Task-Based Access Control for Virtual Organizations

Panos Periorellis, Savas Parastatidis
School of Computing Science, University of Newcastle,
Newcastle upon Tyne, NE1 7RU, UK
{Panayiotis.Periorellis, Savas.Parastatidis@newcastle.ac.uk}

Abstract. GOLD (Grid-based Information Models to Support the Rapid Innovation of New High Value-Added Chemicals) is concerned with the dynamic formation and management of virtual organisations in order to exploit market opportunities. The project aims to deliver the enabling technology to support the creation, operation and successful dissolution of such virtual organisations. A set of middleware technologies are designed and being implemented to address issues such as trust, security, contract management and monitoring, information management, etc. for virtual collaboration between companies. In this paper we discuss the set of requirements for access control in dynamic virtual organisations that have been defined as part of the trust-related work. We propose a solution, which extends the ideas of role based access control (RBAC), and we examine the use of existing and emerging Web Services technologies as an implementation platform.

1 Introduction

The GOLD project aims to build the infrastructure that enables organisations to collaborate securely in order to achieve some common goal, enabling the formation of virtual organisations (VOs). The project deals with a number of concepts such as roles, permissions, obligations within an environment comprised largely of autonomous components which are brought together to form a VO. Part of the requirements state that interactions between components are done dynamically without any prior collaboration history. This has an obvious impact of the amount of trust participants[1] in the VO can place on each other. Our aim is to alleviate this problem (i.e. establish trust without historical information) by enhancing certain security aspects of our system. Within GOLD, we are tackling trust issues in 3 distinct ways:

- We are implementing access control policies that will protect each party's assets from unauthorized use while allowing sharing;
- We are working on issues relating to dependable systems architecture as a way of enhancing trust through system trustworthiness [1]; and

[1] In this paper, the terms "party" and "participants" refer to those autonomous components that form part of the system and collaborate/interact using the GOLD infrastructure.

- We are investigating the formation of trust zones[2] by developing rules to which all participants of the system adhere.

In this paper we are looking into access control requirements for dynamic systems such as virtual organizations. The paper is organised as follows. First we take a look at trust in general and discuss access control and the current state of the art. We discuss the general access control requirements and propose a solution which draws from previous experience and past projects the authors have been involved in. We conclude with a discussion on the current state of the art of the Web Services technologies that can be used to provide such a solution.

2 Trust

Trust is a non-functional system property. It emerges through sound structure, security and dependability. In computing literature the meaning of trust has shifted from a purely non-functional property, related directly to dependability [1], to a measurement of the accuracy of access control models and contract management. In the course of this paper we will address both structuring issues as well as functional aspects.

Trust is based mainly on information (how much you know about someone), history (past transactional experiences with someone), or context (being within a trust zone or a boundary of rules and regulations with someone). In all the above cases trust can be a direct evaluation between 2 parties such that *a* trusts *b*. In highly dynamic environments such as virtual organisations parties may not have the opportunity to create a history of transactions. Additionally, there is also a requirement to maintain one's privacy which gives rise to identity issues. Since historical data and identity can be compromised, the only visible avenues towards achieving some degree of trust between two collaborative parties are *third parties* and *value*. Value and the wider notion of added value (although out of the scope of this work) is a motivating factor when trust related information is missing. In the virtual world and in particular in virtual collaborations that require anonymity, privacy and they lack any historical context, the only visible avenue for achieving some degree of trust is the transfer of trust from the collaborating parties, to the medium via which they collaborate. In our case this medium is the infrastructure which we hope to provide in GOLD. It is the infrastructure that provides guarantees to each party involved in a collaboration, that their *trust policies* will be enforced while at the same time maintaining identity and identity related information private. Trust policies, which are the focus of this discussion, are essentially security policies which express a party's requirements to engage into a transaction or any form of collaboration with any other party. A party for example may require that data is exchanged encrypted using a specific encryption algorithm.

[2] The term refers to a conceptual boundary around all the components of the system. Components within a trust zone have agreed on certain rules and regulations prior to offering any services.

Policies can be dynamic and altered to reflect new security requirements. This generates the need for an environment with dynamic activation and de-activation of access control rights, and a move from a *passive security* policy, where permissions are assigned to particular roles, to an *active security* policy that distinguishes between roles and their instances.

3 Access Control

Active Control Lists (ACLs) have already been proven to be inefficient [2, 3]. Access lists built for every user lead to repetition of lists for users with similar rights. An extension to the ACL model is Role Based Access Control (RBAC) [4], which provides an additional security layer between the user and the resource. Access rights are given to roles (usually pre-determined) and then users are associated with those roles. The role-based access control model is more efficient since users are assigned roles rather that access lists. This implies that users of similar access rights are assigned a single role (i.e. a single ACL is assigned to many users), making the roles based approach a lot faster.

