Enhanced Context-Aware Role-Based Access Control Framework for Pervasive Environment

Tawfiq S. Barhoom[1], Mohammed O Al-akhras[2]

[1] *AssociateProffessor at Islamic University Gaza & Gaza, Palestine, tbarhoom@iugaza.edu.ps*
[2]*Resaercher at Palcore Microsystems Co. & Gaza, Palestine, md.alakhras@palcore.com*

Abstract— Utilization of contextual information considered very useful for improving access decision making process against systems resources, to be more effective in providing authorized service for a large number of end users.We selected model makes decisions based on context information sensed and collected from user environment. Then we enhanced context utilization and framework performance based on theoretical idea previously published [14], through studying the process of making decision based on context information validity. We focused on enhancing the distributing and management process of context information over users by using the proxy, which works as observer to enforce policy for short term context information. In case of any change, breaks access control policy rules, the proxy on user device will automatically send revocation/grant request based on change made for context information related to the user in his local environment. After the change made to context information listed within the available policy rules, the proxy will re-evaluate it on user device, and utilize available resources on the device, then grant or revoke permissions, finally will update the web service to be up-to-date. Such enhancement will highly increase system responsiveness and enhance authorization for end users..

Index Terms— context aware, role base access control, performance, grant, revocation.

I INTRODUCTION

Pervasive environment means that processing of information becomes integrated with everyday life activities, where computers involved in human life will not stop them to practice daily life, unlike desktop computers which enforce users to stop any other daily life activity to use them.

The emergence of pervasive environment or ubiquitous computing as coined by M. Wesier [1]. The surrounding ambience becomes smarter, our actions and existent become noticed and measured by computers and sensors which provide computer applications with information about us, Such environment poses new concerns that should be taken into account, one of the major concerns is security where users resources in such environment is vulnerable for unauthorized access, data and services confidentiality and integrity is now more under risk, also the opportunity to quick join for unknown users to the networks becomes larger.

One of the accompanied concepts of pervasive computing is the use of context awareness, emergence of context aware applications and services where an increasing demand for such applications. Context aware applications utilize implicit information to adapt system or application behavior. Researchers produced several definitions for context, B. Schilit [2] referred to as location, people interact with and objects and any change to these objects. Later B. Schilit [3] added to the definition new parameters like network connectivity, communication costs, bandwidth, resources or social situation.

While aspects and parameters for context cannot be enumerated due to situation change, so as a consequence A. Dey [15] defined context as follows: "any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant or the interaction between a user and an application, including the user and applications themselves".

As we stated before security is a major concern, especially is how to control access to resources and services which defined as access control mechanisms which defined as authorization.

One of the recently wide spread techniques is Role Based Access Control (RBAC), this mechanism which conforms companies hierarchical structure which makes it more desirable. Advancement in pervasive computing makes traditional RBAC, A. Dey [15] issued by National Institute Standards and Technology (NIST) has limitations, where an extension that take context information into account becomes an urgent need.

Many enhancements approaches adopted, where their model design based on:

1. Flat RBAC model for enhancement [4].
2. Constrained RBAC model for enhancement [4].
3. Non RBAC models: models which built and implemented access control based on mechanisms other than role base which could be Access Control List (ACL) or Discretionary Access Control (DAC).

The different contributions provided and described don't

provide complete image for framework enhancement and development for pervasive computing. For example context information validity, which means how much the measurement of context information, will be compliant with real value, so the decision made based on the value of specific context information will be valid or not. So as a result of the change of context information value will lead to a change in permission state.

Our work will introduce enhanced framework which will combine some techniques from exist models and presented another, the framework will enhance permission revocation and restoration based on context information validity.

II RELATED WORKS

Researchers use context awareness in their suggestions for new applications paradigm, also access control mechanisms is one of disciplines affected by this new paradigm, efforts made in improving access control mechanisms to be enriched with context aware, using our research perspective. We categorized efforts into RBAC based and Non-RBAC based, so we revise shortly Non-RBAC then RBAC based research in detailed manner.

A Non-RBAC Access Control Mechanisms

- Semantic based approach

Presented semantic context aware access control framework for Pervasive Computing Environment A. Dey [15], designed semantic context-aware policy model adopts ontology and rules to express context information and context-aware access control policies. Mainly focuses on non-organizational bodies and spontaneous interaction scenarios and on enhancing policy dynamicity and adaption, so the need to focus and solve access control for centralized business that has its own resources needed to be protected and controlled from unauthorized users and also considered as pervasive environment.

