2005 IEEE International
Symposium on Signal Processing
and Information Technology

A Scalable Security Architecture Enabling Coalition Formation between Autonomous Domains

Petros Belsis, Stefanos Gritzalis, Sokratis K. Katsikas
Laboratory of Information and Communication Systems Engineering *(Info-Sec-Lab)*
Department of Information and Communication Systems Engineering
University of the Aegean, Samos, Greece,
{pbelsis, sgritz, ska} @aegean.gr

Abstract- *Coalitions between autonomous domains are often formed in real life scenarios in order to enable access permissions to shared objects on grounds of bilateral resource-sharing agreements. The dynamic nature of coalitions poses new challenges relative to security management and joint administration of resources; therefore we distinguish a need for reconciliation and extension support to single-domain oriented security models, so as to incorporate location, time and context based related parameters in their role definition schemes. In this paper we introduce a robust and scalable solution that enables the realization of coalition formation in a multi-domain policy ruled environment.*

Keywords- *Coalitions of Autonomous Domains, Security Policies*

I. INTRODUCTION

In many real-life situations there is a need for collaboration between different organizations; many environments can benefit from a joint administration of applications and resources. For example, e-healthcare is an area that can suffice from the creation of collaborations between interconnected medical information systems; e-Government is also another environment where the joint participation of agencies and ministries in coalitions can improve response times and enable the provision of high quality services as well as to improve proactive crime incident detection.

The dynamic nature of coalitions demands a flexible way for joint administration of resources. The participation in a coalition under the framework of a collaborative environment, can be established by resource access agreements ranging from those of peer to peer sharing of applications or files to those of joint administration of access control policies of autonomous domains [2]

Distributed systems contain a large number of heterogeneous resources spanning often organizational boundaries; their management cannot be centralised but requires flexible and dynamic administration. Many solutions have emerged to enable the automated management of resources by means of security policies. In most of the cases the determination of policies and access control rights is in accordance with the Role Based Access Control (RBAC) model, due to its capability and expressiveness to reflect organizational hierarchy and to map users to roles and to associate them with security related attributes and privileges.

In contrast to the internal determination of roles and security parameters which can be managed to some extent, there are many potential dangers and an increased complexity related with every attempt to establish a joint collaboration between different organizations, thus exporting rights to third party roles; there is also an increased danger for potential unauthorized information leakage.

In this paper we define a system that incorporates constituent RBAC elements and additively incorporates time and location related extensions to the standard RBAC. Authorization can also be provided based on context attributes, for example based on domain specific parameters (example all users who belong to defence.gov.uk or all users who belong to finance.gov.uk). We also provide support for within time-intervals role activation (RBAC sessions), in contrast with other approaches for multi-domain environments that do not provide support for sessions [2]. Our aim is to enable users to access a coalition's resources transparently, no matter if the assets reside in the same organization with the users, or they reside in a remote collaborative domain. For example the doctors from a hospital in the surgery department could be appointed to work for some hours a week as general practitioners (GP's) in another department. While they are on duty in one department, it is possible that they will need access at the distant domain. One solution could be that the hospital provides a portal so that the doctors could log on to the system with a web browser, providing their credentials and accordingly exercise some tasks. This adds extra complexity and extra costs to the development of the information system, as it requires the creation of specific interfaces for distant access. A more challenging option could be the integrated security management of both domains, which enables users to perform the same task transparently for tasks performed either on the remote or the local domain.

The rest of the paper is organized as follows: Section 2 presents related work; Section 3 presents the main RBAC principles and discusses the necessity for adding parameterization and extension support to the standard model in order to adjust to multi-domain environments. Section 4 presents our policy representation, based on the Resource

Description Framework (RDF), which is more expressive than the standard XML-based policy recording, and presents our scalable solution for role appointment between different organizations. Section 5 presents an application scenario, while section 6 concludes the paper.

