Chapter 25
Context-Aware Privacy and Sharing Control in Collaborative Mobile Applications

Ahmad Kamran Malik
Vienna University of Technology, Austria

Atif Manzoor
Vienna University of Technology, Austria

Schahram Dustdar
Vienna University of Technology, Austria

ABSTRACT

Mobile applications are being used in every field of life. Latest advances in mobile computing technology and applications make it a new level of communication proxy for its users. Despite their power as personalized service provider and an internet connected computing device, mobile systems have their inherent limitations, like small display area and limited power and memory, which must be handled in mobile-based applications. Context-awareness is being used to cope with the limitations of mobile systems and is an important area of recent research on mobile and ubiquitous system. Context plays a fundamental role in awareness applications. Activities of mobile users can be monitored by the context provided through sensors connected with user and her environment. One of the basic requirements in context-aware mobile applications is privacy and sharing control in Collaborative Working Environment (CWE). Sharing control, in the authors' system, is the distributed and dynamic control of sharing policies and information being shared. Dynamic nature of context is helpful in making automated decisions based on the current situation, for example, dynamic adaptation of level of context information being shared among collaborating users, dynamic adaptation of sharing control decisions, and dynamic adaptation of rules for sharing control.

DOI: 10.4018/978-1-61520-655-1.ch025

1. INTRODUCTION

In past, privacy and access control policies related to users were centrally administered by enterprise. These policies were usually static in nature, without using context. In current mobile-based dynamic environments, centralized systems and access control policies are being replaced with distributed, peer to peer and, Web-based sharing control policies where users control sharing of their own data. Recent research efforts are focussing on owner-defined context-based dynamic policies for sharing control (Malik et al., 2009). There is a need to shift control of sharing policies from central administrators to owner of context. In this way, distributed and fine-grained level of sharing can be achieved. For dynamic collaborative systems, we motivate use of the term "*sharing control*" in contrast to access control, whose meanings are twofold: owner-based control of context data being shared and sharing of control policies between owner and her enterprise. In untrusted systems, owner of context wants full control of her policies and context. An owner may want to change her policies with change in context and user interaction. Apart from fully restricting user access, different users can be granted different level of access rights based on their role in enterprise and current context. In addition, resource constrained mobile devices also need content adaptation. Both of these scenarios require that context information should be organized in a way so that only certain level of data can be shared whenever needed. In mobile applications, content being sent should be based on context of receiver and receiving device. Only required level of content is sent so that low memory device can store them with less battery consumption and can be easily displayed on small screen (Dorn et al., 2007). Our research efforts include owner-based dynamic sharing control using context of all involved entities, and control of context being shared at fine-grained level of all involved entities. An owner can modify her sharing policy for any entity, for example, user, activity, team, and enterprise. Context is organized in hierarchical order and sharing control system provides context at a level that is allowed to requesting user. In the following sections, we describe research efforts in areas related to privacy, access control, mobile-based systems, context-based systems, semantic techniques, and Web services, and compare them with our sharing control techniques in CWE. At the end, we describe our architectural framework for owner-based dynamic sharing control.

Section 2 describes collaborative working environments in context of mobile applications. Section 3 explains the privacy, sharing control systems including Role-Based Access control (RBAC) and trust-based systems. In Section 4, context related issues like context-based access control and adaptation techniques for mobile and ubiquitous systems are described. Section 5 describes semantic techniques commonly used for context-based privacy and access control. Importance of Web services and SOA-based systems in mobile computing environment is described in Section 6. We discuss our research work about dynamic sharing control (DySCon) in CWE (Malik et al., 2009), in Section 7. Finally, Section 8 concludes the chapter and describes future work.

2. MOBILE-BASED COLLABORATIVE WORKING ENVIRONMENTS

Collaborative working environment (CWE) is one of the most demanding areas for mobile applications where distributed users having mobile devices collaborate to achieve a common goal. With the advancement of distributed systems and pervasive computing devices, new opportunities and challenges arise in the area of context-based collaborative systems development. Context sharing in dynamic CWE is important for knowing the current state of collaborative tasks. Mobile applications are being used in CWE which may

be trusted or untrusted. Users may not know each other, so they may not be willing to reveal their personal context and data to anonymous users. There is a need for methods and applications which preserve user's privacy without affecting the collaborative task. A description of access control systems for collaborative environments is presented in (Tolone et al., 2005). A study about CWE presented in (Skopik et al., 2008) shows that security, open standards, and open source software are considered critical issues in CWE applications. It also shows that mobility is an important requirement for CWE; so that users can work from anywhere, having any device. In addition, the support for mobility requires security, context awareness, and semantic technologies.

3. ACCESS CONTROL, PRIVACY, AND TRUST

This section describes various research efforts for providing privacy and sharing control in CWE. The use of access control and trust in CWE is also presented.

