

# AN ACCESS CONTROL FRAMEWORK FOR PERVASIVE COMPUTING ENVIRONMENTS

Sarah Javanmardi, Hadi Hemmati, Rasool Jalili

Network Security Center,  
Department of Computer Engineering,  
Sharif University of Technology, Tehran, Iran

**Abstract.** *The explosive evolution of pervasive computing environments presents several new challenges such as smart spaces, invisibility, localized scalability, and masking uneven conditioning. One of the key challenges in such environments is how to manage security and access control. A suitable approach for managing access control in pervasive computing environments should consider different features of such environment which are not supported by traditional access control mechanisms. The features impose the access control mechanisms to satisfy especial requirements. In this paper, we propose an access control framework supporting different requirements of pervasive computing technology. The framework is capable of providing solutions to the requirements by some functionalities such as decision making based on user intent and in an invisible manner and re-authorization of granted services based on context changes.*

**KEYWORDS**—Access Control, Security Framework, Pervasive Computing.

## 1. Introduction

With the event of pervasive computing, transparent functionalities of pervasive applications integrated to homes and communities make living easier and more comfortable for people. In comparison with distributed systems and mobile computing, Pervasive computing environments (PCE) introduces new goals such as *effective use of smart spaces; invisibility, localized scalability and masking uneven conditioning* that make the environment provide some properties to reach them [1]. Different researches have identified different properties for pervasive computing and the most important ones are *user intent, context-awareness, dynamicity, automatic evolution, interoperability, and adaptability* [2, 3]. These new properties particularly in the domain of security introduce new challenges that were not addressed in traditional and static security approaches [4]. When considering the security challenges of many PCE applications, access control often emerges as a central element in the design of the whole security system [5]. Access control

is the mechanism that allows service providers to define, manage and enforce access control policies applicable to each service [6, 7] and to assure that users can access only and all services authorized to them.

To harmonize access control applications with other PCE applications, several research address PCE properties and according to them, tried to state access control requirements [8]. Some has targeted challenges for specific applications and has tried to find a solution for them [9, 10]. Another research trend is adding the access control requirements to the general access control mechanism. Context-awareness has been addressed in several works with different aspects. Sandhu et al. [11] has introduced a socio-technical view of access control models and by defining the new required entities and their relationship in PCE, has provide a border between classical access control models and the ones needed in PCE. Some other work focuses on access control policies and attempt to remove traditional static policies by adding the attribute of context-awareness to them and try to make access control decisions be dynamic with the change of context [6]. Kouadri et al. [12] aims at presenting a new model for specifying context-based policies based on contextual graphs and rely on the contextual information collected from the system and user's environment. Invisibility and User intent are other aspects of PCE properties, which aims at helping users to have hidden and non-intrusive authorizations which are less burdensome while their preferences should be respected during the access control steps [1, 2, 3].

This work is aimed at introducing PCE access control requirements; especially we are concerned with context-awareness, user intent and invisibility. Based on these requirements, we present a framework supporting service discovery and access control schemes in an efficient way. The remainder of this paper proceeds as follows: Section 2 describes some related works. Section 3 states fundamental access control requirements for pervasive computing. In section 4, our framework is presented and the related procedure is explained and section 5 underlines some concluding regards and future research lines.

## 2. Related Works

Several access control frameworks for PCE have been presented that attempt to apply solutions for aforementioned challenges. Convington et al. in [6] has provided a context-aware framework (CASA) based on RBAC. There are some extensions on CASA like ECASA [13] that try to make CASA more efficient through detecting those context changes that are effective to the authorization. In [14], a conceptual architecture for access control is proposed which is based on automatic distributed acquisition and processing of knowledge. In the architecture agents are responsible to gather information about users and devices for better decision making. In this work we present a framework capable of addressing new requirements.

## 3. Access Control Requirements for PCEs

The shift from centralized to distributed systems and mobile computing poses new security requirements, especially in the domain of access control such as interoperability, scalability, usability, privacy and trust management [4,8]. Pervasive computing as the next generation of mobile computing has brought new security requirements which are specific to it. In this Section we state these requirements and address them in a scenario.