- Web service based approach

Such approach depends mainly on using web services to control access of users on objects and resources, Claudio Ardgana [8] presented architecture that utilizes web service to enforce policies for controlling access.

B RBAC based Access control Mechanisms

The first attempt to utilize RBAC in contextual manner done by M. Covington [9] provides a model to create and access information about homed residential and resources within the home called Generalized RBAC it depends on environmental roles in addition to traditional roles provided by RBAC about subject and object, RBAC notion of a role is generalized to capture the state of the environment. Presented a model for securing future applications, which uses generalized approach in handling context information, the model incorporate the notion of roles and environment roles with notion of subject roles. Where homes equipped with high technologies needs high knowledge about information security, which usually not found by most of home residents. Systems should make it very simple to define and manage security policies; also another challenge security mechanism should be usable and non-intrusive.

For example the days in the week split into two groups holidays role and weekdays role or based on locations upstairs, downstairs and guestroom. Permission Assignment in GRBAC done not only based on subject roles but also on active environmental roles.

Providing such roles for environmental state is considered heavy load on pervasive environment systems. Also I think applying such example to be generalized on more wide situations where large number of user than a home, which make such system more complicated to maintain high accuracy measurements. Also the authentication of user requests for access doesn't introduced, which considered very important module to authenticate users interacting with the system, and preventing any deception or any identity spoofing. Y. G. Kim [10] introduced a mechanism that uses state machine matrix SCM to grant or deny access privileges based on context.

The additions for traditional RBAC are:

- State Checking Matrix: handles context information like location, time, and others.
- State Checking agent: handles roles subset for each user.
- Context aware agent: handles permissions subset for each role.

G. Zhang [5] introduced a model as an extension to RBAC, where roles dynamically assigned to users depending on their context, where they used two state machine for each user one for representing assigned subset of roles and permission assignment hierarchy and both of the subset changed dynamically depending on context change where monitored and transferred to central authority by context agent.

Generating such state machine for each user in pervasive environment especially if the resources or services being targeted by user not exist on large servers or central computing power, in other words if the resource or the service exist of limited power devices with existence of large request form large number devices, this model will be at the expense of responsiveness of the system.

W. Han [11] introduced a formal model for context sensitive access control, where Reference Monitor responsible for making decision.

The proposed architecture doesn't concentrate on how:

- Integration or extending new context factors could be done.
- Also how much such a model could be convenient to pervasive environment?

J. Hwan [12] introduced formalized definition for managing dynamic roles and permissions assignment, also three major components responsible for three major operations as follows:

- Access Control Manager-ACM: responsible for processing access control request.
- Context aware User Assignment Manager-CAUAM: provides roles assignment based on context requirement defined in each table.
- Context aware Permission Assignment Manager-CAPAM: provides permission assignment based on context requirements, also provides personalized access control via utilizing user preferences information stored in user profile repository.

T. Devdatta [13] presented a model for context aware RBAC in pervasive computing applications, the model uses context information in role admission policies also how application behave when context condition fails to hold.

S. Sadat [14] introduced context aware access model based on RBAC for pervasive computing environment called CAP, the context information is grouped into long term (LTC) and short term (STC) context information. CAP introduced as a solution for RBAC drawbacks for handling unknown users that join pervasive network through using dynamic user role assignment and using context information for dynamic permission activation for roles as well , but this model has drawbacks as stated by authors in next step research contribution as follows S. Sadat [6]:

- Fetch many context information values to make a decision some of them may not be used, which causes overhead at execution time.
- Doesn't support role hierarchy.
- Uses limited combination of context conditions for assigning roles or activating permissions.

S. Sadat [6] Introduced enhanced version of CAP called iCAP, and tries to overcome some of the limitations in CAP, as noticed author's transformed describing context from a 4-tuple to triple<contexttype,contextrelater,value>,where previous published paper introduced as follows: <entity,contexttype,contextrelater,value>. Also the iCAP now handles roles hierarchy when assigning roles which includes inheritance.

Contribution added in iCAP is how to handle and assign role hierarchy which includes inheritance feature among roles, which in turn allows transferring permission set from parent to child. The iCAP does not provide a mechanism for permission revocation when the condition changes.

The need for authentication component to identify or authenticate users in such environment where unknown users or malicious user has the ability to join such environment and try to access resources and services with no right.