3.1 Privacy and Sharing Control

Privacy of user's context information is required in centralized as well as distributed context sharing CWE. For preserving privacy of personal context in CWE, in contrast to well known techniques of access control, we describe the concept of *sharing control*. It refers to distributed and dynamic control of sharing policies and context information being shared. Sharing control can be implemented in presence of a centralized administrator where administrator creates access policies for whole team and an owner is allowed to override centralized policies related to her personal context information, temporarily in a certain context, as described in DySCon (Malik et al., 2009). Another method is to use owner created roles and no centralized policy as described in (Franz et al., 2008). This method can create management problems for individual role and policies. The work presented in (Lederer et al., 2002) describes privacy issues in ubiquitous computing environments. In the following paragraphs, we describe systems which make use of privacy and sharing control techniques.

An enterprise-based dynamic sharing control system is described in (Malik et al., 2009) which uses peer to peer and Web services technologies. The system focuses on the privacy issues related to individual's context being shared with other members of a team. Enterprise-based policies are used for sharing control while owner-based policies override enterprise policy in certain situations, for example, owner of context wants to share her context information with other known member of a team while enterprise policy under current context conditions do not allow for it. Context conditions include context of all entities, i.e., owner, requester, activity, team, and enterprise. These contexts are used in three ways: adding context constraints in role-based sharing policy, setting predefined values for user's current status, for example, *busy*, *not at work* etc., and dynamically adapting enterprise policy based on current context values to restrict context sharing with other entities like user, activity, team, and enterprise. Context being shared is modelled in different levels of granularity so that users having different roles and involved in different activities, teams, enterprises, should get access to relevant level of context.

Use of context for context access is described in (Groba et al., 2007). It uses owner created roles for privacy of personal context and provides methods for integrity and availability of service in mobile environment. On request of a user, the owner sends her relevant roles and requester picks one of them for context access. As user created roles may not have unique names so their description is also sent to requester. A proxy, like mobile or personal computer, is used for high availability in the absence of owner, which sends required

context. In this technique, requester can face difficulty in finding the best role out of all provided roles for context access and understanding role description of each role is inconvenient for her. Semantic techniques can be helpful in bridging this gap. In addition, a hybrid role management system using enterprise-defined roles and owner-defined roles described in (Malik et al. 2010) provides a user friendly method for handling user-defined roles which are partially based on enterprise-defined roles.

Privacy-Enhancing Identity Management (PIM) and RBAC are integrated in (Franz et al., 2008) to achieve balance between convenience and privacy for protecting user's personal and context data. PIM uses partial identities (unlinked subsections of personal and context data), and pseudonyms (instead of real names) for partial identities. System also uses data abstraction; it arranges data in different levels. It describes requirements for privacy preserving system; preserving user privacy using unlinked partial identities, minimum trust in other parties, owner controlled sharing, efficiency of system, automated decisions, and feedback to owners describing who can access their personal context or data. Access control rules are used to restrict access of user to certain object and data abstraction rules are used to grant certain level of context access according to current conditions. Data abstraction and owner-controlled sharing are important contributions in this system. It hands over full access control to individual user which is difficult to manage in large scale CWE where multiple teams and organizations are involved with their own interests.

The systems described above try to preserve owner's privacy by allowing her to control her own access rules. Most of these systems are used for context-based context access. Some of these systems used owner created roles to grant access to requesters (Franz et al., 2008) which can create role management problems, while our system (Malik et al., 2009) uses techniques to allow owner to temporarily override enterprise access policy, in certain context conditions.

3.2 Access Control in CWE

In past, access control in collaborative systems, for personal or organizational data, was provided by an access matrix describing subject and its access right for object as a two dimensional matrix (Shen et al., 1992). Due to the difficulty of enterprise in access right management of large number of users, RBAC model was created by (Sandhu et al., 1996) and later RBAC standard was described in (Ferraiolo et al., 2001). In RBAC, a role can have many rights and can be assigned to many users. RBAC is static in nature where role and policies cannot be dynamically adapted at runtime based on context. Context-based access control systems are being created in past few years which try to provide active access control to fine grained level (Covington et al., 2001; Hulsebosch et al., 2005). Access control describes who can have which type of access to which object. Incorporating context into access control gives additional control by specifying context conditions i.e. who can have access to which part of data in a certain context condition. For example, context condition can be current time or location of requester or the owner of data. Access control has been commonly used as a centralized role management system where a central administrator creates and assigns roles to each member of CWE. Access control systems in CWE are described as follows.

A model for team-based access control using context constraints is described in (Georgiadis et al., 2001). In this model, users, roles and, teams can be activated or deactivated at run time, and roles can be activated by users when required. Permissions are granted to user in the following way. A user gets permissions granted by her team combined with the permissions granted by her active roles. Context is used to filter user permissions. A user finally gets permissions based on her current context conditions. This system provides the idea of team-based permission assignment in addition to role-based permissions. The plus point is team and context-based access control, while the owner-based control is not used in this system.