II. RELATED WORK

The concept of parameterized roles in RBAC is not new; most real world systems provide support for such a parameterization. This trend has been further reinforced by specific parameterized models present in the NIST RBAC standards [11]. Equally, many researchers have examined how to include dynamic environmental interaction in their RBAC models [1, 3, 4]. Our concern is how to incorporate several context based parameters in the role specification and access control enforcement process. In addition, codification in an both and interoperable and expressive way as the one presented in our approach is highly desirable.

Joshi et al. [1] define a language for multi-domain environments and introduce an approach that utilizes Extensible Markup Language (XML) for role representation, user-to-role assignment and permissions-to-role assignment. In their model they consider time and location based role-parameterisation.

Khurana et al. [2], introduce the problem of coalition formation between autonomous domains and address ways so that dynamic coalitions could be formed by reaching a commonly accepted by the coalition members' access-state where all the resources are jointly shared. Access to common resources is enabled by using shared public keys. Their work does not support the concept of RBAC sessions, while role-hierarchies are not supported. Additively for coalition resource management all the roles that participate in the coalition are presupposed to have access to the shared resources.

Belokosztolski [9], [3], introduces interface policies and the notion of contexts, which control information flow in order to enable policy mappings. Interface policies are generic policies that act as mediating policies between the collaborating domains. XML is utilized for policy storage. We utilize a more expressive way based on RDF role representation, while we avoid the additive overhead in defining security policies for each domain and mediating policies.

Our work enables the formation of coalitions between autonomous domains, while it provides with a robust and scalable solution for the realization of resource sharing between the participants in the coalition. We utilise XML based access request messages representations, while we adopt a more expressive framework for policy recording, that enables the determination of role-hierarchies based on the Resource Description Framework (RDF) [7] ontology language. We introduce the concept of role-mappings, while we enable the single-way role cross-appointment between different domains.

III. PARAMETERISED RBAC

The RBAC model [11] has become dominant due to its capability to reflect real world situations. Fundamental concepts of RBAC are users, roles and sessions. The key-concept of roles introduces an abstraction, implying that a specific user is assigned specific permissions that indirectly associate a specific individual with the privileges and permissions associated with the job he/she performs within the organizational context.

To extend the support for least privilege, sessions are introduced. Sessions serve the case where a user signs on a system to perform some specific task. Sessions introduce some abstraction between users and roles. Therefore, sessions enable users to activate temporarily different roles, or to log with two or more roles at the same time.

Several modifications and extensions to basic RBAC are necessary to adjust it to multi-domain environments. In our approach we consider the determination of role-related parameters, that:

- Are pre-settled by the system administrator, and that the user must specify in order to be granted authorization to activate a role (parameters domain specific mainly, that correlate a user with a specific domain, ex IP addresses, or DNS domain names).
- Enable time periodicity, for example allow access within pre-specified time-intervals
- Enable context based role-assignment and role correlation for roles specified in different domains.

A. Multi-domain security models

Multi-domain coalitions are prominent in a big number of emerging networked infrastructures. Two kinds of access control systems can be considered under this (collaborating) framework: peer-to-peer networks, and autonomous domains. Peer to peer networks are formed by networked community members who share a common purpose. The binding that characterises peer-to-peer networks is more loosely coupled than in the case of autonomous domains. The second category of multi-domain coalitions can be met in many real-life systems, such as interconnected e-healthcare information systems, or e-government environments consisting of interconnected ministries or public agencies. In the latter case, sensitivity of the data poses more security restrictions and establishing a common state for knowledge exchange requires both that organizational roles are well defined in terms of access rights and obligations based on the grounds of a well-stated security policy, while a common access state between different organizations is unambiguously allocated.

