USER-CENTRIC ADAPTIVE ACCESS CONTROL AND RESOURCE CONFIGURATION FOR UBIQUITOUS COMPUTING ENVIRONMENTS

Mike White, Brendan Jennings, Sven van der Meer
Telecommunications Software & Services Group (TSSG),
Waterford Institute of Technology, Waterford, Ireland

Keywords: Access Control, Policy Based Management, Conflict Detection and Resolution, Ubiquitous Computing.

Abstract: Provision of adaptive access control is key to allowing users harness the full potential of ubiquitous computing environments. In this paper, we introduce the M-Zones Access Control (MAC) process, which provides user-centric attribute-based access control, together with automatic reconfiguration of resources in response to the changes in the set of users physically present in the environment. User control is realised via user-specified policies, which are analysed in tandem with system policies and policies of other users, whenever events occur that require policy decisions and associated configuration operations. In such a system users' policies may habitually conflict with system policies, or indeed other users' policies; thus, policy conflict detection and resolution is a critical issue. To address this we describe a conflict detection/resolution method based on a policy precedence scheme. To illustrate the operation of the MAC process and its conflict detection/resolution method, we discuss its realisation in a test bed emulating an office-based ubiquitous computing environment.

1 INTRODUCTION

Ubiquitous computing systems typically employ a greater range of user interfaces than traditional computing environments; for example, communal displays, voice-based command interfaces, and gesture recognition systems. The presence of such interfaces heightens users' awareness of, and requirements for, privacy protection measures. When using such interfaces, users may desire that the system automatically reacts to the presence and/or activities of other individuals in their physical vicinity – typically so that their privacy concerns are addressed. In such cases the process of defining what actions need to be taken will depend on the relationships between a user and these other individuals. Furthermore, a user's access rights should be determined based on ongoing analysis of the access rights and activities of other individuals present in his/her environs.

In this paper we investigate how a management system for a ubiquitous computing environment can control access rights in a manner that adapts to the changing profile of the set of individuals present, or active, within a physical space. We see context information, particularly location and presence, as a key trigger for the reconfiguration of services and resources in order to adapt user access rights and protect user privacy. We adopt a user-centred approach, in which users are afforded the opportunity to define their own policies, which embody their preferences for actions to take place based on changes in the user set present in their vicinity. User policies could address the presence of specific individuals, individuals with specified roles, or individuals possessing specified access rights. For example, in an office environment, a user may wish to ensure that guests are never given the opportunity to view commercially sensitive information.

The paper is structured as follows. First, we briefly review previous work on access control approaches in ubiquitous computing environments, focussing in particular on approaches for automatic detection and resolution of conflicts that may arise between access policies that users or administrators wishes to deploy simultaneously. We then describe the operation of the M-Zones Access Control (MAC) process, which, through use of XACML, provides for adaptive attribute-based access control with conflict detection/resolution, and is integrated

White M., Jennings B. and van der Meer S. (2005).
USER-CENTRIC ADAPTIVE ACCESS CONTROL AND RESOURCE CONFIGURATION FOR UBIQUITOUS COMPUTING ENVIRONMENTS.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 349-354
DOI: 10.5220/0002578103490354

with a policy-based management system to allow for the reconfiguration of services and resources to protect user privacy. We outline the implementation of the MAC process in a test bed emulating an office-based environment, and describe its operation in the context of a specific use case scenario. Finally, we summarise the benefits of the proposed approach.

2 ACCESS CONTROL IN UBIQUITOUS COMPUTING ENVIRONMENTS

Ubiquitous computing environments are generally understood to contain hardware resources providing a wider range of command and information delivery interfaces than provided by traditional computer consoles. The nature of these resources can mean that managing them in cases where users can simultaneously request access to them can be a complex task. Typically, only a single user can control operation of resources such as projectors at any given time and access to them must be controlled in a manner consistent with system policies and user preferences. Indeed, the user-centred focus of ubiquitous computing environments suggests that users should, within limits, have the ability to adjust access rights for themselves and others as they desire (Kagal et al. 2001), thereby realising a more dynamic access control system, that adapts to changing user needs.

