the task status gets priority over the time-dependent context condition.

The system can use priority levels for the different situations. Foreseeing and modeling all such situations is difficult, so the system also learns from the history. It searches the history of all past accesses for this object, from this subject or from other subjects having similar roles. It also evaluates context-based conditions by matching the subject context and the policy context. If it finds any matching patterns then the rules adaptation can occur and the access can be granted.

Most of the information used in our system is related to context information which is dynamic in nature and describes the current situation of the entity and environment. Context plays fundamental role in collaborative information sharing applications. Context information can consist of user's location, devices, current tasks, and information related to user and environment which can be directly provided by user or can be measured automatically using sensors. Context can also help in defining sharing rules and context-based conditions. Our system manages context information at different level of details and uses different types of contexts like personal context, shared context, historical context. Context-based constraints are added in the sharing rules which must be evaluated to true to share context information. Context-based rules help in creating dynamic sharing policy. Sharing rules are adapted dynamically at runtime based on current context of user and her environment. Policy for access control is defined by the enterprise in form of roles assigned to each user based on her duties. Role is used to access services according to policy defined for role. Policy can contain context-based conditions so that it can be dynamically changed in accordance with the new context conditions. In our system, we use context constraints for policy description which makes it dynamic policy at runtime.

Following types of context information is used by our context model.

- Personal context includes user and her environment-related features. For example, user's location, resources etc.
- Shared context consists of task and team-related features, for example, current tasks, task status, scheduled tasks, calendar of tasks.
- Collaborative relationship context, for example, collaborating users in a task, team, or enterprise, and their collaboration level.
- History of collaboration-related features, for example, sharing history, number of accesses, type of sharing.

IV. Dynamic Sharing and Privacy-Aware Role-Based Access Control (DySP-RBAC) Model

Like NIST standard RBAC model [4], our DySP-RBAC model is composed of three DySP-RBAC components: core DySP-RBAC, hierarchical DySP-RBAC, and constrained DySP-RBAC. Hierarchical component of the DySP-RBAC model introduces team hierarchy, task hierarchy, Collaborative relationship (Cr) hierarchy, Access level (Al) hierarchy, objects (obs) hierarchy, and purpose (pur) hierarchy in addition to the role hierarchy described in RBAC model. Using hierarchies, a higher-level entity in a hierarchy can inherit lower-level entities in the same hierarchy, for example, a higher-level role can inherit permissions of its lower-level roles. This inheritance through hierarchies is helpful in minimizing the number of authorization rules defined for each entity. Constrained DySP-RBAC model contains static separation of duty and dynamic separation of duty components. Separation of Duty (SoD) is an important constraint mostly used in security models and is a fundamental requirement in RBAC model. Its main role is to separate the sensitive combination of duties in an enterprise at design time as well as at runtime to ensure that fraud or major errors cannot occur. In RBAC model, two types of constraints are called Static Separation of Duty (SSD) and Dynamic Separation of Duty (DSD). Static separation of duty in RBAC is used to restrict number of roles assigned to users from a specific role set. It also describes restrictions in presence of role hierarchies. Dynamic separation of duty is used to restrict number of roles activated in a session. DySP-RBAC uses team-based and task-based SoD constraints in addition to role-based SoD constraints. Due to page limitations hierarchical and constrained DySP-RBAC model components are not discussed in this paper. We describe the core DySP-RBAC component in detail in the following subsection.

A. Core DySP-RBAC

DySP-RBAC model shown in Figure 2, is based on the notion of dynamic sharing and privacy of the information being shared among collaborating users working in multiple overlapping teams and tasks. A task and context related information sharing among collaborating users helps in fulfilling collaborative tasks efficiently. In dynamic team-based and task-based environments, user can join and leave a team whenever needed, and can participate in more than one teams and tasks at a time. Such a dynamic environment requires active access control, in which access decisions can be adapted at runtime using changing collaborative relationships and context conditions.