We can classify coalitions according to the access models they adopt, to the following two categories:

- Trust based. The notion of trust is introduced mainly in complex, non-hierarchical or inter-related systems such as the Internet. Trust based systems enable the mapping of unknown users, which carry some type of credentials to privileges. The level of trust associated with each user can vary and so can the assigned privileges.
- Autonomous, policy-managed, with well formed security policy and well defined organizational structure. These systems are ruled by an internal security policy.

Our research focuses on the second category of coalitions, characterized by well-defined organizational policy and cooperate and on the grounds of a commonly established resource sharing agreement enabling access from one domain to the other.

B. Policy-based management

Large organizations contain a huge number of heterogeneous components, which span often over a large number of networks. Centralized management solutions are inadequate for these environments. Security can be simplified with the utilization of security policy languages, which enable the automated administration for both a large number of target objects as well as requesters. The policy is stored in repositories either databases or can be recorder in XML files. XML is preferable for several reasons; among else its interoperability features and platform independence, as well as its tendency to be adopted as an evolving standard in the area of distributed computing.

We have utilized the Extensible Access Control Markup Language (XACML) [5] RBAC-based framework as a basis for domain specific authorization. XACML is a language for expressing access control policies that enables its codification in XML format. It allows control over actions and supports resolution of conflicts. XACML does not support role-hierarchies and it is not characterized by the expressivity and extensibility provided by semantic web languages [6]. In order to overcome these limitations we have chosen RDF as a basis for policy representation, which is more expressive and enables us to provide support for role hierarchies, still retaining the main principles of XACML for request formulation and evaluation according to the specified policy. Moreover, the basic principles of our approach are not language specific and can be easily applied to different interpretations of RBAC for each domain. Therefore our framework and implementation are language neutral.

In Table 1, we are defining examples of time-enabled activation of logging for a set of roles, as well as we provide context-based authorization for the users belonging to a specific domain.

Table 1. Example of context-enabled authorization for multiple roles during pre-specified time intervals.

```
<Rule RuleId="EveryoneDuringBusinessHours" Effect="Permit">
<Condition FunctionId=" Function#time-in-range"> <Apply
FunctionId="function:time-one-and-only"><EnvironmentAttributeDesignator
DataType=http://www.w3.org/2001/XMLSchema#time  AttributeId="
environment:current-time"/></Apply>
<AttributeValue DataType="
http://www.w3.org/2001/XMLSchema#time"> 09:00:00
</AttributeValue>
<AttributeValue DataType="
http://www.w3.org/2001/XMLSchema#time">17:00:00
</AttributeValue>
</Condition></Rule>
```

IV. RDF BASED RBAC ROLE REPRESENTATION

XML role representation is advantageous due to the widespread support and interoperability features that XML enjoys. Lately, it also tends to become de facto standard for distributed environments. XML though lacks in terms of expressiveness comparatively to RDF; we have chosen RDF because it enables us to express more complicated relations between roles, for example role-hierarchy representations, something that in XML would require considerable extra cost in terms of time and complexity for both the construction of XML files as well as for applications.

Table 2 depicts part of the role representation document (Fig. 1 presents schematically the role hierarchy). Role names are available at the location http://defenseMinistry.org/roles, while the vocabulary related to their permissions is available at http://defenseMinistry.gov/permissions. The hierarchy between the roles is defined through the predicate "supervises" that contains roles supervised by the containing role. The "prm:supervises" tag is a collection of role names as it is shown in the rdf-based example. The example defines for all the roles their activation and deactivation times through the "prm:activation-time" and "prm:deactivation-time" elements. Environmental related predicates are also defined (for example the domain's name "intelligence.defenseMinistry.org" in the prm:DomainDescription tag can be used as a predicate that enables access to all members of the domain).

A. Defining policy mappings

In order a user to access resources to a remote domain, the correspondent role on the target domain has to be retrieved. The mapping process has to be activated. A one-by-one role mapping between all the domains of the coalition would increase complexity and would seriously put the system's scalability under question. Therefore, we adopt the following solution: we introduce a general role hierarchy, with generic roles, to which the local roles from the participating domains have to be mapped (Fig 1).