A team and task-based access control model is presented in (Zhou et al., 2007). Users have their individual roles, but only those roles can be activated which are also part of team roles. Permissions available to a user include the permissions from their active roles and teams. As this system describes the notion of task-based permissions, so the final permissions of users are dependent on current tasks of the user. This system used another level of permission assignment which is current task of a user. It does not allow for user-defined policy but user has limited control for his access policy through his current task.

Access control systems generally use RBAC model defined in (Sandhu et al., 1996). Some of the systems described here, for example, (Georgiadis et al., 2001), make use of context for access control. Use of context in most of the systems is limited and so they are not active access control systems (systems which activate/deactivate role and permissions based on context). Additionally, access rights management is mostly centralized and users are not allowed to change their own access rights, which is unacceptable for collaborative and mobile environments.

3.3 Trust-Based Access Control in CWE

Trust in collaborating parties enables them to confidently share requested data. A user can personally know and trust in other party or she can have an indirect trust through a trusted third party. Trust is level of confidence in other party which can be gained in different ways. Current context of both parties, history of access and collaborative relationships among users can help in making trust in other party. The following systems describe the use of trust for context sharing in CWE.

Trust can help in searching the best collaboration partner in CWE (Skopik et al., 2009). Trust-based system described in (Skopik et al., 2009) is a service-oriented system which makes use of trust for efficient collaborations within team members. Trust is calculated using past collaborations, previous successes and working competencies of team members. It aggregates contextual information of individuals and describes three views of trust; individual view, team view, and global view. The collaboration metrics for trust are calculated by aggregating data from all the data sources so that meaningful information can be deduced from it. Here trust has been used specifically for searching the most trusted collaboration partners in service-oriented CWE. Use of trust based on collaborations of users can certainly help in sharing control systems.

A trust-based access control model is presented in (Bhatti et al., 2005). It tries to enhance access control with context and instead of using identity- or capability-based access control schemes it introduces trust based on a third party. Context-aware features are required for access control and trust creation in Web services-based environments. Trust level is reduced for a user who is violating her normal profile during an access. This trust value can later be changed by analyzing context with each service access made by user. For achieving Web level scalability, trust provided by a trusted third party is used to assign roles to users instead of using user's identity or capability. Services can be provided for limited time after which they are automatically revoked. In case of access policy violation, an event-based revoke can be instantiated by user. It is an interesting system which makes use of context-based and trust-based access control policies. A centralized system is used to manage trust and access control instead of using owner-based policies.

Trust-based access is an important method for sharing control in CWE. Trust has been frequently used in CWE (Skopik et al., 2009) and access control systems (Bhatti et al., 2005) to find trusted partners for collaborations and to grant access to resources respectively. For sharing control, when sharing rules and current context does not allow accessing a resource; trust in requesting party can be used by calculating history of previous access (Malik et al., 2009).

4. CONTEXT-BASED ACCESS CONTROL AND ADAPTATION IN MOBILE-BASED SYSTEMS

Mobile-based systems are the heart of dynamic collaborative working environments. In CWE, users are mostly distributed and dynamic. During movements users commonly make use of various constrained devices like mobile devices and laptops to contact with their team and other users. Mobile devices are resource constrained, for example, they have limited battery power, memory, and display. Due to advances in mobile technologies, their advantage of anytime and anywhere connectivity is dominating their limitations. During user movements, mobile systems can act as a proxy for users and their computers, by sharing limited amount of data on their behalf. In many applications of CWE, for example, disaster scenarios, mobiles are the only way to connect and share required context of situation.

Context-based access control systems are being investigated in literature for providing required level of privacy to users. An owner can specify with whom she wants to share what level of her personal data. Using the dynamic nature of context, access rule adaptation can be performed at runtime. It is possible to dynamically adapt the behaviour of system by capturing the current context of requester, provider, resources, and environment. Context-based systems are helpful in fulfilling the owner centric dynamic access requirements (Groba et al., 2007). A survey on context-aware systems is provided in (Baldauf et al., 2007). Following are the research efforts for access control and adaptation in mobile environments.

A context-aware access control system for anonymous users is provided in (Yokoyama et al., 2006). Context is used in this system for three purposes. Firstly to subscribe events and retrieve status from sensors, secondly to use conditions and analyze situation from the events and status, and lastly to generate events according to situation. It does not use role-based system arguing that RBAC is inefficient in ubiquitous environments where relations are ad-hoc. It describes a method for sharing data between anonymous users using context collection certificates. Anonymous user provides a context collection source certificate when it needs to access an object. This certificate tells how to collect context of user from sensors. Proxies on both sides are subscribed to context monitoring on context server so that any change in context may revoke the grant. This technique can be used in large scale collaborative environments where users do not know each other and are not bound to specific roles and teams.