There are cases however when role-based access is also insufficient and, hence, additional security layers are required. Roshan [5] pointed out an example with a hospital emergency room, which demonstrates the problem. He explained that within an emergency room, *patients* and *doctors* are two typical roles. *Diagnosis* and *prescription* are rights of the *doctor* and therefore *doctors* need to have permissions to access *patient*'s details. This however does not imply that *doctors* are permitted to prescribe to any *patient*. Only assigned *doctors* have access rights and consequently permission to prescribe to a particular *patient*. In other words, there is a binding between a *doctor* role instance with the *patient* role instance, which implies the need for an additional layer that binds instances of roles with instances of resources. We therefore need a more dynamic access management system than RBAC.

GOLD requires similar bindings since we want to be able to restrict role access on resources depending on the execution context. In a virtual organisation the execution context can be regarded as the particular project or the goal that all participants have come together to achieve. In a virtual organisation there can be many projects with various participants resulting to complicated interrelationships between them. For example, some of them may play the same role in various projects, carrying out the exact same tasks, or have different roles within the same project depending on the performed task.

The question we raise is 'Should a role have the same permission and access rights throughout the set of similar or even identical projects and the tasks within those projects?' Our view is that in dynamic access control systems we should separate roles from role instances. Different role instances may require different permissions and indeed additional levels of authorisation depending on the project and task in which they are active.

The main GOLD case study involves the collaboration of a number of participants to enable the development of chemicals. The domain itself requires the sharing of sensitive information between participants who may have conflicting interests. In

order to raise the levels of trust within such a VO we need to make sure that we have developed fine grained access control mechanisms. RBAC or other traditional techniques do not provide this level of granularity.

The main issues regarding access control relate to the degree of granularity embedded in the controlling mechanism itself. By granularity we refer to the level of detail for which we are prepared to define access rights. In GOLD, the simple subject-object permissions model on which RBAC is based is not sufficient. We need fine grained permissions for instances of roles as well as instances of objects. For example, a chemist role may be granted access to chemical documents but we do not however wish to grant access to all chemical documents produced by the system. Instead, we want any access permissions granted to the chemist role to be project-specific (e.g., the instance of a particular collaboration) as well as task-specific (e.g., the instance of a particular pre-defined activity). The management of roles and access permissions in GOLD is integrated with the management and monitoring of dynamic service level agreement or contracts between the participating services. The contracts can capture the expectations from specific tasks, using pre- and post-conditions. Permissions for roles can be activated and de-activated based on the progress of the monitored contracts.

4 Our Approach

4.1 Requirements

We need to develop a framework that defines conceptual boundaries around projects and tasks so that the roles and permissions can be scoped. Since a VO is comprised from a set of services, messages between those services need to propagate context-related information that can be used to identify the specific scope in which the security-related decisions can be made. Services can use the context information to determine whether the requestor has sufficient permission to perform operations or access resources. Other levels of requestor verification, such as authentication, could be implemented using existing methodologies (e.g., certificate-based authentication) and RBAC using access lists associated with every role.

The framework must allow for the dynamic activation and deactivation of permissions and roles based on progress monitoring of projects and tasks based on established contracts. For example, if a role fails to deliver its obligations within a task, we may want to de-activate certain permissions to that role. Also, if a task is completed successfully, we may want to add more permissions to a particular role so that resources and actions in a following task become accessible. In essence, the effective permissions are linked with the scope of the activity being performed.

4.2 Solution

The solution we propose follows the same model as previous works in handling transactions in distributed systems [6], the concept of spheres of control [7], and the coordinated atomic actions concept [8]. Coordinated Atomic (CA) Actions is a unified scheme for coordinating complex concurrent activities and supporting error recovery between multiple interacting components in distributed systems. CA Actions can be regarded as providing some of the generality of spheres of control and within them they offer a full support for maintaining consistency and achieving fault tolerance. They have been successfully applied [6, 9] as a structuring mechanism and they are relevant because they allow us to design, structure and provide fault tolerance in GOLD where autonomous organisational entities co-operate. We distinguish between role instances, resource instances as well as the notions of projects and tasks (. We are implementing the conceptual elements of this solution using WS standards. We discuss these in section 6.