Many systems employ role-based access control, in which users are assigned one or more roles, typically mapping to functions in an organisational hierarchy, with each role being associated with a defined profile of access permissions. Whilst offering flexibility and relatively low management overhead, it can be argued that this approach does not provide the fine-grained control required in many ubiquitous computing applications. Additionally, it is difficult to apply it effectively in cases where resource competition conflicts occur when access is required by identical organisational roles. To overcome these drawbacks, access control decisions can instead be based on specific attributes associated with a user, rather than on the user's identity or assigned role(s). This approach is at the core of the eXtensible Access Control Markup Language (XACML) (Godik and Moses 2003), which has been used to provide access control in systems such as Cardea (Lepro 2003).

Ideally, access control solutions would automatically adjust access rights based on the changing context in which users are requesting access to resources – this is known as context-based access control. From an access control perspective, user location, presence of a group of users in a location, the relationship between the users within such groups, and the particular activities users are engaged in, are probably of most relevance. Basing access control decisions on this kind of information is an important research topic presently; for example, Corradi at al. (2004) have developed UbiCOSM, a context-centric access control middleware that assigns access rights taking into account context, user profiles and system/user-level authorisation policies. Also, Sampemane at al. (2002) address aspects of context-based access control for ubiquitous computing environments, describing a system that changes access rights depending on the set of users and the activity being undertaken in a physical space.

Based on the above observations we conclude that access control systems for ubiquitous computing environments should be:

- *user-centric:* allow users the freedom to adjust access rights as their needs evolve;
- *attribute-based:* access control decisions should be based on evaluation of appropriate user attributes, not rigidly on their identity or pre-assigned role(s);
- *context-driven:* access rights should be dynamically assigned based on analysis of context information provided by the environment.

To realise these properties we employ a policy-based management approach in which access rights assignment, as well as resource configuration based on access rights, is achieved through context-driven analysis of system and user-specific policies. In such a system, an important consideration is how to detect and resolve conflicts that are likely to occur in certain operational contexts between user and system policies and/or between policies specified by different users. Before introducing our access control process, we briefly discuss recent work on policy conflict detection and resolution, identifying the most appropriate approach for our system.

2.1 Policy Conflict Detection and Resolution

Conflicts between policies occur if, at any given time, the behaviour mandated by those policies cannot be simultaneously expedited. For example, one policy may oblige a user to take a certain course of action at a certain time, whilst another policy may forbid the user, at all times, access to a resource required to take this course of action. Policy conflicts are often broadly classified as being *static*

or *dynamic* (Lupu and Sloman 1999, Dunlop et al. 2003); static conflicts can be detected in advance (at "compile time"), whereas dynamic conflicts are dependant on "run time" state and thus cannot be detected in advance.

Conflict detection involves the identification of actual or potential policy conflicts. Methods for automatic conflict detection focus on analysing all policies relating to particular subject/action/target tuples and identifying whether the there is a conflict between the modality of these policies. Policies are generally constructed to reflect the obligation, permission and prohibition modal operators of deontic logic, thus modality conflicts are exhibited by policy pairs that, for a given subject/action/target, indicate behaviour that is prohibited vs. allowed, obliged vs. prohibited or obliged vs. not obliged. Other kinds of conflict detection, in particular those relating to the semantics of the policies, are significantly more difficult to detect automatically.

Once potential policy conflicts have been detected, a decision must be made as to whether to seek to resolve the conflict, with this decision being application-specific, but typically related to the probability of occurrence of the identified conflict. Two approaches to conflict resolution are possible: revoke, re-specify and re-deploy offending policies, or let the system assign precedence levels that dictate which of the conflicting policies are actually invoked. The latter approach is more practical, and researchers have investigated/adopted numerous schemes for assigning policy preference, for example see (Lupu and Sloman 1999, Dunlop et al. 2003). For example, precedence can be assigned based on: specific policies overriding general policies; newer policies override older policies; policies specified by a higher authority overriding those specified by lower authorities; negative policies overriding positive policies and vice versa; or explicit assignment of policy weights to govern precedence. These schemes all have strengths and weaknesses, but it is agreed that none is suited for use in all application scenarios.