The mapping will be responsibility of administrators, which are aware of the consequences of an incorrect mapping and potential information exposure to non-authorized personnel. The global schema role hierarchy assumption to which local policies map is not restrictive, as it is usually the case in many agencies or ministries to have similar roles but

not identical. For example ministries have ministers, general secretaries, Department Directors and so on.

Similarly, hospitals have Ward Managers, Specialized Doctors, and Nurses. Now doctors who work in one clinic as general Practitioners in one clinic could be appointed to work also in another clinic. So a mapping between similar roles could be established in terms of a common role hierarchy, or more precisely in our case based on a generic (global) ontology role representation. We do not present solutions for composition of proposals in the absence of a mutual trust environment between the coalition domains. Such issues demand invocation of game theory principles [12]. In addition we do not consider the case where domain-policies are sensitive and policy exposure between the different domains may consist of a risk factor for some or all of the domains.

Table. 2. RDF-based Role attribute and hierarchy definition (fragment)

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF>xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:base="http://defenseMinistry.gov/roles"
xmlns:prm="http://defenseMinistry.gov/permissions">
<rdf:Description rdf:ID="GenSecretaryA">
<prm:activation-time>9:00</prm:activation-time>
<prm:dectivation-time>23:00</prm:dectivation-time>
<prm:DomainDescription>intelligence.defense.org</prm:DomainD
escription>
<prm:supervises parseType:="Collection">
<rdf:Description rdf:ID="SectorA2Director"/>
</prm:supervises>
</rdf:Description>
<rdf:Description rdf:ID="NavalManager">
<prm:activation-time>9:00</prm:activation-time>
<prm:deactivation-time>17:00</prm:deactivation-time>
<prm:DomainDescription>intelligence.defense.org</prm:DomainD
escription>
<prm:supervises parseType:="Collection">
  <rdf:Description rdf:ID="AgentRole1"/>
  <rdf:Description rdf:ID="AgentRole2"/>
</prm:supervises>
</rdf:Description>
</rdf:Description>
</rdf:RDF>
```

We enable role mapping to be performed on single-direction basis. For example a role in one organization could acquire the permissions of another role on the target domain, without the opposite. Therefore we distinguish two types of mappings, in-mappings towards the central ontology, and out mappings directing towards local roles. For the mapping process we define paths by using the XPATH [8] query language. XPATH aims in addressing parts of XML documents. It represents location of data in an XML document correctly and efficiently, which makes it a suitable language for both XML query and access control [10]. An example mapping based on XPATH is presented in Table 3, where roles from one domain are mapped to another domain's roles indirectly through the central role hierarchy.

Table 3a. XPATH based role mapping (in-mapping) between local and central ontology hierarchy roles

DefenseMinistry	CENTRAL
Minister/GenSecretaryB/SectorB2Manager	**Minister/GenSecretary/SectorBDirector**

Table 3b. A role from the Central hierarchy maps to two local roles (out-mapping)

PublicAffairsMinistry	CENTRAL
Minister/GenSecretary/SectorADirector	**Minister/GenSecretary/SectorBDirector**
Minister/GenSecretary/SectorBDirector/EmergencyReactionHead	**Minister/GenSecretary/SectorBDirector**