III MODEL ARCHITECTURE

Our enhancement will be made on S. Emami [14] CAP, CAP has two main parts domain authority and session agent as shown in figure 1 , session agent created when the user starts or enter new session:

A Domain Authority

Collects long term contexts and responsible for assigning roles to the user in the beginning of the session based on long term context conditions related to the role, this part responsible for filling S-R in the session according to RPC, also appoints SPA and then sent it to the Session Agent (SA) to manage access, domain authority consist of the following components:

- Long-Term Context Manager: collect long term context information from sensors, and then convert it to predicate formula to be stored.
- Session Manager: responsible for handling session requests from users, also assigns session agent and session to a user.
- Dynamic User Role Assigner: assigns roles to the user session based on role assignment conditions then fills S-R storage.

B Session Agent

Collects short-term context information and evaluates user's access requests according to SPA, if Request authorization function accepts request then permission granted to the request issuer otherwise rejected, the main components in session agent:

- Short-term Context Manager: collect short term context information.
- Permission Authorizer: makes a decision about users access requests based on role permissions in the session.

IV FRAME WORK DESIGN

A. Dey [15] defines context aware framework as "The framework will allow application designers to expend less effort on the details that are common across all context-aware applications and focus their energies on the main goal of these applications". Our improved framework will send to client required context information set with change factor for each one, then the client will check if the previously required context information breaks this threshold or factor, as a consequence will notify the web service in order to reevaluate permissions granted or denied based on this context information.

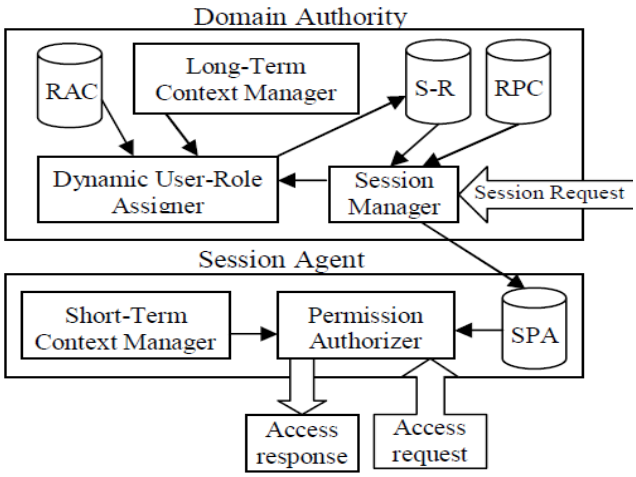

Fig. 1 RBAC relational diagram [16]

A Domain Authority

As shown in figure 2, the improved context aware framework as described in previous contributions and studies split into two main components domain authority and session agent:

- Domain Authority

contains session manager which responsible for handling session requests issued by clients, long term manager which handles and manages context information acquisition and distribution, dynamic user role assigner handles and enforce assignment of roles to users according to long term context information (LTCI) and role assignment conditions (RAC) then fills session role S-R database.

- Session Agent

Contains short term context manager which responsible for managing context information acquisition and distribution which classified as short term, permission authorizer make decisions about user access requests based on granted permission for the session which belongs to, stored in the dynamic database Session Permission Assignment (SPA), also session agent contains new component called environment context information validity controller which also responsible for controlling validity of context information such as environment related for example time, permission authorizer will check its validity before making decision.

- Client Proxy:

Will be responsible for enforcing and ensure the integrity of context information measured through secure communication channel to prevent fraud attempts from malicious users, client agent will contain context information manager responsible for acquisition and distribution of measured context information, and user context information validity controller which will check that any context information change and is less than change threshold or not to issue event to the web service to revaluate access request decision made.

TABLE 1
Web Service execution time average using JUnit testing.

Test Case	Validate User	Session Request	Loading granted Permissions	Loading Local conditions	Total Response Time
1	16	615	85	482	1301
2	10	560	65	417	1152
3	19	636	50	482	1308
4	19	585	43	474	1297
5	13	539	41	508	1212
6	14	526	43	472	1165
7	25	621	53	493	1308
8	11	495	72	413	1090
9	46	501	39	470	1214
10	20	594	55	581	1354
Average (run time)	19.3	567.2	54.6	479.2	1240.1