5 Project-Task Structure

In order to achieve the level of granularity mentioned earlier we need to provide several levels of access control enforcement. In the first instance these manifest themselves as boundaries; that is project and task boundaries. The task boundary encapsulates all roles and objects related to an atomic action or activity. A project boundary encapsulates all tasks – atomic and otherwise – that take place within a GOLD VO in order to achieve a common goal. The concept of task is an implementation of the concept of spheres of control and is employed to provide a conceptual boundary encapsulating a specific action. Within its boundaries we find the following:

- a task name, i.e. unique identifier of a particular instance of a task;
- a set of roles;
- objects/resources used by that task; and
- reference(s) to resource(s) relevant to the task outside the tasks boundary.
- Pre-conditions, Post-conditions.

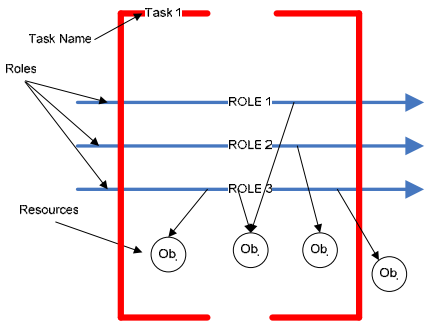

Figure 1. The structure of a task

Figure 1 provides a conceptual representation of a task within GOLD. The box denotes the boundaries of the task within which the instances of roles and the set of access permissions are active. This logical task can span across multiple GOLD services. Each of the resources/objects in Figure 1 is associated with access permissions. The system evaluates access to the resources according to the role that attempts to perform an action always within the context of a particular task.

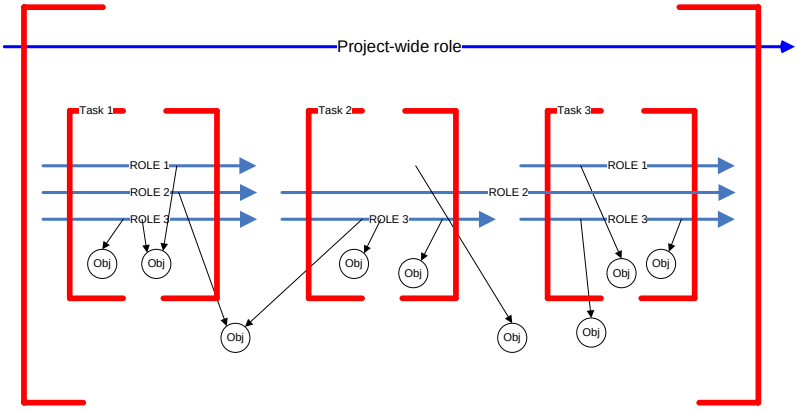

Figure 2. Project structure

The Figure 2 illustrates the relationship between projects, tasks, role instances, and the objects/resources. In the context of GOLD, the project would be a VO that is established for the development and market exploitation of a chemical compound while a task could be a chemical hazard analysis. Objects that are shared between tasks (as project-bound objects) can be VO-wide documents while task-bound objects can be temporary notes used for the completion of that task. Roles may evolve according to the contracts put in place to manage the VO. For example, an employee from one company with an assigned role *L* may have access to some of the documents from another company during task *B* but will be denied access to them during the preceding task *A*. The contracts that are in place to establish a VO determine the roles and their permissions for each of the task that VO is to perform. The runtime support for the VO monitors the progress of each task in relation to the contracts in place and dynamically changes the permissions of the roles where appropriate.

6 Implementation Approach

The test bed used to experiment with the concepts investigated by GOLD is built using Web Services (WS) technologies and follows the principles of service-orientation [10-14]. Each company/partner in a GOLD virtual organisation is represented as a service and communication between them is facilitated through the exchange of messages. Here we describe the design part of the GOLD test bed that deals with the task-based access control issues discussed in this paper.

6.1 Representing Tasks Using Context

Since a task may span any number of services within a particular virtual organisation, information needs to be propagated in messages so that the receiving services can determine the scope of the interaction. The context-based approach to scoping message exchanges are used in a number of WS specifications (e.g., WS-SecureConversation [15], WS-Coordination [16], WS-AtomicTransaction[17], etc.). Industry vendors have proposed WS-Context [18] as a standard way for representing context-related information, its management, and its inclusion in messages. In GOLD, we propose the use of WS-Context (or a similar specification) to model a task that spans services. A GOLD task will be represented by a context structure which would be propagated with every message being exchanged within the scope of that task. Since a GOLD service may be used by many VOs at the same time, each VO has to be identified through a context structure and since a VO may have multiple tasks, each one is represented by a context structure. There is a parent-child relationship between a VO context and a task context. Furthermore, a task may be composed of sub-tasks, so the context structure needs to support this. Listing 1 shows an example of a pseudo XML structure to represent the interaction context within GOLD.

```
<wsctx:context>
  <gold:vo>
    <!-- General information about the VO or just an identifier
or a reference. -->
    <gold:task>
      <!-- General information about the task or just an
identifier or a reference. -->
      <!-- Perhaps some sub-tasks -->
    </gold:task>
  </gold:vo>
</wsctx:context>
```