Therefore we define paths that allow the mapping of roles between different role schemata. Notice that due to the expressiveness of XPATH, one can represent more complex role mappings in a very compact way, by grouping together equivalent roles in one XPATH expression, without having to write separate rules for each role.

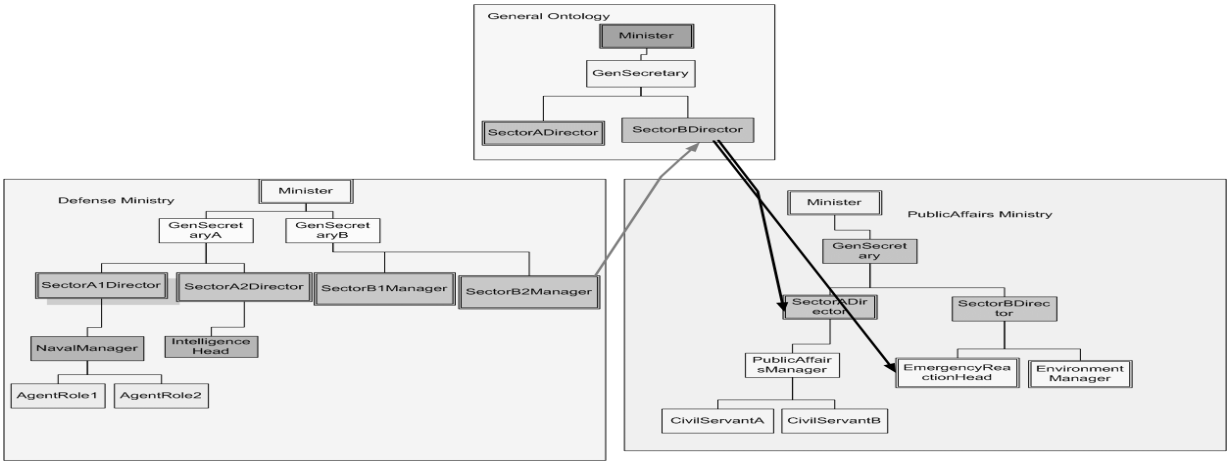

Fig. 1. Role mapping across local and global role hierarchies. A local role on domain A (ministry) maps to global hierarchy and accordingly to different roles on another ministry

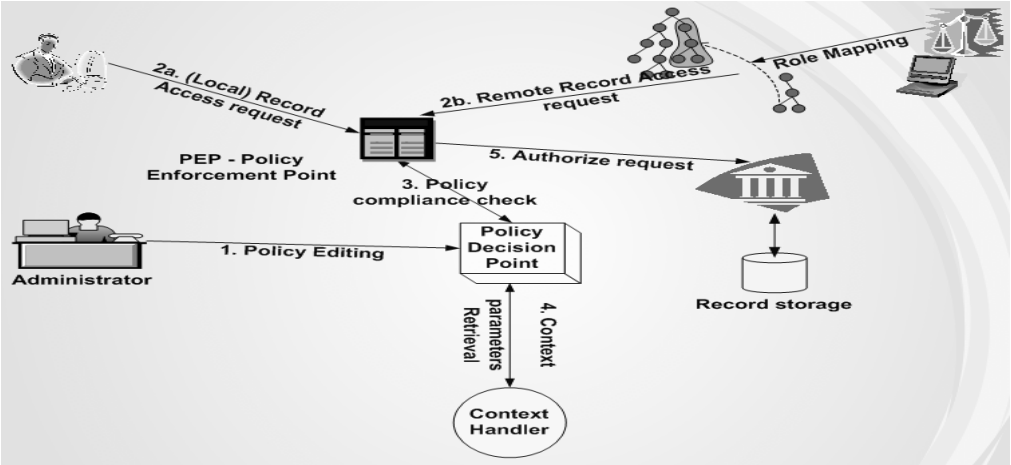

Fig. 2. (Local and remote domain) Authorization process in steps

B. Local and Remote Authorization Processing

Our authorization module follows the basic XACML principles, with the modifications that enable RDF based higher level policy editing and interpretation and the deployment of a module that enables mapping of roles between different domains through the commonly agreed intermediate authorization hierarchy. The basic modules that enable authorization and access control enforcement in our framework are: the Policy Enforcement Point (PEP) where the access control decisions are enforced, the Policy Decision Point (PDP) where requests are evaluated according to the local policies and contains also a registry with role-mappings (both in and out mappings) and the context handler that collects context-parameters to facilitate domain-specific authorizations.

The basic operating principles of the authorization module are the following (Fig. 2): The policy administrator is responsible for editing the policy and making it available for the When a request for a resource appears, it has to be validated for its consistency with the local security policy prior to its execution. domain, through the Policy Decision Point (PDP). Accordingly, each request is directed to the Policy Enforcement Point (PEP). The request is constructed in an appropriate XML message and directed to the Policy Decision Point (PDP). Prior to the validation of the request, the context handler is sending additional subject, resource, action and environment attributes to the PDP. At last, the request is validated from the PDP and a response message is sent to the policy enforcement point (PEP), which handles the details for providing authorization to the requester.

In case of a request from a remote domain, the remote role is transformed to its equivalent in the destination domain through the role-mapping process. This process consists of the following steps: first of the creation of an appropriate message from the remote domain to the destination domain, containing the request for a specific resource, the corresponding role on the central authorization hierarchy (by retrieving from the remote domain's PDP and its registry the appropriate in-mapping), and accordingly by creating a digitally signed message with the remote domain's specific signature in order the destination domain to validate the origin of the request. The domain which receives the message, evaluates the request by retrieving through its mapping registry (stored with the PDP) the appropriate mapping and assigns a corresponding role in its domain, or in case something like that does not exists it denies access to the remote user.