A high level policy description language is provided in (Ahn et al., 2006) which consists of context entity relationship definitions and context-based access control and adaptation policies for ubiquitous environments. It is a nested hierarchical tree structure language which can easily define spatial entity relations between entities, for example, spaces (building, floor, and room), static objects (printer), as well as moving objects (PDA). Access control rules describe access mode of subject to an object in a given context condition. Adaptation rules respond to events happening with entities. A java-based runtime environment is provided and given policy specifications are translated into java classes for each context entity using JCAF (Java Context Aware Framework). This policy description language can be effective in mobile-based CWE where access control and context-based adaptation are required. It can be helpful for providing owner-based privacy and adaptation techniques.

In ubiquitous environments, context of objects changes due to their movement which affects their access control requirements. These context changes are described using a meeting scenario in (Toninelli et al., 2006), for example, identities of the meeting participants may not be available or they change during meeting, so the static role-based access control techniques are not suitable. It uses context of requester, resource, and environment for policy update, using context-based

grouping and searching of required policies. Consider another example of dynamic adaptation of policies where meeting goes beyond specified time, access to required resources will automatically continue by sensing and reasoning the current context, which shows that meeting is still continued. This system provides context-based access and adaptation concepts using a meeting scenario. We use similar adaptation concepts for owner-based sharing control in CWE (Malik et al., 2009).

A context service middleware is explained in (Springer et al., 2006) which is distributed on each heterogeneous context source. A proxy is created for each remote source. Multiple source configurations and their context integration are performed. A context broker manages context sources and maintains remote peer to peer connections with other context services. Context in sources is represented as a layered model in the form of type, subtype and restriction, so that multiple domain specific context models are supported. Concepts like context collection from multiple sources, context integration and representation are described in this system. These are basic requirements of context-based systems on which privacy and access control applications are provided.

A context-aware access control model based on RBAC using a State Checking Matrix (SCM) is described in (Kim et al., 2005). It describes that access rights are changed with change in user context, and access permissions related to a resource are changed with the change in its system information like bandwidth or memory. Roles and permissions are activated in this system based on context. State checking agent maintains roles of user by monitoring context of user and changing active role of user with change in context. It maintains all contexts in the form *active* or *de-active.* When all contexts like time, location are active then the role becomes active. This is a context-based access control system describing the context-based activation/deactivation of role and permissions which can be used to restrict users from accessing services in varying context conditions.

Mobile applications are subject to context changes due to user mobility, which can invalidate context-based access rules. Context-based adaptation for mobile applications is an important area of research as shown in above described systems. A user can predefine context adaptation rules that can be predicted (Toninelli et al., 2006), and should be allowed to change the rules at runtime when required (Malik et al., 2009).

5. SEMANTIC TECHNIQUES FOR CONTEXT-BASED PRIVACY AND ACCESS CONTROL

Following systems describe the use of semantic technologies, for example, using ontology to create domain hierarchies and context hierarchies for bridging the semantic gap between queries and access control policies.

Semantic techniques are used in (Toninelli et al., 2006) to describe context and policies at high level of abstraction which allow classification and comparison of rules and context. It helps in finding conflicts between policies and creating new information from existing one in dynamic situations. It represents context with context ontology and uses description logic for classifying context models and discovering their relationships. It uses context aggregation and context instantiation rules to find collocated users and current project instantiation respectively. Semantic techniques have been used for multiple purposes in this system, for example, classification, comparison, searching, grouping, and instantiation of context and policies. These techniques are useful for context-based systems. These techniques can be used in our scenario for helping owner to control her level of context sharing with others, by grouping her context and policies and comparing them with the requirements of other users.

A semantic context-aware model is presented in (Ko et al., 2006) which uses a context ontology to bridge semantic gap between contexts specified in access rules and actual context of requester in query. Using ontology, the system arranges context in hierarchies according to abstraction levels. It arranges all concepts of a domain in order so that parent and child of each concept are evaluated. It uses reasoning rules taken from context ontology to bridge semantic gap between context in policy rules and context from requester's query. If query concept is smaller (contained) than rule concept, then requested object is granted. This system uses context ontology for solving problem of mismatch in query and policy. It can be extended to other context types and their uses like, grouping and integrating contexts, searching required context.

Dynamic nature of ubiquitous systems makes service delivery very difficult in presence of large number of services and clients (Riaz et al., 2005). Semantic technologies like, semantic attributes of services and ontology, are useful in searching and controlling access to Web services. This system presents an architecture which uses user's context instead of roles. Services and access policies are activated based on context. Only certain services and policies are enabled in a given context. It uses semantic attributes in query to match with active services, and after that it uses user's context to filter required services. Domain ontology is used to overcome semantic gap between query concepts and services. This system is different from our system (Malik et al., 2009) in that it only uses context and does not use role. Context-based service, policy activation, and domain ontology are the concepts which can be helpful in sharing control systems.

The systems discussed above make use of context ontologies to arrange context of user, her resources and environment, which can be used for matching with query context. Some systems like (Riaz et al., 2005) use domain ontology to match domain concepts with query concepts. These semantic techniques are helpful in managing resources and their context and for effective performance of queries to resources.