- **Providing a context-aware access control mechanism:** High degree of dynamicity in pervasive environments introduces new challenges to access control mechanisms. Any change in the context such as user profile, service profile or environmental context can be important to the authorization procedure [6, 13]. It means that, granting or revoking a request is highly dependent on the current context and effective context changes should be considered for re-authorizing the request.
- **Providing high degree of invisibility:** The automation of daily activities is one of the aims of pervasive computing technology. Therefore it is ideal to provide entirely human-free interactions [3]. Complete disappearance of pervasive computing technology should be considered in access control applications. For authorizing an access for a service, it is necessary to gather different credentials of the user. Therefore the framework should try to hide the existence of real interactions that also will result in less user distraction too.
- **Respecting user intents in the access control process:** Pervasive Computing technology forces the majority of interactions between users and the environment to be automatic. Hence user's prefer-

ences should be considered to be satisfied. Users in such environments can provide some light services to other users and authentication of the services should consider the preferences of the user. Besides, each user should be able to define his own service policies and for using these services, the predefined conditions should be satisfied. Reserving the privacy of user's data according to the predefined user preferences and considering user profile changes in authorization can be other aspects of respecting user's intent.

### 3.1 Smart Library Scenario

In this section, a smart library application in a university in PCE is illustrated in order to help to demonstrate some access control requirements for such applications in PCE.

Searching books in large libraries can be a difficult task for novice library users. Smart library provides a map-based guidance to books on user's PDA by uploading software called Lib-Guider. Consider the following example: A user enters the library; this is automatically sensed by the particular reading room. The server in the library uploads Lib-Guider on the user's PDA. Besides, the user can take a radio based identification device (an RFID-reader) that has wireless access to the local network using Bluetooth. When a user walks out with one or more hard-copies of books which have RFID tags, the books will be added to his basket by using a RFID reader embedded in the departure gate. In the library, users who are members of the library can share the e-books and e-documents on their PDAs with each other through wireless access to the local network using Bluetooth. In the scenario some points should be highlighted based on the access control requirements:

The Access Control framework should be able to make decision based on the current context including:

- User Profile, for example when just one copy of a book is left, there should be a policy that let those students that are urgent need to borrow that book.
- Service Context, consider the case when a user can access Lib-Guiders only when they are not many people in the library and the server is not overloaded by other tasks.
- Environmental Context, for instance, users cannot have access to registration services during weekends.
- The Access Control system should be invisible from the user consciousness, consider the case when a user enters the library the user profile from his PDA will be transferred to the server of reading room and after evaluating the user based on the

profile information, different services will be granted to him while all of these processes is hidden from the eyes of the user.

- The Access control system should consider user intent. By accessing the user preferences or logging the behavior patterns of the user, the access control system can choose an authorization path for user evaluation which is more convenient to him. For example if in the contextual graph of the policy [12], there are more than one authorization paths that the system can choose the one that has less explicit data gathering from the user.

## 4. Access Control Framework for PCE Services

In a typical PCE, by exploiting some wired hardware infrastructures or computer servers called surrogates, we can amplify the capabilities of the resource-limited mobile devices and let the users be able to run their application satisfactorily [1,3]. Our access control framework for PCE services consists of two main parts. The first part is installed on some of powerful surrogates, which we call them Central Servers and the other part is installed on the all user's devices. Central servers are distributed in PCE and divide it to some zones, where in each zone one Central Server is responsible for managing the zone service access and provision. Surrogates which are not Central Servers can provide services but they don't have Access Control components. Each user device can provide services too, and additionally have the user part of the Access Control Framework.

In the following part, we introduce our Access Control framework and then we review the access process based on the components in the framework.

### 4.1 Access Control Framework Architecture For PCE Services

Fig.1 shows the PCE services access control framework (FW). The framework consists of two main parts, the Central Server and the User Device. The Central Server side of FW which we will call it server application in the rest of the paper is responsible to authorize users and provide different services to the authorized users. The User Device side, which we will call it user application is responsible to manage the user's activities in the PCE.

#### 4.1.1 The User Application

In PCEs users are mobile and besides the Central server, a User Application can provide some light services to others. Users can send requests for different services to the Central Server. To give a broader

view, in this part we will explain basic components of a User Application:

- **User Profile:** In PCE any information about the user can be a part of the user profile. In Fig.1, this repository contains some information about the user which will be updated by any changes. Other specific information about the user like his preferences is also kept here.
- **User Preference:** According to the privacy of each piece of information, a user can define a degree of privacy for each, or in a better way, for a group of similar information [15]. For example one can have a group of highly personal information and does not let any information from this group be sent while requesting for routine services. Based on the type of the requested service from a user, when the provider of the service asks for some kind of information to evaluate the user, the User Application can determine which sets of information can be sent based on the user preferences.
- **User Handler:** The User Handler the interface between the User Application and the Central Server. It has access to information in different repositories like the User Profile and Service Policy and handles the communications.
- **Service Provider:** PCE is full of service providers that present different services to users. As mentioned before, a User Application can provide some light services based on their Client-thickness [1]. When the Central Server decides to grant an access to a user, Service Provider is responsible to provide the service. Service Provider contains the Service Policy repository.
  - **Service Policy:** Since always it is the provider of the service that defines the conditions under which the access should be granted, there is a repository in this component for storing such rules and can be updated by the provider when necessary.