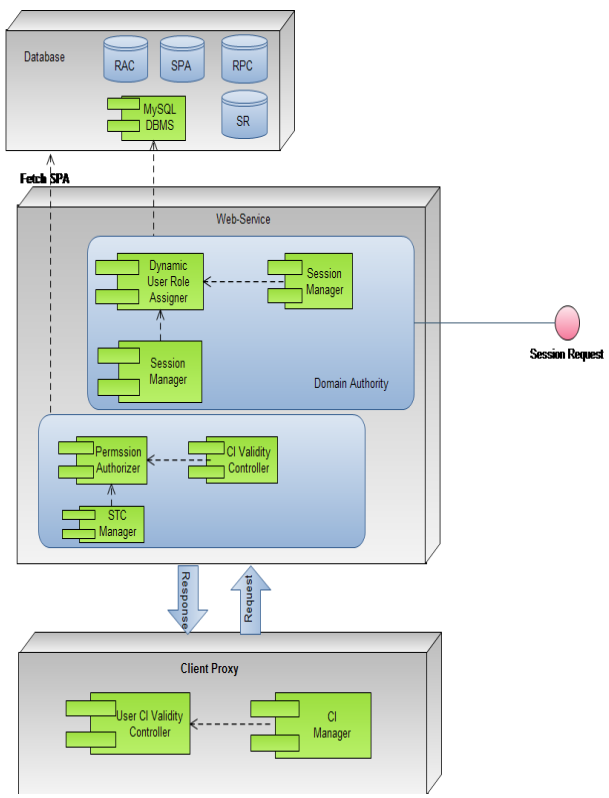

Fig. 2 Enhanced RBAC relational diagram [16]

1. User Context Information Validity Controller:

Responsible for handling and monitoring context information validity based on value change and how that change affect permission condition validity being guard, while the number of conditions being guard for each user individually, will dramatically degrade required performance to server side framework.

2. User Context Information Manager

Collects context information and format it to be processed by validity controller, such module needs to be pluggable module that enables adding new tools and APIs to broaden its ability to measure and sensing user and environmental context information, with reasoning mechanisms.

V FRAMEWORK TESTING AND EVALUATION

We will show the testing of the framework results for improvement made:

A. Experimental setup

We describe in this section what are needed to set experimental environment in order to evaluate the system from perspectives being under focus. Our framework includes web service implemented using java programming language, and deployed using GlassFish Server 4.0 , the web service use as a back end database MySQL 5.1.37 database management system. The platform incubate such server applications, is Intel(R) Core™2 Duo CPU P8400 @2.26 GHz, 3072 MB RAM. Also the experiment includes mobile device with android operating system, has the following specification, 1 GHz dual core Cortex-A9, 800 MB RAM, Wi-Fi 802.11 a/b/g/n dual band, Android OS, v4.4.2 (Kitkat).

All devices are connected using wireless network access point 150 Mbps, Model No: TL-WA701ND. The testing and evaluation measurements will be shown independently, due to inexistence of similar development for such frameworks.

B. Performance

Evaluation of the performance considers response time as criteria to measure system performance, also we will measure response time for critical operations and tasks done on both web service response time, then we will test response time for the proxy, which in turn call web service remotely to access services or resources, also for the most expensive and important tasks:

1) Calculation of web service tasks execution time evaluation:

Using JUnit test:

1. Validate User
2. Loading granted roles – FillSR
3. Loading granted Permissions – ValidateAllPerms
4. Loading conditions needed for gaining role permissions – FillCondsByRole

The table [1] represents the results of executing the web service which handle the main evaluation process for access decision made. We didn't include load granted roles, because execution done once during the session, also our enhancement focuses on changes made on short term context, where granting roles related to long term context done one time each session.

2) Calculation of proxy execution time evaluation:

We evaluate and statistically compare results on the device:

1. Load Local RPC
2. Load Local SPA

As shown in figure 3 based on result collected in table 1, we can see that the session request operation and Loading local conditions operation have high execution time includes the following main operation with high execution time, then we need to analyze session request operation as follows:

1. Establishing a session for the end user.

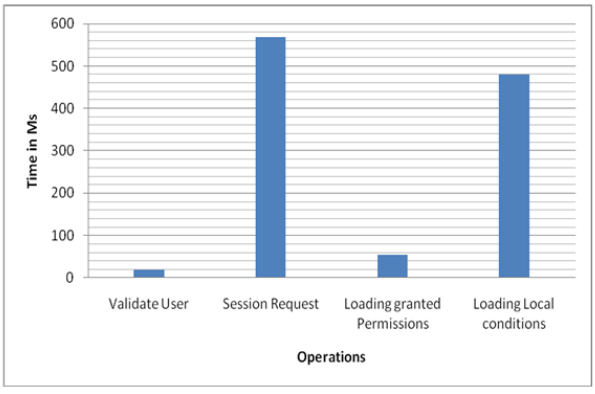

Fig. 3. Operations execution time scale chart.