6. SOA AND WEB SERVICES FOR CWE

With the increased use of internet on mobile, collaborative systems are using Web-based technologies to connect distributed users. Service-oriented techniques using Service-Oriented Architecture (SOA) and especially Web services are state of the art in this area. Service-oriented computing uses the idea of assembling application components into a network of loosely coupled services to create flexible business processes and applications (Papazoglou et al., 2007). Web services are platform-independent, autonomous and reusable entities which use Internet and open standards like Simple Object Access Protocol (SOAP) for communication and Web Services Description Language (WSDL) for defining services. As more and more mobile applications are using Internet and Web services, it seems that Mark Weiser's view of ubiquitous computing (Weiser, 1993) is not far from today. Some context-based access control systems which make use of Web-based and ubiquitous techniques are described here.

Context sharing for mobile Web services is described in (Dorn et al., 2007). This paper describes that context can enhance Web services in mobile environments. To protect privacy of users, it manages context in levels of hierarchies, and presents a context access control, subscription and query language which allow fine-grained subscriptions and control over context. To handle resource constraints of mobile devices, it minimizes the amount of context data sent by Web services on mobile devices using context hierarchies. Each hierarchy contains context at deeper level. Hierarchies are general and using similar semantics, different domains and context models (ontology,

object-oriented, key value) can map context data into hierarchies. It also uses context dominance concept; changes in less important context are not sent in presence of dominant context. A Web service, in this system, can automatically communicate with other Web services and can subscribe for required context changes. In all, it is a mobile-based and Web service-based system which uses concepts like context hierarchies, and dynamic context-based query and subscriptions, for sharing context. In our owner-based sharing control system, we are using the concept of context hierarchies. Context-based subscriptions are helpful in saving time and resources rather than repeated querying.

Importance of context-aware models for access control using Web services is described in (Haibo et al., 2005). This paper presents a context-aware role-based access control model using composite Web services and global roles. It realizes the need for access control in Web services and especially composite Web services. It describes mechanism to access composite Web service (global service). Global roles are used to access global services and local roles are used to access local services. Global role is activated only when user accesses a service and presents her security certificate as well as current context. User must have minimum global role to activate global service. Minimum role is the one having minimum rights to access global service. Global roles are automatically mapped to local roles when a service is accessed. This system presents another view of Web services-based access control using global roles and services to access the composite Web services. Managing global roles, local roles and services for large enterprise-based systems is difficult and can face scalability issues. Also it is managed centrally which is not suitable for our decentralized and peer to peer scenario.

Web service-based systems explained above describe the use of Web services for context-based access control. Our system (Malik et al., 2009), uses Web services for dynamic context sharing and owner-based privacy. Some of the systems shown above describe context sharing (Dorn et al., 2007), while owner-based dynamic policy adaptation is not described. In addition, some other interesting concepts have been described like Web services-based subscriptions for context sharing (Dorn et al., 2007) and composite Web services (Haibo et al., 2005). Due to the platform-independent nature of Web services they can be used with various platforms and operating systems, and well suited for the dynamic mobile-based environments.

Table 1 shows a summary of research efforts described above. In particular, it indicates various techniques used for access control and privacy and shows the relevant research works.

7. DYNAMIC SHARING CONTROL FOR CWE

In this section, we describe the development of our sharing control architecture for CWE. Our Dynamic Sharing Control (DySCon) architecture is given in (Malik et al., 2009) and shown in Figure 1. This architecture includes methods to preserve the privacy of a user's context while sharing it with collaborating users in a dynamic team-based environment created by different cooperating enterprises. It allows owner-based dynamic adaptation of policies that were defined by enterprise for the role of individual users. Owner-based rules override enterprise defined rules temporarily for specific purposes at different entity levels using many types of context.

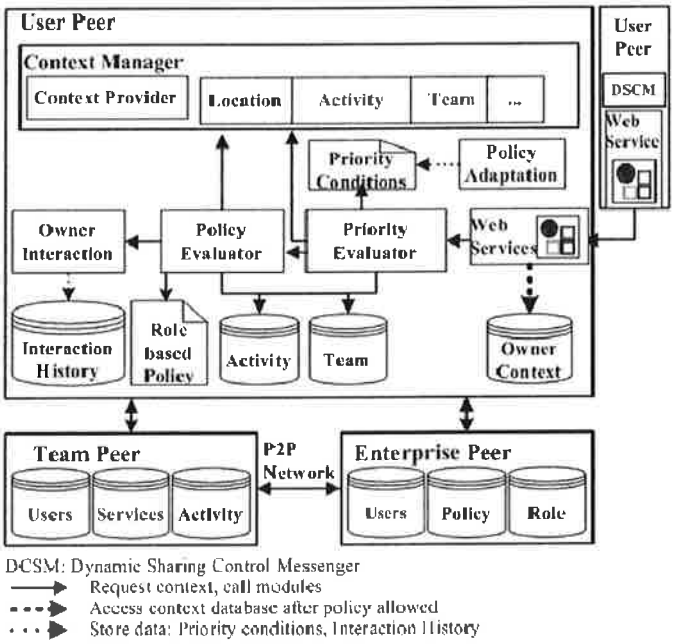
It tries to model an important requirement of sharing in collaborative working environments, i.e., when to share which context with a particular member, team or enterprise at what granularity level. Two types of context services are described here, *personal context services* related to context of individual user, and *shared context services* related to context of team or enterprise. We want to preserve the privacy of user and so our goal is privacy of user's personal context.