#### 4.1.2 The Central Server

There are some surrogates in each local zone in PCE and by installing different applications they can play different roles like Compute Servers or data-staging servers [3]. The Central Server here is one of these surrogates that according to the components that are explained here, different applications should be installed on it. When a mobile user enters a zone, the local Central Server tries to provide different services to him based on the requests. Most of the components that are necessary for decision-making are embedded in the Central Server.

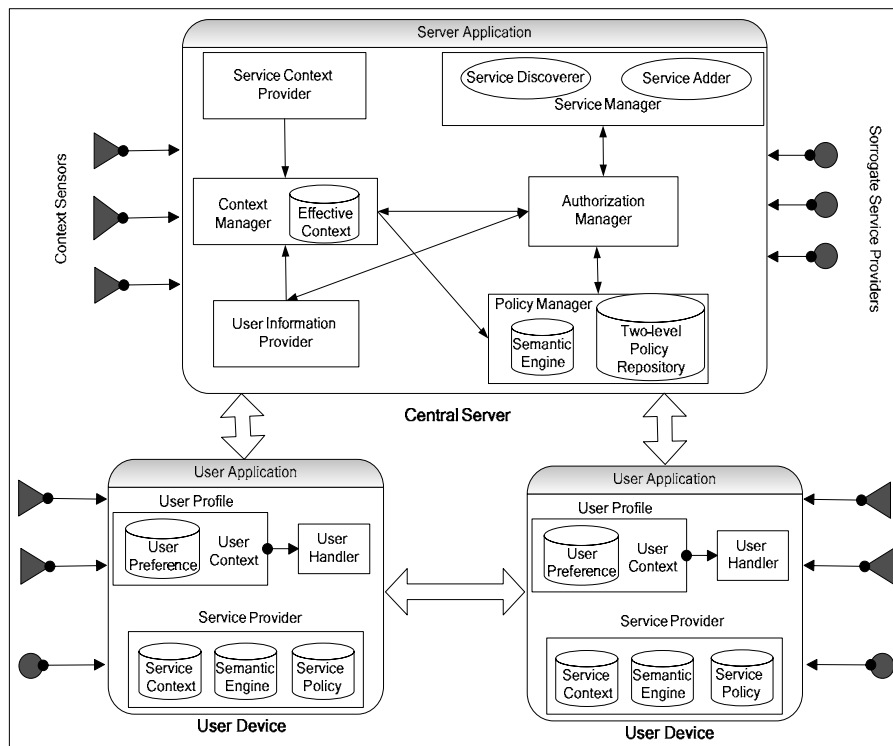


Fig. 1. The PCE Services Access Control Framework.

In this section we try to define different components of the Central Server and just those functionalities that are related to our main focus, access control.

- **Service Manager:** Service Manager deals with the incoming requests for services and has two main components:
  - **Service Adder:** Service Adder is a component that adds a new service to the Central Server's Environment Services List. A service can be added to the Central Server in two different ways. First, when the Central Server or other local surrogates of the zone decide to add a new one. In other words the Central Server should be able to see different services and service providers in the zone to be able to manage the requests, it means that the Central Server is aware of the range of access in this zone and can assign different requests to related services effectively. Second, when a new user introduces the services that he provides to the Central Server, which means that, the user is ready to present his services to other User Applications through the Central Server that later we explain it in more detail. While adding a new service it is very important to add its properties (service context) to the repository called Service Context Provider. Besides, when a mobile user decides to share some services, his service profile containing the service context will be transferred to the Central

Server. In this way the Central Server is aware of different services provided by different User Applications and can transfer each service request to a proper User Applications.