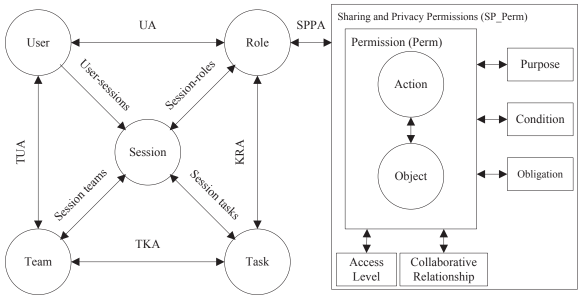

Fig. 2. Core DySP-RBAC model

Our DySP-RBAC model is based on and extends NIST standard RBAC [4]. We introduce the notion of team and task data elements in addition to standard user, role, and permission data elements. For sharing and privacy, we introduce sharing elements Collaborative Relationships (CR) and Access Level (AL), and privacy elements Purpose (Pur), context-based Condition (Con), and Obligations (Obl). User, in our system, is a human being. Role represents a job function in an enterprise. Objects, in our system, are information related to collaborating users, their team, task, resources, or their environment context. Action, in our system, is executable image of a program, which user can execute to perform some function. Permission (Perm) is an action allowed on an object. Sharing and privacy-based permissions (SP_Perm) include sharing and privacy data elements. SP_Perm are assigned to roles. Conditions, in our system, include context-based privacy conditions as well as entity-based sharing conditions that allow or restrict one or more involved entities (user, role, team, task, enterprise, and collaborative relationship).

Sharing data elements (CR, AL) control the level of information sharing among collaborating users. The CR element restricts the sharing of information to only those users who are in certain collaborative relationship with each other. Information, in our system, is mostly related to task and team-related information or context information related to a user and her current environment. This information is stored in different levels of hierarchies. The AL element uses these information levels to control the sharing of information among users having different collaborative relationships and roles. Controlling information sharing using collaborative relationships results in enhanced level of information flow among certain collaborating users, for example, users working in same team and performing same task will enjoy enhanced information sharing among each other to efficiently complete the mutual task. It depends on the priority of collaborative relationships that uses priority of data elements.

The Privacy elements (Pur, Con, and Obl) are used to preserve the privacy of user information which is going to be shared using CR and AL. Privacy elements restrict the information being shared, asking the purpose of information being accessed, and fulfilling the context-based conditions and certain obligation. Purpose of access should confirm to the purpose defined for the object being accessed which is called purpose binding. Conditions related to user, her team, task, or environment must be satisfied. Obligations required to access the information, for example, logging the access details, should be performed to gain access to required information.

The team and task are data elements introduced in DySP-RBAC to extend RBAC model. One or more users are assigned to a team and one or more tasks are assigned to a team. Users being part of a team can participate in certain tasks that are assigned to them by their team. A user can participate in more than one teams and more than one task at a time. Users in one task collaborate to perform the task and share their task related information as well as their context information among each other. Users in a team can be assigned same or different tasks created by their team. They can share information related to their team and their context.

Dynamic collaborations described in our scenario are handled easily in DySP-RBAC using team and task relation. The relation between team and task data elements is a many to many relation. A team can create many tasks each performed by a some of its members, thus creating a dynamic collaboration within a team. Members of this dynamic collaboration (single team task in DySP-RBAC) are allowed to share more information among each other than other team members. This is controlled by sharing data elements (CR, AL) in DySP-RBAC. Due to many to many team-task relation, a task can be created mutually by many teams. Members from all these collaborating teams can perform the task, thus creating a dynamic collaboration across the teams. These members of a dynamic collaboration across teams can also share their personal context-related and task-related information among each other, but they are restricted to some extent by their team and enterprise-based privacy elements (purpose, condition, and obligations in DySP-RBAC model).

One or more users can be assigned to one or more roles. Roles are also assigned to task. A user can participate in a task only if she is assigned the same role/s that is/are assigned to the task. A session is a mapping of one user to one or more roles, teams, and tasks. A user can establish a session and can activate her roles, teams, and task in that session. One session is associated to one user only called $session\_user$ and one user can establish more than one sessions called $user\_sessions$.

B. Formal Description of Core DySP-RBAC

Here we describe the data elements, their assignments and functions used in Core DySP-RBAC model. Following are the data elements in DySP-RBAC model:

U, R, Obs, Ops, T, K, Pur, Con, Obl, CR, AL are users, roles, objects, operations, teams, tasks, purposes, conditions, obligations, collaborative relationships, and access levels.

Following are the core RBAC model data elements that are also used in DySP-RBAC:

- U: the set of users in our model.
- R: the set of roles.
- 2^R: the power set of R.
- Obs: the set of objects that need to be accessed/shared.
- Ops: the set of operations on objects. Operations are executable image of a program that performs some operation on objects when invoked by user. Operations (Ops) are shown as *"Action"* in DySP-RBAC model.

Following are the main data elements added in DySP-RBAC model to extend RBAC model:

- T: the set of teams in the system.
- K: the set of tasks assigned to a team/teams.

Following elements are added to preserve the privacy of user's information:

- Pur: the set of purposes defined for accessing objects in the system.
- Con: consists of disjunction of conjunctive statements containing context variables, operators, and values.
- Obl: the set of obligations that must be fulfilled as a result of access permission.