Table 1. Techniques for context-based sharing and privacy in mobile-based CWE

Methods	*Descriptions*	*Used by*
Context	Context hierarchy	(Dorn et al., 2007; Springer et al., 2006)
	Context-based adaptation	(Toninelli et al., 2006; Malik et al., 2009; Ahn et al., 2006; Yokoyama et al., 2006)
	Context-based context access	(Malik et al., 2009; Franz et al., 2008)
Access policy	RBAC	(Zhou et al., 2007; Bhatti et al., 2005; Kim et al., 2005)
	Owner-based	(Malik et al., 2009; Franz et al., 2008)
	Trust-based	(Skopik et al., 2009 ; Bhatti et al., 2005)
Web services	Web service access	(Dorn et al., 2007; Haibo et al., 2005; Malik et al., 2009)
	Web service search	(Riaz et al., 2005)
Semantics	Context ontology	(Ko et al., 2006; Toninelli et al., 2006)
	Domain ontology	(Riaz et al., 2005)
Mobile	Mobile-based access	(Toninelli et al., 2006; Malik et al., 2009; Yokoyama et al., 2006)
	Content adaptation	(Dorn et al., 2007)

Context of all involved entities is used by our model including requester, owner, team, activity and enterprise. DySCon uses context for multiple purposes: to share current context of a user with other users in CWE, to use context-based constraints to restrict access to context services, and for policy adaptation by owner. Policy adaptation can be performed in two ways: either predefined

Figure 1. DySCon architecture (2009, IEEE Computer Society, Malik et al., 2009)



templates like *busy, not at work,* are used, or an entity (user, team, activity or enterprise) can be restricted by the owner from getting access to one or more services.

As requirements of dynamic CWE are distributed, dynamic, and mobile-based environment, our DySCon architecture is based on a peer to peer model which is well known for its distributed, dynamic and autonomous nature. Interaction between peers is handled by the Web services. These are context-based Web services to share personal context of users within and across activities, teams, and enterprises. As we consider dynamic collaborative environment where one user can be part of multiple teams created by different enterprises, so the same level of context cannot be shared with all users. DySCon manages context at three levels to share with different types of users. For example, a user will share the details of her context with collaborating users who are working within same activity with her and are related to her enterprise, whereas the user will share context only at a lower level of granularity with other users, who are not members of same activity, team, or enterprise, depending on their nature of collaboration with her.

DySCon Architecture and Dynamic Sharing Control Policy

DySCon architecture consists of three peer types; user peer, team peer and enterprise peer. These peers are connected with each other through a P2P network. Enterprise peer controls users, their roles, and policies which are used by team peer and user peer. Team peer controls users of team, their activities and services provided by team, which are used by user peers to find details of their team members and activities. User peer consists of a context manager, various policy descriptions and evaluation modules, and Web services to connect with other peers. The context manager collects context from external context sources and manages each context item at three granularity levels. Context managed by context manager include, local context features related to user, shared context features related to team, activity, and enterprise, and collaborative context features related to history of collaborations.

A user peer calls the required service of another user peer, requested peer replies by asking the requester's current context. A requester sends her context if she wants to share her context, policy evaluation is performed at requested peer which can result either in deny, or grant the requested context at a particular level of granularity defined by sharing control rules of requested peer. Owner of context can define own rules, overriding the enterprise-defined rules to allow some user, activity, team, or enterprise to gain access at a

Figure 2. Example of owner-based priority rules (2009, IEEE Computer Society, Malik et al., 2009)

```
<PriorityRules>
  <priority>
    <resource>activity service</resource>
    <role>developer</role>
    <team>team1</team>
    <activity>
      <name>planning</name>
      <status>continued</status>
    </activity>
    <action>allow</action>
    <Level>L1</Level>
  </priority>
  <priority>
    <resource>location service</resource>
    <role>leader</role>
    <team>team2</team>
    <activity>
      <name>design</name>
      <status>continued</status>
    </activity>
    <action>allow</action>
    <Level>L3</Level>
  </priority>
</PriorityRules>
```