2. Evaluating the roles assignment based on conditions from preset policy and represented previously in a database table.

3. Evaluating the permission assignment based on conditions from preset policy and represented inside the database table.
   a. Loading granted Permissions – ValidateAllPerm.
   b. Filling them to the SPA.

To approve our enhancement made by adopting the new approach, let us suppose that we have a session with 100 access request, then the execution frequency for each user as the following:

As shown in Table 1 the Validate task and FillSR done once at the beginning of the session, which FillSR depends

on long term context information as noted before such LTCs doesn't change during the session average time. The most frequent operation which occurs with every change to the context information provided by proxy, such change enforce the framework to revaluate conditions validity, in order to identify which permissions will continue grant or will be revoked.

C. Security

We should take into account for matching security CIA triad:

1) Confidentiality and Integrity:

To address system confidentiality we have two choices:

- Transfer soap request using SSL.
- Using SOAP extension WS-Security.

While the pervasive environment communication resources is expensive and limited, also the computation power is relatively low, where most devices which communicate is low power devices also to decrease battery life impact for portable devices, that leads us to choose less expressive technique to avoid the constraints that we have.

We use SSL as technique to transfer soap messages between the framework parts, through using third party library ksaop2-android, to ensure security matching side by side save system parts performance to be more convenient for pervasive environment.

Also using proxy side to control context information foster integrity, where malicious users will be prevented from injecting or providing the web service with false values for the framework web service which leads for false access control decision.

2) Availability:

The most risky reasons for falling down or stopping the system from functioning return to the following problems:

- System failures

Table. 2. comparison of estimated of tasks execution time frequency during the session.

Operation		Estimated Execution Frequency per session	Execution time	Total Execution time
Validate User		1	19.3	19.3
Session Request	Establishing a session	1	110	110
	Evaluating the roles assignment based on conditions (FillSR)	1	188	188
	Evaluating the permission assignment (ValidateAllPerms && FillSPA)	100	215	215000
Loading local permissions		100	54.6	5460

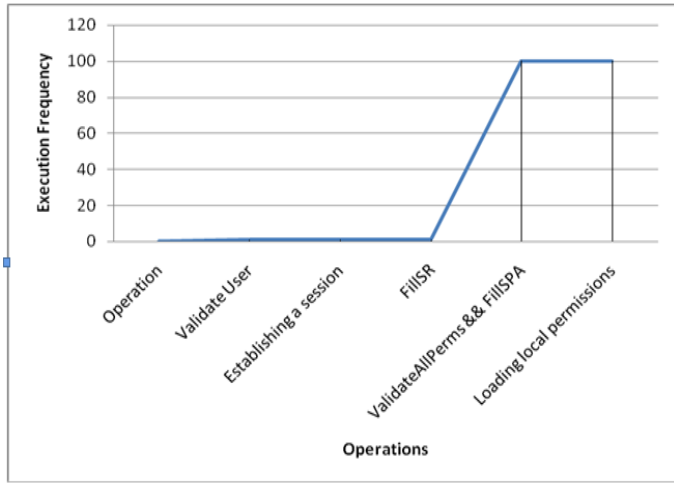

Fig. 3. Operations execution time scale chart.

Failures such inability to read data sources, or to make comparisons or business logic processing, so we conduct unit testing, to ensure that each functionality works and output results correctly, also we foster during programming testability of the software, like simplicity and independence of modules and components also to improve fault tolerance we use exception handling.

- Denial of service attacks

Another important drawback for availability is service interruption, as a consequence our system architecture are distributed which in turn simplify and facilitate monitoring of each part, so that's leads to reduce service bottleneck problem, we used connection pooling for managing connections in order to alleviate any overload made on the system due to user large numbers.

Table. 3 Samsung Device with Android OS run the framework

Test Case	Validate User	Session Request	Loading granted Permissions	Loading Local conditions	Total Response Time
1	60	805	407	838	3136
2	116	852	315	859	3986
3	119	730	330	828	3207
4	156	948	416	584	3227
5	108	998	525	1277	4802
6	215	935	448	1013	4825
7	101	611	367	764	3230
8	93	572	318	694	3189
9	112	659	145	629	2951
10	99	805	197	776	3163
Average (run time)	117.9	791.5	346.8	826.2	3571.6

VI. CONCLUSION

Context awareness application has increasing significance in various environments and domains; one of these domains is authorization, while authorizations techniques are various, we selected role based technique as a base for research, such techniques fits the organizational structures, where roles is a matching for persons functional behavior for a specific system or domain.