In our case, policies can be authored by individual users, as well as by administrators of systems. Sets of users in a physical space are likely to have policies with the potential to conflict with each other, and/or with system policies. We envisage policy conflict detection and resolution being performed every time the set of users in a space changes (as users enter/leave), or as the activities they are performing change (for example, a project meeting commences in a meeting room). We view the former as a form of static conflict detection and the latter more as dynamic conflict detection. We see conflict resolution based on higher authorities overriding lower authorities as the most appropriate scheme in our scenario: system policies are given precedence over user policies, and between users precedence is based on users' profile, including their current roles within the space. User roles are assigned in accordance with system policies and may change over time. For example, in a meeting context, a user may be assigned a speaker role when he/she is detected as standing on a podium, and system policies will dictate that speakers have control over the projector, lights and other resources. In this case that user's policies relating to, for example, lighting settings, will have precedence over those of other users.

3 M-ZONES ACCESS CONTROL (MAC) PROCESS

The M-Zones access control solution we propose has been realised in the context of a "Ubiquitous Management Architecture" (UMA) (Barrett et al. 2004), developed as part of the M-Zones research programme (M-Zones 2005); which approaches management of ubiquitous computing environments, specifically smart spaces, by introducing the concept of "Managed Zones" (M-Zones) corresponding to administrative domains encompassing one or more distinct smart spaces. The UMA adopts a policy-based management approach to facilitate intra- and inter- smart space management, with policy decision points (PDPs) organised in two levels, following the hierarchical approach described in (Ghamri-Doudane et al. 2004). The PDP at the upper (M-Zone) level is responsible for all high level policies relating to the administration of the smart spaces. At the lower (smart space) level PDPs and PEPs control the discovery and execution of services. Ongoing decision making relating to access rights occurs at the M-Zone level, with access rights being communicated to individual smart spaces in the form of access control lists, which are enforced by the local PDP and PEPs.

There are two other UMA components involved in access control: the Context Information Manager (CIM) and Personal Information Managers (PIMs). The CIM is responsible for gathering, aggregating and semantically enhancing context information subscribed to by the M-Zone PDP and notifying the M-Zone PDP when context changes occur. Each user has associated with him/her a PIM, which stores their user profiles, preferences and policies, and also acts as their interface to the system. Operation of the CIM and PIMs is described in (Barrett et al. 2004).

3.1 MAC Process Operation

The MAC process is responsible for reaching access control policy decisions and for collating relevant information from other UMA components needed to inform these decisions. Access rights are assigned based on policies relating to the smart spaces themselves – "system" policies, and "user" policies (retrieved from the PIMs of users currently present in the smart space), in response to context change events notified by the CIM.

The MAC process is realised via a XACML policy engine (Sun Microsystems 2005), which allows for reaching policy decisions on the basis of the values arbitrary user attributes – which in our case are stored in user PIMs, or values of attributes of the environment itself (as notified by CIM context events). As well as re-configuring access rights, the MAC process analyses whether user policies indicate that specified actions be requested as a result of the notified event. For example, users may desire that configuration actions be taken to protect their privacy in the presence of users unknown to them. Once the process has been completed PDPs at the smart space level are forwarded new access control lists to be enforced for users currently in the space, and configuration actions generated from analysis of user policies.