Figure 3. Example of sharing control policy for developer role (2009, IEEE Computer Society, Malik et al., 2009)

```
<policy>
 <resource>
  <location service>
    <condition>
      <loc>office</loc>
      <time>9:00 to 17:00</time>
    </condition>
    <access level>
      <teammembership>same team</teammembership>
        <level>L1</level>
      <teammembership>different team</teammembership>
        <level>L2</level>
    </access level>
  </location service>
  <activity service>
    <condition>
      <activity status> continued <activity status>
      <loc>office</loc>
    </condition>
    <access level>
      <activitymembership>same activity</activitymembership>
        <level>L1</level>
      <teammembership>same team</teammembership>
        <level>L2</level>
    </access level>
  </activity service>
 </resource>
</policy>
```

predefined level of granularity of context in certain context conditions. These owner-defined rules are defined in priority conditions document shown in Figure 2. First, the priority evaluator accepts the request and compares against the priority conditions document, to find if there exist some priority conditions for requester. If the priority conditions exist and current context of requester match given conditions then she is granted access to context at a granularity level defined for her. In case of no priority condition found for the requester, policy evaluator compares the request with role-based policy containing enterprise-defined rules for the role assigned to requester as shown in Figure 3. Requests with mismatched context are logged into a file waiting for owner interaction and are handled manually by owner using interaction history and context of the requesting user.

8. CONCLUSION AND FUTURE WORK

In this chapter, privacy, access control, and context related issues for mobile-based CWE are presented. We explained existing systems in detail. Main focus of the chapter is to describe privacy and sharing control of user context information in mobile-based CWE. At the end, we described our research in this area as the *Dynamic Sharing Control (DySCon) Architecture* which provides methods to achieve the desired level of privacy for user context while sharing it with collaborating users. DySCon uses various contexts at different levels of granularity at all entity levels, and allows the owner of context to dynamically change enterprise-defined access rules for sharing her context.

Most of the existing systems use enterprise-based, RBAC-based, and centralized concepts for access right management of users. Centralized policy is undesirable in situations where users want to grant access to other collaborating users using dynamic context conditions. Our system DySCon, allows owner of context to override enterprise-based access policy defined for roles. Another choice is to use owner-defined roles rather than enterprise-defined roles. As there can be many users and roles in a CWE, owner-defined roles can create management problems for the requester and owner. There is trade off between owner privacy and management of access rights. There is need to define methods which can reduce role management and rights management burden in dynamic CWE while preserving the privacy of individuals. A hybrid method consisting of enterprise-defined roles and owner-defined roles is one of the solutions. This method can be effective only if total number of roles in system is controlled, and clear boundaries are defined between enterprise roles and individual roles. In addition, sharing control for shared services need to be provided, like the services used by activities, team and, enterprise. In general, research efforts are required in the areas of context-aware services-based sharing control, privacy, and adaptation in dynamic mobile environment. Effective use of semantic techniques are needed in the areas related to context description, context-based access, searching, grouping, and finding conflict in sharing policies.

REFERENCES

Ahn, J., Chang, B.-M., & Doh, K.-G. (2006). A policy description language for context-based access control and adaptation in ubiquitous environment. In *EUC Workshops, Lecture Notes in Computer Science 4097,* (pp. 650-659).

Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, *2*(4), 263–277. doi:10.1504/IJAHUC.2007.014070

Bhatti, R., Bertino, E., & Ghafoor, A. (2005). A trust-based context-aware access control model for Web-services. *Journal of Distributed and Parallel Databases*, *18*(1), 83–105. doi:10.1007/s10619-005-1075-7

Covington, M. J., Long, W., Srinivasan, S., Dey, A. K., Ahamad, M., & Abowd, G. D. (2001). Securing context-aware applications using environment roles. *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies, SACMAT,* Chantilly, Virginia, USA. (pp. 10-20).

Dorn, C., & Dustdar, S. (2007). Sharing hierarchical context for mobile Web services. *Journal of Distributed and Parallel Databases*, *21*(1), 85–111. doi:10.1007/s10619-006-7005-5

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security*, *4*(3), 224–274. doi:10.1145/501978.501980

Franz, E., Groba, C., Springer, T., & Bergmann, M. (2008). A comprehensive approach for context-dependent privacy management. *Proceedings of the Third IEEE International Conference on Availability, Reliability and Security, ARES*, (pp. 903–910). Washington DC, USA.

Georgiadis, C. K., Mavridis, I., Pangalos, G., & Thomas, R. K. (2001). Flexible team-based access control using contexts. *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies, SACMAT*, New York, USA (pp. 21–27).

Groba, C., Gross, S., & Springer, T. (2007). Context-dependent access control for contextual information. *Proceedings of the Second IEEE International Conference on Availability, Reliability and Security, ARES*, (pp. 155–161). Vienna, Austria.

Haibo, S., & Fan, H. (2005). A context-aware role-based access control model for Web services. *IEEE International Conference on e-Business Engineering (ICEBE)*, (pp. 220-223).

Hulsebosch, R. J., Salden, A. H., Bargh, M. S., Ebben, P. W. G., & Reitsma, J. (2005). Context sensitive access control. *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies, SACMAT*, (pp. 111-119). Stockholm, Sweden.