- **Service Discoverer:** The Service Discoverer finds a suitable service or a set of services according to the current request received from a User Application. The discovered set of services can be sorted according to the user intent. It means that if the user shares some of his profile information with the Central Server, the Service Discoverer can sort services based on these preferences. In next section we will show that when and how this sharing can be done.
- **Context Manager:** The Context Manager component collects and preserves context information. Context information includes any information about the users, the services and the environment. Therefore, User profile, service profile and environmental context are the information that Context Manager deals with them. Context Manger should be aware of any changes in the context, for example after granting an access, it should transfer changes of some important context called Effective Context to the Authorizer component. Effective context records [10,13] are those context that the value of them is necessary in decision-making phase and any changes in their value can result in another decision-making

and may be another result even during using a service by a User Application. The entire effective context is stored in a repository called Effective Context.

- **Effective Context:** The Effective Context repository contains a subset of context in the form of records. Each record is a triple tuple. These tuples contains some information about the context under which a request is made like the current user profile information, properties of the requested service and the context of the environment. Fig.2 shows a possible definition for a triple tuple where the UserProfile, SrvContext and EnvContext are information about the user, the requested service and the environment in which the request is made. The terminals which are not mentioned here are fine-granular pieces of context like temperature, location [16]. As it is mentioned in some other work [10,13], to satisfy context awareness it is not necessary to store all the context but we need those subset of context that the decision was made on them.

$$\begin{aligned}
 &Triple\ Tuple = (UserCond, SrvCond, EnvCond) \\
 &UserCond \rightarrow UserCond \wedge UserCond \mid \\
 &\quad Userprofile\ Operator\ Const \\
 &\quad \mid Userprofile\ Operator\ Userprofile \\
 &SrvCond \rightarrow Srv\ Cond \wedge SrvCond \\
 &\quad \mid SrvContext\ Operator\ Const \\
 &\quad \mid SrvContext\ Operator\ SrvContext \\
 &EnvCond \rightarrow EnvCond \wedge EnvCond \\
 &\quad \mid EnvContext\ Operator\ Const \\
 &\quad \mid EnvContext\ Operator\ EnvContext \\
 &Operator \rightarrow <|>|\leq|\geq|\neq|\equiv|in|not\ in
 \end{aligned}$$

Fig. 2. The Formal Presentation of Triple Tuple.

For example in the Smart Library scenario the related policy says that a user can access to the e-book on the PDA of other user only if he is a member of the library and while both of them are in the library. Now consider the case that user A and user B are in the library and user A as a member wants to access an e-book on the PDA of user B, the access is granted and only the changes in their locations or the membership situation of user A can result in another decision making (Re-Authorization) and it is not necessary to save other context like the temperature of the environment. Fig.3 shows the record that is kept after granting the access to user A.

$$\begin{aligned}
 &(Requester.\ Location == Smart-Library \wedge \\
 &Service-Provider.\ Location == Smart-Library), \\
 &(Requester's.\ Membership == valid), ()
 \end{aligned}$$

Fig. 3. A Sample Effective Context Record.

- **User Information Provider:** This component is responsible for handling different communications between the User Application and the Central Server or between two User Applications. For example, the Context Manager receives user profile information from this component.
- **Context Sensors:** There are many Context Sensors spread in PCE that according to the application, they sense and gather environmental context and send any changes to the Context Manager.
- **Service Context Provider:** This component gather changes in the context and sends the changes to the Context Manager.
- **Authorization Manager:** The Authorization Manager receives different access requests for different services and based on the current context and policies, decides whether to grant an access or not. Re-authorization of granted request is another duty of this component.

$$\begin{aligned}
 &PolicyRule = (PreCond, ReqInfo, Result) \\
 &PreCond \rightarrow Cond \\
 &ReqInfo \rightarrow Cond \\
 &Cond \rightarrow PreCond \wedge PreCond \\
 &\quad \mid Context\ Operator\ Const \\
 &\quad \mid Context\ Operator\ Context \\
 &Result \rightarrow Grant \mid Revoke \\
 &Operator \rightarrow <|>|\leq|\geq|\neq|\equiv|in|not\ in
 \end{aligned}$$

Fig. 4. The Formal Presentation of a Two-level Policy Rule.