In order to adjust to the demands of highly distributed scenarios, we have extended the XACML authorization scheme by deploying redundant PDP (Policy Decision Point) and PEP (Policy Enforcement Points) entities in the network- instead of only one- to cope with issues such as the presence of a single point of failure or in order to be able to adjust the authorization module resource-demands to low resources devices such as the ones present in pervasive infrastructures. So instead of deploying a single, centralised PDP we attempt to share this responsibility between a number of domain network nodes which collectively act as a centralised authorization module several redundant PEPs can also be deployed, ensuring that prior to accessing a shared asset, the requester's privileges are compatible with the predefined domain's policy.

V. APPLICATION SCENARIO

Consider the following scenario: A civil servant is responsible for issuing professional licenses. In most of the cases a proof of criminal record status is necessary, issued by a third party (ministry of justice). The process of issuing such a certificate could delay until the citizen brings the certificate or until the two ministries contact each other, request from each other the documents, they verify the requester's ID and accordingly they exchange the necessary information. In a collaborative environment things can become simpler and procedures can become simplified amazingly. The local role on one ministry could be mapped to a correspondent role on

the other ministry with appropriate level of security and when making a request, a local role would be assigned to the remote requester and perform the remote task like being a member of the target organization (with appropriate security clearance level). The same mapping could be helpful for another civil servant in another ministry for example the ministry of finance who could want also to examine a citizen's background for previous convictions for economic crimes and so on for many other domains. The only additional cost is in the creation of the global role hierarchy and to perform the correct mapping. Usually though, for cooperating organizations there are similar hierarchies (for example ministries share similar role-schemas, or hospitals also). The global hierarchy scheme is not necessarily stored to a single location but copies of it can be replicated to all the participating domains, so that domains can make proposals for appropriate mappings. Again, we consider that domain administrators have awareness of the legal implications of inappropriate mappings, or that mappings s have to be agreed on a basis of mutual agreement, as for example would happen in cooperating organizations who operate under the same framework (ex. Ministries, hospitals etc.), if the presence of a transparent cross-organizational authorization mechanism was not present as in our case.

Our architecture retains basic principles [1] for the collaborative interoperation of domains, such as:

- Autonomy: actions permitted within individual systems are allowed within secure interoperation
- Security: no unauthorized action within the individual system's boundaries should be allowed under secure interoperation

By introducing the role mapping process as aforementioned we enable authorization and access control enforcement for roles from distant domains, in an easy to achieve and interoperable solution. In addition, the number of the participating domains may increase, and therefore our solution is characterized by its scalability features. At the same moment, it is robust, since the mappings are established in a deterministic way, which does not allow involvement of risk factors such as the ones presented in trust-based approaches, which makes our solution suitable for high-sensitivity environments. Also, with minor modifications we can deploy redundant modules of our mechanism adjustable to highly distributed environments where instability urges for topology reconciliation [13].