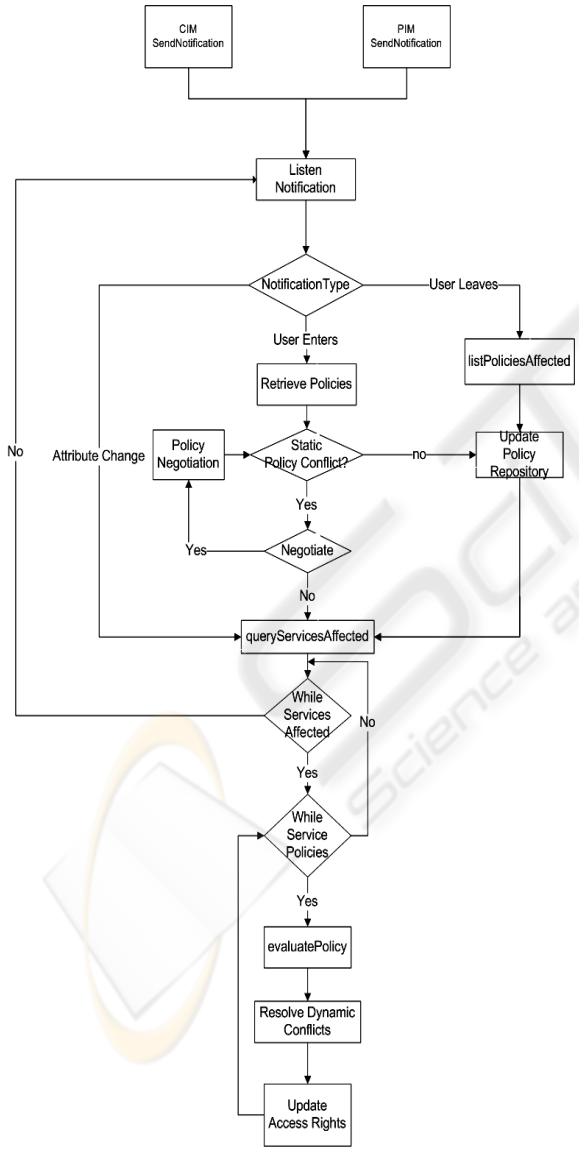

Figure 1: MAC Process Flow Chart

Figure 1 below outlines the flow concerning the access control decision mechanism implemented by the MAC process. The MAC process subscribes to the relevant components (PIM/CIM) in order to be notified should an event that requires access control decisions. Events that are typically monitored include the entry/exit of users or the changing of an attribute value that was considered as a condition of an active policy. When a user enters the smart space in question the MAC process retrieves the relevant user policies from the respective PIM. Static policy conflict detection is then carried out with respect to system and user policies in order to eliminate redundant policies and (re-)deploy the relevant policies. The MAC process then evaluates the respective access rights for the resources in question. This is achieved through iterating through the applicable policies governing the specific resources. Should the event the MAC process has been notified of concern the change in the value of a relevant attribute, this again necessitates the identification of relevant polices and iteration through the various polices in order to reconfigure the access rights concerned.

Central to the success of the MAC process in providing an adaptive access control is its ability to detect and resolve conflicts between various user and system policies. The MAC process extends the XACML policy engine's policy conflict/detection resolution facilities to realise configurable policy precedence schemes, by allowing specification of sets of attributes based on which precedence can be evaluated. Thus, how a scheme is implemented will be environment-specific, but will be based on appropriate attributes, for example security levels, or date/time. Our implementation targets an office environment (see §4.1), and ranks policies based on the resource in question, the policy author's organisation role, policy author's project affiliation, policy author's smart space role (for example, presenter) and the project context in which resource is being accessed.

4 IMPLEMENTATION AND CASE STUDY

Figure 2 below illustrates the test bed in which the MAC process has been deployed. For the test bed, the Ubiquitous Management Architecture has been partially deployed onto the TSSG/O_2 Home of the Future ubiquitous computing environment (TSSG/O_2 2005). Access to resources and services is managed through the policy-based management system described in (Ghamri-Doudane 2004), which is based on the COPS-SD protocol.