Following elements are added to control/enhance sharing of information:

- CR: the set of collaborative relationships defined in system.
- AL: the set of access levels (hierarchical levels defined for objects) defined to allow access at certain level of granularity.

Following are the assignment relations among elements of DySP-RBAC model:

- $UA \subseteq U \times R$, a many to many mapping user-to-role assignment relation.
- $Perm = 2^{(Ops \times Obs)}$, the set of permissions.
- $SP\_Perm = (Perm, CR, AL, Pur, Con, Obl)$, the set of sharing and privacy based permissions.
- $SPPA \subseteq SP\_Perm \times R$, a many to many mapping sharing and privacy based permission-to-role assignment relation.
- $KRA \subseteq K \times R$, a many to many mapping task-to-role assignment relation.
- $TKA \subseteq T \times K$, a many to many mapping task-to-team assignment relation.
- $TUA \subseteq T \times U$, a many to many mapping user-to-team assignment relation.

V. Discussion

In this section, we describe the usage and benefits of our model in dynamic collaborations that are described in our scenario. First we describe the evaluation of the final user permissions in our model and second we describe the effectiveness of our model in dynamic collaborations.

A. Final User Permissions

A user in DySP-RBAC establishes a session and activate some roles and teams. After that only those tasks can be activated that are assigned to the user. Team roles in a session are the intersection of team-user roles and team-task roles. Permissions available to a team are the union of permissions in all active team roles. Permission available to a user are the union of permissions in all user roles and all team permissions. In this way, a user gets all those permissions that are assigned by her given roles and by her team-tasks. Finally, these user permissions are constrained by the sharing and privacy elements in the DySP-RBAC model. The privacy elements can apply context-based conditions on user permissions and sharing elements apply collaborative relationship and access level constraints.

B. Dynamic Collaborations

The collaborations described in our system are of various types. Team based collaborations and user based collaborations are defined in our scenario. Teams in our system are dynamic where users can join and leave but there are emergency situations which need collaboration of different team members or with external users. These are temporary and short lived situations. To handle these temporary situations, instead of creating a new team, DySP-RBAC creates a new task and assigns different users across teams. This task is activated and task-based permissions can be assigned to users that can use them for sharing required information for lifetime of the task. As task can be activated independently in our model, it can be used to assign task-based roles and rights to users. In this way, inter-team and intra-team dynamic collaborations can be handled dynamically and users get permission through this dynamic collaboration.

VI. Background and Related Work

In collaborative working environments, access control is a requirement for controlling the access to a user's personal information. In past access control used simple two dimensional matrix [5]. This model is not scalable because access right management is difficult with large number of users and their rights. Role-Based Access Control (RBAC) model described by [2] and standardized in [4] is used to group number of rights into a role which can be granted to one or more users. This idea makes management of rights very easy and makes it a scalable model, but RBAC is a rather static model and our scenario is dynamic so it needs to be extended. RBAC model is different from traditional Mandatory Access Control (MAC) and Discretionary Access Control (DAC) models. It not only handles the read and write access for objects but a number of permissions having complex operations. Role-based access policies have been widely used in collaboration systems due to its scalable nature and ease of maintenance. It reduces cost and complexity of the security administration.

A study based on user preferences for access control in awareness systems is presented in [6]. It shows that in collaborative awareness applications users prefer to create different groups for managing access rights. The system described in [7] presents a subject object based access control model for collaborative environments and identifies different roles of users and their collaborative rights. The research work presented in [8] describes the use of RBAC in collaborative systems. The access control system [1] presents a description of access policies for collaborative environments but does not handle user defined policies. An owner-based sharing control system is presented in [9]. It uses owner and administrator-based hybrid policy and makes use of context of all involved entities in the system. A survey on context-based systems is presented in [10]. A hybrid sharing control policy model is presented in [11] which describes the hybrid policy and entity priority-based methods for handling conflicts in hybrid policies. The RBAC model is a very efficient model for the management of large scale access control systems. Still its core model lacks in many areas of handling collaborative systems. The RBAC model lacks in handling of fine-grained access control which is required frequently in collaborative systems. Also RBAC is a static model, while the collaborative environments are mostly dynamic in nature which make use of different types of contexts, collaborations, and tasks.