- **Policy Manager:** Policy Manager has to add a new policy and delete one when necessary. Adding a policy is done when the Central Server finds a new service on other servers or on User Applications in the same zone. In other words, when the Central Server chooses a service that belongs to another Service Provider, the policies of that service will be transferred from that Service Provider to the Policy Repository in the Central Server and after using the service the added policy will be removed from the repository. The Policy manager is also responsible to detect different conflicts between fired rules and resolve them. A conflict occurs when based on the current context, more than one policy is fired while some of them result in granting the access and others result in revoking the access, some works like [17,18] discussed conflict detection and resolution based on the context changes.
- **Two-Level-Policy repository:** This repository contains two-level policy rules, according to the cur-

rent context. The formal presentation of a policy rule is shown in Fig.4. According to the current context, some rules are fired from the repository. The idea of multi-level policy comes from the user evaluation in an incremental manner. In other words to respect the user intent in Access Control system in PCE, it is important that the system tries to gather information in a sequence that is more convenient for the user. In traditional Access Control mechanisms according to each condition on the user information in a policy, the user is responsible to satisfy the condition explicitly, for example by entering a password. While in PCE, since the system is aware of the profile of the user there is no need for frequent questions for gathering required information. In our scenario when a user enters to Smart Library, his user profile is transferred to the server and according to his requests some conditions will be satisfied or not satisfied in an invisible manner, it means that some parts of a related policy are satisfied that means policy is fired in the first level. But for example when the request is about accessing to an e-document that contains highly sensitive budget information about a special project, it is necessary to ask the user whether he is a member of that project to satisfy the last condition and if he proves his membership the policy will be fired in the second level, Fig5. Besides, the system cannot expect users to share all of their profiles when entering the library and for respecting their privacy, they should be able to choose which part of their profile can be transferred and which can not.

|   |
|---|
| (User. Membership == valid $\wedge$<br>Membership. Class == First), |
| (Project-A. Membership == valid),<br>Grant                          |

Fig. 5. The Fired Policy Rule in its two levels.

## 4.2 The Access Control Process

In this section, it is explained that how our framework deals with the incoming service requests and manages the access control procedure. The procedure is divided into five phases. The detail of each phase is as following:

**I. Initiation:** When a user enters a zone and User Handler sends the User Profile and sends information about its services to the Service Context Provider.

**II. User Request and Service Discovery:** After that, a user can send its requests to the Central Server. After receiving a request, Service Discoverer starts to find a set of suitable services. The services can be on the Central Server or other Service Providers in the zone. After finding some services, Service Discoverer sorts the services according to the user intent and makes a list. If the service is chosen from another service provider, context of the service and service policies are sent to the

Service Context Provider and Policy repository. Now Service manager sends a message to the Authorization Manager for example in the form of ( $\{\text{Service\_Id}\}$ ,  $\text{User\_Id}$ ) to start decision making. If no service is found, Service Discoverer passes the request to the nearest Central Server.

**III. Authorization:** Authorization phase contains the following sub-phases.

- **Firing of policy rules in the first level:** For making decision, Authorization Manager needs some policies. After sending a request for some policies based on the current context to the Policy Manager, policies will be fired in the first level. As mentioned in last section, if now there is a need for some extra information about the user, Information Provider component gathers the requested information from the User Handler.

**IV. Making final decision:** The final decision of Authorization Manager should be sent to the Service Manager. If it is grant, through the service provider which can be the Central Server, other server in the zone or a client, the service will be provided to the service requester. But if the access is denied Service Manager checks the list to see if another service exists, and the whole process starts again until the list becomes empty.

**V. Re-authorization:** As mentioned before, the final decision of Authorization Manager, is based on the current context. It means that any changes in the current context can affect the current decision. For handling the changes, there is a need to re-authorize the request. Therefore after granting a request, Authorization Manager makes a triple tuple including related context and adds it to the Effective Context repository. From now on, for each granted request, any changes in the tuple will result in a re-authorization request from Context Manager to the Authorization Manager. The result of re-authorization can be deny that means the user will be prevented from using the service any more.

## 5. Conclusion and Future work

In this paper we presented a framework for access control in PCE services. We focused on satisfying some specific requirements of access control which are related to the environment's Relation-Awareness or in a larger scale environment's Semantic-Awareness. This is a new and interesting challenge which is rarely addressed in PCE security works. Some works [7, 15, and 19] especially in the area of semantic web described the necessity of semantic-awareness in highly dynamic environments and have presented some access control models for supporting them.

In our future work we try to add semantic-awareness to our framework by presenting a suitable access control model for pervasive computing environment.

## 6. Acknowledgements

In this part it is necessary to thanks from Prof.Nahid Shahmehri and Dr.Claudiu Duma because of their precious help for presenting this work.