Kim, Y.-G., Mon, C.-J., Jeong, D., Lee, J.-O., Song, C.-Y., & Baik, D. K. (2005). Lecture Notes in Computer Science: *Vol. 3528. Context-aware access control mechanism for ubiquitous applications. Advances in Web Intelligence, AWIC* (pp. 236–242). Berlin, Germany: Springer-Verlag.

Ko, H. J., Won, D. H., Shin, D. R., Choo, H. S., & Kim, U. M. (2006). A semantic context-aware access control in pervasive environments. *Computational Science and Its Applications, Lecture Notes in Computer Science, 3981*, (pp. 165 174). Berlin, Germany: Springer-Verlag.

Lederer, S., Dey, A. K., & Mankoff, J. (2002). Everyday privacy in ubiquitous computing environments. *Workshop on socially-informed design of privacy-enhancing solutions in ubiquitous computing,* Goteborg, Sweden.

Malik, A. K., & Dustdar, S. (2010). *Context-aware sharing control using hybrid roles in inter-enterprise collaboration.* Fifth International Conference on Software and Data Technologies, ICSOFT 2010. Athens, Greece.

Malik, A. K., Truong, H.-L., & Dustdar, S. (2009). DySCon: Dynamic sharing control for distributed team collaboration in networked enterprises. *International Conference on Commerce and Enterprise Computing, IEEE CEC*, (pp. 279-284). Vienna, Austria.

Papazoglou, M. P., Traverso, P., Dustdar, S., & Leymann, F. (2007). Service-oriented computing: State of the art and research challenges. *IEEE Computer*, *40*(11), 38–45. doi:10.1109/MC.2007.400

Riaz, M., Kiani, S. L., Lee, S., Han, S.-M., & Lee, Y. K. (2005). Service delivery in context aware environments: Lookup and access control issues. *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA,* (pp. 455–458). Washington, DC, USA.

Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-Based Access Control Models. *IEEE Computer*, *29*(2), 38–47. doi:10.1109/2.485845

Shen, H., & Dewan, P. (1992). Access control for collaborative environments. *Proceedings of the ACM Conference on Computer-Supported Cooperative Work,* (pp. 51-58). Toronto, Ontario, Canada

Skopik, F., Truong, H.-L., & Dustdar, S. (2008). *Current and future technologies for collaborative working environments. Study Report, ESA ITT Number AO/3-12280/07/NL/CB, Distributed Systems Group*. Austria: Vienna University of Technology.

Skopik, F., Truong, H.-L., & Dustdar, S. (2009). VIETE - Enabling trust emergence in service-oriented collaborative environments. *5th International Conference on Web Information Systems and Technologies, WEBIST,* (pp. 471-478). Lisbon, Portugal.

Springer, T., Kadner, K., Steuer, F., & Yin, M. (2006). Middleware support for context-awareness in 4G environments. *Proceedings of the International Symposium on World of Wireless, Mobile and Multimedia Networks, IEEE WoWMoM,* (pp. 203-211).

Tolone, W., Ahn, G.-J., Pai, T., & Hong, S.-P. (2005). Access control in collaborative systems. *ACM Computing Surveys, 37*(1), 29–41. doi:10.1145/1057977.1057979

Toninelli, A., Montanari, R., Kagal, L., & Lassila, O. (2006). A semantic context-aware access control framework for secure collaborations in pervasive computing environments. In *International Semantic Web Conference, ISWC, Lecture Notes in Computer Science, Vol. 4273* (pp. 473-486). Berlin, Germany: Springer-Verlag.

Weiser, M. (1993). Some computer science issues in ubiquitous computing. *Communications of the ACM, 36*(7), 75–84. doi:10.1145/159544.159617

Yokoyama, S., Kamioka, E., & Yamada, S. (2006). An anonymous context aware access control architecture for ubiquitous services. *7th International Conference on Mobile Data Management, MDM,* (p. 74).

Zhou, W., & Meinel, C. (2007). Team and task based RBAC access control model. In *Latin American Network Operations and Management Symposium, LANOMS,* (pp. 84–94).

KEY TERMS AND DEFINITIONS

Collaborative Mobile Applications: The applications created for CWE making use of mobile-based devices for communication and sharing information among collaborating users.

Collaborative Working Environment (CWE): A distributed and dynamic environment in which users work for one or more teams and enterprises, collaborate for achieving a common goal.

Context-Aware System: Uses context data taken from various types of sensors to create awareness about user and her environment, and can use context for various purposes including sharing of context information, context-based access control and, context-based adaptation.

Context-Based Access Control: In this type of access control access control policies can be dynamically evaluated and adapted using context-based conditions.

Dynamic Collaborations: Dynamic collaborations are the temporary collaboration in which users can join and leave the teams whenever needed.

Owner-Based Privacy: Owner-based control of her privacy and sharing control rules for her personal information being shared with other users.

Sharing Control: Describes two purposes: Firstly, it is distributed and dynamic control of sharing policies among user and her enterprise, and secondly, it describes owner-based control of the information being shared.