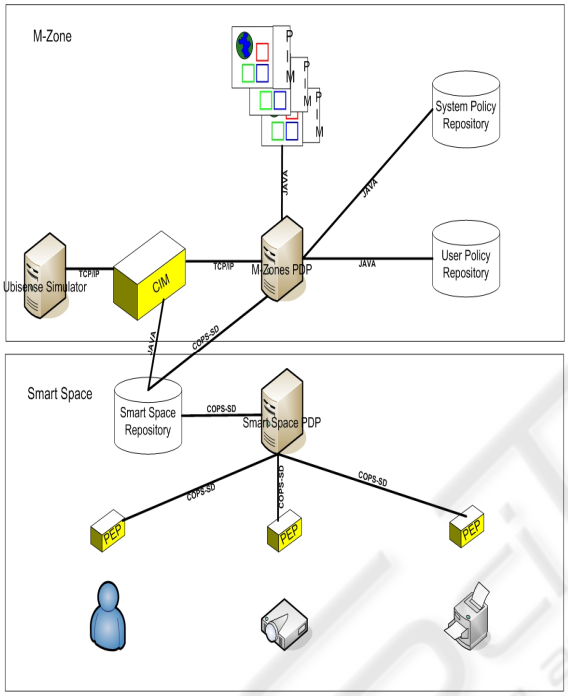

Figure 2: Test bed Architecture

The test bed consists of a number of PEPs controlling household devices by means of UPnP, Jini and proprietary approaches to service discovery and resource access. PEPs have COPS-SD wrappers to allow them communicate with their smart space PDP. Access to devices in the test-bed is ultimately controlled by the M-Zone PDP's MAC process, which uses COPS-SD to inform the smart space PDP of access rights to enforce. To test the operation of the MAC process a COPS-SD wrapped PEP was developed; this PEP controls a HP projector – the device used to realise the office-based use case scenario as described in §4.1.

The PIM has been implemented as a web service to host the user profiles. When a user initially enters an M-Zone they provide a link to their PIM, which will then be queried for the credentials required for authentication. The PIM also acts as a repository for user policies, including those dictating desired actions in the presence of other users. The PIM provides notifications to the M-Zone PDP if user attributes or policies are modified by a PIM, as, in many cases, these modifications will necessitate reconfiguration of access policies, or generation of new configuration action requests.

Context notifications, specifically user location and presence information are generated using the Ubisense simulator software (Ubisense Ltd. 2005), which has been used to model the movement of individuals along predefined paths through an office environment equipped with ultra-wideband (UWB) location detection. The current CIM implementation passes on context notifications as requested by the M-Zone PDP, expressing them in ontological terms understood by the MAC process. In the future the test-bed will be further developed to allow the CIM implementation gather information from a real Ubisense UWB location detection system, as opposed to simulated data.

4.1 Office-based Access Control Case Study

We now discuss a use case scenario in order to illustrate the MAC process functionality, in particular, how access rights adapt to the changing user set present and how policy conflicts are handled. The scenario concerns a meeting room owned by Company X, in which a projector service is used by meeting participants.

Bill is a project manager and Alice is an accountant; both are assigned to the M-Zones project, and are conducting a meeting together, in which Alice is presenting the project accounts using the projector service. Both have been previously authenticated and authorised regarding the various services, including the projector service, available in the meeting room smart space. Bill has defined a policy which permits all users to use the projector service at all times. On the other hand Alice has specified a policy that denies her access to the projector service should a non M-Zones project member enter the room. This is to prevent unauthorised users seeing sensitive financial information relating to the project, so, if a guest enters the meeting room her access to the project is revoked, thus her presentation will be immediately removed from the projector. Clearly, these policies conflict with each other. However the precedence scheme for the meeting room is configured such that policies specified by a user currently presenting and thus a presenter smart space role and accountant organisational role always have precedence over

policies specified by a user that has a manager organisational role and an audience smart space role, thus in this case Alice's policy is enforced. This contrasts with typical "Higher Authority Overrides Lower Authority" based approaches where policies specified by users higher in the company hierarchy have precedence over those specified by users lower in the hierarchy (which would favour Bill in our scenario).