## 7. References

- [1] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges", *IEEE Personal Communications*, August, 2001.
- [2] M. Weiser, "The computer for the twenty-first century", *Scientific American*, pp. 94–104, September 1991.
- [3] D. Garlan, D.P. Siewiorek, A. Smailagic, and Steenkiste, "Project Aura: toward distraction-free pervasive computing", *Pervasive Computing IEEE*, Volume 1, pp. 22 – 31, Issue 2, April-June 2002.
- [4] C. Yeun, E. Lua, J. Crowcroft, "Security for Emerging ubiquitous networks", *IEEE International Conference one-Vehicular Technology*, Volume 2, pp. 1242-1248, 25-28 Sep. 2005
- [5] T. Woo, and S. Lam, "Designing a distributed authorization Service", *In Proc. of IEEE INFOCOM*, volume 243 of *LNCS*, pp 227–245, Springer-Verlag, 1998.
- [6] M.J. Covington, P. Fogla, Z. Zhan, and M. Ahamad, "A Context-Aware Security Architecture for Emerging Applications", *In Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, Las Vegas, Nevada, USA, December, 2002.
- [7] L. Qin, V. Atluri, "Concept-level Access Control for the Semantic Web", *ACM Workshop on XML Security, Fairfax, VA, USA*, pp.94-103, October 31, 2003.
- [8] J. Wang, Y. Yang, and W. Yurcik, "Secure Smart Environments: Security Requirements, Challenges and Experiences in Pervasive Computing", *NSF Infrastructure Experience 2005, NSF/CISE/CNS Pervasive Computing Infrastructure Experience Workshop*, Siebel Center for Computing Science University of Illinois at Urbana-Champaign, July 27, 2005.
- [9] V. Standford, "Pervasive Health Care Applications Face Tough Security Challenges", *IEEE Pervasive Computing*, Volume 1, Issue 2, pp.8-12, April-June, 2002.
- [10] K. Takata, J. Ma, and B.O. Apduhan, "A Context Based Architecture for Ubiquitous Kid's Safety Care Using Space-oriented Model", *In proceeding of 11<sup>th</sup> International Conference on Parallel and Distributed Systems*, Volume 1, pp 384-390, 20-22 July 2005.
- [11] R.K. Thomas, and R. Sandhu, "Models, Protocols, and Architecture for Secure Pervasive Computing: Challenges and Research Directions", *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, Percom 2004.
- [12] G.K. Mostefaoui, and P. Brezillon, "Modeling Context-Based Security Policies with Contextual Graphs", *In Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004.
- [13] R. Zakeri, and M. Niamanesh, "Enhanced Context Aware Security Architecture", *Proceeding of the First International Conference on Modeling, Simulation and Applied Optimization*, Sharje, U.A.E. Februry 1-3, 2005.
- [14] 10. A. Seleznyov, S. Hails, "An Access Control Model Based on Distributed Knowledge Management", *18th International Conference on Advanced Information Networking and Applications (AINA 04)*, 2004.
- [15] U. Hengartner, P. Steenkiste, "Exploiting Information Relationships for Access Control", *In proceeding of third IEEE International Conference on Pervasive Computing and Communications*, Percom 2005, Kauai, Island HI, March 2005, pp 296-278.
- [16] H. Shen, F. Hong, "A Context-Aware Role-Based Access Control Model for Web Services", *EE International Conference one-Business Engineering*, pp. 220-223, 12-18 Oct. 2005.
- [17] E. Syukur, S.W. Loke, and P. Stanski, "Methods for Policy Conflict Detection and Resolution in Pervasive Computing Environments", *In Policy Management for Web workshop in conjunction with WWW2005 Conference*, Chiba, Japan, 10-14 May 2005.
- [18] H. Kamoda, M. Yamaoka, S. Matsuda, K. Broda, and M. Sloman, "Policy Conflict Analysis Using Free Variable Tableaux for Access Control in Web Services Environments", *Policy Management for the Web, A WWW2005 Workshop 14th International World Wide Web Conference*, 10 May 2005, Chiba, Japan, pp.5-12, May, 2005.
- [19] M.I. Yague, A. Mana, J. Lopez "Applying the Semantic Web Layers to Access Control", *In proceeding of 14<sup>th</sup> IEEE International Workshop on Database and Expert Systems Applications*, Percom 2005, Kauai, Island HI, March 2005, pp 622-626, 1-5 September, 2003.