In this paper we attempts to investigate and enhance the model S. Emami [6] to be applicable to work in pervasive environment, through adding the proxy module, such module works with end users as interface that facilitate to them access specific system resources based on preset policy by system administrators, also responsible for monitoring context information collected by the framework to make access decision, where any change occur will notify context manager for the change to revaluate related permissions or roles granted based on such context information.

Also the proxy increasingly enhance the model performance when applied, where the framework will not require revaluation for policy rules each request made, instead each device for end user will independently monitor and notify for the change when occur and hold resource utilization.

The enhanced framework also will enhance system applicability from the security perspective, where malicious change for context information, which intended to change access decision, became impossible.

VI I. FUTURE WORK

The future expansion of enhanced framework has multiple directions, where such of research includes many areas and disciplines needs to be enhanced and covered, especially researching the dynamicity of mounting context information from available sensors, focus study for best selection of encryption techniques for messaging which has neutral effect on performance. Also the certainty of context information values should be researched to enhance value measurement with lower battery impact. Also we need applying further study for user privacy preferences to be taken into account when applying the framework policy, another important research issues should be extended this study especially in making the framework has zero configuration to enhance usability in various domains and business environments.

ACKNOWLEDGMENT

Table. 3 Samsung Device with Android OS run the framework

Test Case	Validate User	Session Request	Loading granted Permissions	Loading Local conditions	Total Response Time
1	60	805	407	838	3136
2	116	852	315	859	3986
3	119	730	330	828	3207
4	156	948	416	584	3227
5	108	998	525	1277	4802
6	215	935	448	1013	4825
7	101	611	367	764	3230
8	93	572	318	694	3189
9	112	659	145	629	2951
10	99	805	197	776	3163
Average (run time)	117.9	791.5	346.8	826.2	3571.6

We are really feel [illegible] port introduced from information technology faculty at Islamic university – Gaza for their support represented by deanery; also I really appreciate help introduced from Palcore Microsystems Co. for their sponsorship for research activities and financial requirements.

REFERENCES

[1] Mark Weiser. “The Computer for 21st centuray” [Journal]. - 1991.

[2] B. Schilit, M.Theimer. “Disseminating Active Map Information to Mobile Hosts” [Journal]. - 1994.

[3] B. Schilit, Norman Adams, Roy Want. “Context-Aware Computing Applications” [Journal]. - 1994.

[4] R. Sandhu D.F. Ferraiolo, D, R. Kuhn. “The NIST Model for Role Based Access Control: Toward a Unified Standard” [Journal]. - 2000.

[5] G. Zhang and M. Parshar. “Context-aware Dynamic Access Control for Pervasive Applications” [Journal]. - 2004.

[6] S. Emami , S. Zokaei. “Context-Senstive Dynamic Role-Based Access Control Model” [Book]. - [s.l.] : ISecuri, 2010.

[7] A. Toninelli1, Rebecca Montanari1, Lalana Kagal2, and Ora Lassila. "Context A Semantic Context-Aware Access Control Framework for Secure Collaborations in Pervasive Computing Environments." 5th International Semantic Web Conference, ISWC 2006. 2006

[8] C. Ardgana, Ernesto Damiani, Sabrina Vimercati,Pierangela amarati. A web service architecture for enforcing access control policies".Proceedings of the First International Workshop on Views on Designing Complex Architectures, 2004.

[9] M. Covington M. Moyer, and M. Ahamad. “Generalized role-based access control for securing future applications” [Journal]. - 2000.

[10] Y. G. Kim C. J.Mon,D. Jeong,C.Y. Song and D. K. Baik. “Context-aware access control mechanism for ubiquitous applications” [Journal]. - 2003.

[11] W. Han Junjing Zhang, Xiaobo Yao. “Context-sensitive Access Control Model and Implementation” [Journal] // IEEE. - 2005.

[12] J. Hwan Choi Hyunsu Jang and Young Ik Eom. “CA-RBAC: Context Aware RBAC Scheme in Ubiquitos Computing Environments” [Journal]. - 2010.

[13] T. Devdatta K. and Anand. “Context-Aware Role-based Access Control in Pervasive Computing Systems” [Book]. - 2008.

[14] S. Emami M. Amini, S. Zokaei. “A Context-Aware Access Control Model for Pervasive Computing Environments” [Journal]. - 2007.

[15] A. Dey, K., D. Salber and G. D. Abowd (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction 16. To appear in 2001.