Bob, a guest, now enters the meeting room. The MAC Process receives notification via the subscription/notification agreement it has with the CIM. The notification triggers the assignment of access rights to the new entrant based on system policies and any relevant policies of other users in the space. In this case the MAC Process examines the relevant policies governing access to the projector service. This leads to a conflict between Bill and Alice's policies. The precedence scheme employed favours Alice's policy as outlined above; thus her own access to the projector service is revoked, and the projector is blanked before Bill has a chance to see any potentially sensitive information.

5 SUMMARY

This paper has outlined an approach to access control in ubiquitous computing environments that realises adaptive, attribute based access control, based on analysis of both, system and user-defined policies. It harnesses context information relating to the user set present in a physical space and the context in which resources are accessed, as inputs into the access right configuration process. Furthermore, based on preferences specified in user policies, it supports automatic configure of resources in response to changes in the profile of this user set. Policy conflict detection and resolution is also addressed: a resolution scheme based on configurable assignment of policy precedence based on arbitrary attributes relating to both users and environmental context was described.

The approach allows the organisation administering the ubiquitous computing infra-structure to set policies governing default access rights associated with users, but also gives users themselves scope to dynamically modify access rights of others and to ensure that the environment is automatically configured to ensure their privacy is protected. Management functionality is therefore partially the responsibility of the user, resulting in a more user-centric system that adapts to changing user needs, but not in a manner that violates system-wide policies.

REFERENCES

Barrett K., Carroll R., Osmani V. and van der Meer S. (2004), User-centric Management of Ubiquitous Environments: Challenges and Initial Solutions, in Wade V. (ed), Proc. 2nd Int'l Workshop on Managing Ubiquitous Communications and Services (MUCS 2004), Dublin, Ireland, December 2004;

Corradi A., Montanari R. and Tibaldi D. (2004), Context-based Access Control Management in Ubiquitous Environments, Proc. Third IEEE Int. Symp. on Network Computing and Applications (NCA'04), 253-260;

Dunlop N., Indulska J. and Raymond K. (2003), Methods for Conflict Resolution in Policy-Based Management Systems, in Proc. 7th IEEE International Enterprise Distributed Object Computing Conference (EDOC'2003), Brisbane, Sept 2003, 98-109;

Ghamri-Doudane S., van der Meer S., O'Connor R., Ghamri-Doudane Y. and Agoulmine N. (2004), Resources Discovery and Management Using Policies in Smart Spaces, in Proc. Workshop of the 11th HP OpenView University Association (HPOVUA 2004), Paris, June 2004;

Godik S., Moses T. (eds.) (2003), eXtensible Access Control Markup Language (XACML) Version 1.0, OASIS Standard, available (15/2/2005): http://www.oasis-open.org;

Kagal L., Finin T. and Joshi A. (2001), Trust-Based Security in Pervasive Computing Environments, IEEE Computer, **34**(12):154-157;

Lepro R. (2003), Cardea: Dynamic Access Control in Distributed Systems, NAS Technical Report NAS-03-020;

Lupu E. C. and Sloman M. (1999), Conflicts in Policy-based Distributed Systems Management, IEEE Trans. on Software Engineering, **25**(6):852-868;

M-Zones research programme, information available (15/2/2005): http://www.m-zones.org;

Sampemane G., Naldurg P. and Campbell R. (2002), Access Control for Active Spaces, Proc. 18th Annual Computer Security Applications Conference, 343-352;

Sun Microsystems (2005), SunXACML Implementation, information available (15/2/2005): http://sunxacml.sourceforge.net;

TSSG/O_2 Home of the Future Smart Home Demonstration, information available (15/2/2005): http://www.o2home.com;

Ubisense Ltd. (2005), Ubisense Product Description: Simulator Module, Information available (15/2/2005): http://ubisense.net/Software/Simulate%20environments.htm.