Some of the systems have defined new types of roles for users or context. A merger of role and context-based access control called environment roles is introduced by [12]. This can be used in context-aware systems which need environment context to change the access rights dynamically. The system in [13] models context as context roles. The context role and user role are activated in a session for access decision. The work in [14] modifies the concept of role in RBAC with the notion of team called Team-based Access Control (TMAC). It adds user and object context with roles which are used for accessing objects. The C-TMAC model is a context-based variation of the TMAC model proposed by [15]. The system described in [16] presents owner-created roles to access context and is not suitable due to management overhead.

None of these systems make use of owner-based sharing, collaborations, collaborative relationships, privacy elements, and context for all entity types used in collaborations. For privacy of information and enhances sharing, we use privacy and sharing data elements along with context conditions. Temporary collaboration are used in emergency situations and are short lived - they are handled using task entity in our model.

VII. Conclusion

This paper extends the RBAC model to include team, task, sharing, and privacy elements that enhance sharing in dynamic collaborations and preserve the privacy of owner information in a dynamic collaborative team environment. The DySP-RBAC model is presented and its use in dynamic collaborations (using task elements) is described. Context is used to cope with the dynamic nature of environment. Context constraints related to all entities in the environment must be satisfied to access required information. Information is arranged in a hierarchical model so that only required level of information is shared. Future work includes the detailed investigation of each type of temporary collaboration described in the system.

References

[1] W. Tolone, G.-J. Ahn, T. Pai, and S.-P. Hong, "Access control in collaborative systems," *ACM Computing Surveys*, vol. 37, no. 1, pp. 29–41, Mar. 2005.

[2] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, Feb. 1996.

[3] H. Shen and F. Hong, "A context-aware role-based access control model for web services," in *IEEE International Conference on e-Business Engineering (ICEBE 2005), Beijing, China*, 2005, pp. 220–223. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/ICEBE.2005.1

[4] D. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," ACM Trans. on Information and System Security (TISSEC), 4(3): 224-274, Aug. 2001.

[5] R. Sandhu and P. Samarati, "Access control: principle and practice," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 40 –48, sep 1994.

[6] S. Patil and J. Lai, "Who gets to know what when: configuring privacy permissions in an awareness application," in *CHI*. ACM, 2005, pp. 101–110. [Online]. Available: http://doi.acm.org/10.1145/1054972.1054987

[7] H. Shen and P. Dewan, "Access control for collaborative environments," in *Proceedings of ACM CSCW'92 Conference on Computer-Supported Cooperative Work*, 1992, pp. 51–58.

[8] G.-J. Ahn, L. Zhang, D. Shin, and B. Chu, "Authorization management for role-based collaboration," in *IEEE International Conference on Systems, Man and Cybernetics, 2003.*, vol. 5, oct. 2003, pp. 4128 – 4134.

[9] A. K. Malik, H. L. Truong, and S. Dustdar, "DySCon: Dynamic sharing control for distributed team collaboration in networked enterprises," in *IEEE Conference on Commerce and Enterprise Computing, CEC 2009, Vienna, Austria*, 2009, pp. 279–284. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/CEC.2009.55

[10] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, vol. 2, no. 4, pp. 263–277, 2007. [Online]. Available: http://dx.doi.org/10.1504/IJAHUC.2007.014070

[11] A. K. Malik and S. Dustdar, "A hybrid sharing control model for context sharing and privacy in collaborative systems." AINA 2011, Singapore: IEEE Computer Society, 2011, pp. 879–884.

[12] M. J. Covington, W. Long, S. Srinivasan, A. K. Dev, M. Ahamad, and G. D. Abowd, "Securing context-aware applications using environment roles," in *ACM SACMAT '01*, New York, NY, USA, 2001, pp. 10–20.

[13] S.-H. Park, Y.-J. Han, and T.-M. Chung, "Context-role based access control for context-aware application," in *HPCC 2006, Munich, Germany*, ser. Lecture Notes in Computer Science, vol. 4208. Springer, 2006, pp. 572–580.

[14] R. K. Thomas, "Team-based access control (TMAC): a primitive for applying role-based access controls in collaborative environments," in *Proceedings of the second ACM workshop on Role-based access control*, ser. RBAC '97, New York, NY, USA, 1997, pp. 13–19. [Online]. Available: http://doi.acm.org/10.1145/266741.266748

[15] C. K. Georgiadis, I. Mavridis, G. Pangalos, and R. K. Thomas, "Flexible team-based access control using contexts," in *Proceedings of the sixth ACM symposium on Access control models and technologies, SACMAT '01*, New York, NY, USA, 2001, pp. 21–27.

[16] C. Groba, S. Grob, and T. Springer, "Context-dependent access control for contextual information," in *IEEE ARES '07*, Washington, DC, USA, 2007, pp. 155–161.