VI. CONCLUSIONS

Coalitions between autonomous domains introduce a variety of challenges relative to security management. Different policy models and role representation schemes need to be integrated in order to enable transparent access to authorised users that belong either to the local or to a remote, collaborative domain.

We introduced a scalable solution that enables the formation of coalitions and enables role mapping between different domains. Due to our choice to record role definitions in RDF rather than XML we enable a more expressive role representation scheme and role hierarchy capturing. We also enable context-based authorization according to domain-specific attributes. The proposed solution is characterised by its scalability features as well as for its robustness.

We are currently working on extending the applicability of the aforementioned solutions on specific-requirements environments such as low device-resource pervasive environments.

ACKNOWLEDGMENTS

We would like to thank C. Doulkeridis, V. Zafeiris and A. Malatras for their insightful comments on parts of this paper. We would also like to thank those reviewers who made really insightful comments and helped us improving parts of the paper.

REFERENCES

[1] Joshi J.B.D., Bhatti R., Bertino E., Ghafoor A., "Access Control Language for Multi-Domain Environments", IEEE Internet Computing, Nov. 2004, pp. 40-50.

[2] Khurana H., Gligor V.D., Proceedings of the 13th IEEE Intern. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'04)

[3] Belokolsztolszki A., Eyers D., Moody K., "Policy Contexts: Controlling Information Flow in Parameterised RBAC", Proceedings of the 4th International Workshop on Policies for Distributed Systems and Networks (POLICY'03), IEEE press, pp.99-110.

[4] Bertino E., Bonatti P. A., Ferrari E., TRBAC: A temporal role-based access control model. *ACM Transactions on Information and System Security (TISSEC)*, 4(3):191–233, August 2001.

[5] Organization for the Advancement of Structured Information Standards (OASIS), XACML Extensible access control markup language specification 2.0, OASIS Standard, (available at http://www.oasis-open.org Accessed May 2005.).

[6] Patwardhan A., Korolev V., Kagal L., Joshi A., Enforcing Policies in Pervasive Environments, In Proc. of the MobiQuitous 2004 1st Annual Conference on Mobile and Ubiquitous Systems, IEEE Press.

[7] Beckett D., ed., RDF/XML Syntax Specification, W3C Recommendation,www.w3.org/TR/rdf-syntax-grammar.

[8] www.w3.org/TR/xpath (Accessed May 2005)

[9] Belokosztolski A., "Role based access control for policy administration", Phd Thesis univ. of Cambridge, UK, 2004 available at *http://www.cl.cam.ac.uk/* as technical report No 586

[10] Jeona J.-M., Chungb Y.-D., Kima M.-H., Lee Y.-J., "Filtering XPath expressions for XML access control" Computers & Security (2004) 23, 591-605

[11] Sandhu R., Ferraiolo D., Kuhn R., The NIST model for role-based access control: towards a unified standard. In Proceedings of the Fifth ACM Workshop on Role-Based Access Control (RBAC'00), pages 47–63, 2000

[12] V. D. Gligor, H. Khurana, V.D.Gligor, H. Khurana, R. Koleva, V. Bharadwaj, and J. Baras, "On the Negotiation of Access Control Policies", Proceedings of 9th Security Protocols Workshop, Cambridge, UK, Springer-Verlag, 2001

[13] Malatras, A., Pavlou. G., Belsis, P., Gritzalis, S., Skourlas, C., Chalaris, I., Secure and Distributed Knowledge Management for Pervasive Environments, Proceedings of 1st IEEE International Conference on Pervasive Services, Santorini, Greece, 2005, pp. 79-87, IEEE press