Listing 1. Pseudo XML structure of the context for message exchanges within a GOLD VO

6.2 Security

A combination of WS technologies [19] could be used to meet GOLD's security requirements. For example, WS-Security [20] defines the mechanisms for exchanging security tokens, message signing, and message encryption. WS-Trust [21] can be used when retrieving security tokens for authentication/authorisation purposes from trusted sources, WS-Federation [22] for federating identities and security attributes across different trust realms, SAML [23] for defining security assertions, XACML [24] for describing authorisation policies and making policy-based decisions, etc.

Here we are concentrating only on the task-based access control aspects of security and those specs that we can use for our implementation. We propose that XACML is used within GOLD to define access policies on objects/resources for authorisation purposes (what actions are allowed to be performed on a resource by roles) while each role can be represented using SAML assertions. Security-related information associated with a requestor (the user belonging to a role) can be propagated with

every message so that services can reason about the requestor's security-related claims and determine, based on policy assertions, whether access to local resources should be granted within the scope of the particular task. We achieve task-based access control through the dynamic association of the SAML assertions with the context.

6.3 Example

An example of a pseudo SOAP [25] message carrying context and security information is presented in Listing 2.

```
<soap:Envelope>
  <soap:Header>
    <wsse:Security>
      <wsse:BinarySecurityToken>
        <!-- X.509 certificate -->
      </wsse:BinarySecurityToken>
      <saml:Assertion>
        <!-- Assertions about the requestor's roles
             (e.g., Chemist, Manager, etc.) -->
      <saml:Assertion>
    </wsse:Security>
    <wsctx:context>
      <gold:vo>
        <!-- information about the VO. -->
        <gold:task>
          <!-- information about the task. It could be just an
identifier.
               It may contain sub-tasks. -->
        </gold:task>
      </gold:vo>
    </wsctx:context>
  </soap:Header>
  <soap:Body>
    <gold:ChemicalAnalysisRequest>
      <!-- Task-specific information -->
    </gold:ChemicalAnalysisRequest>
  </soap:Body>
</soap:Envelope>
```

Listing 2. A pseudo SOAP message in GOLD carrying context and security information

When a GOLD service receives a message like the one of Listing 2, it uses the SAML assertions about the role of the requestor and the context representing the VO and the task being performed in that VO to determine whether the request can be satisfied. The decision may use a Policy Decision Point (PDP) as per the SAML specification [23].

In the example of Listing 2, we can assume that the message is signed using the X.509 certificate of the requestor and encrypted using the X.509 certificate of the receiving service, as per the WS-Security [20] specification. Also, WS-SecureConversation [15] could be used to improve upon the performance of a multi-

message conversation. Lastly, a SAML authentication-specific context could be used to replace the need for X.509 certificates and, hence, allow a federated authentication scheme to be adopted within GOLD.

7 Conclusion

In this paper we discussed task based access control as a mechanism for dynamic virtual organisation scenarios where roles and access right policies continuously evolve according to the contracts put in place. Traditional role based models are static and therefore inadequate in such modern dynamic environments. We proposed a solution based on concepts such as spheres of control and coordinated atomic actions to structure our system so that trust emerges as a system property. Current WS standards enables us to quickly develop such a mechanism as we can map our conceptual elements on standardised XML schemas and WS protocols. Furthermore they allow us to dynamically manage access rights depending on the progress of a particular activity such as a transaction. Access rights can therefore be awarded or withdrawn depending on progress.

References

[1] T. Anderson, A. Avizienis, and W. Carter, "Dependability: Basic Concepts and Terminology," in Series: Dependable Computing and Fault-Tolerant Systems Volume 5, J.-C. Laprie, Ed. New York: Springer-Verlag, 1992.

[2] G. Coulouris and J. Dollimore, "Security Requirements for Cooperative Work: A Model and its System Implications," presented at 6th Workshop on ACM SIGOPS European Workshop: Matching Operating Systems to Application Needs, Wadern, Germany, 1994.

[3] K. T. Roshan and R. S. Sandhu, "Task-Based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-Oriented Authorization Management," presented at IFIP TC11 WG11.3 Eleventh International Conference on Database Securty XI: Status and Prospects, 1997.

[4] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models," *IEEE Computer*, vol. 29, pp. 38-47, 1996.

[5] R. K. Thomas, "Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments," presented at Second ACM Workshop on Role-based Access Control, Fairfax, Virginia, United States, 1997.

[6] P. Periorellis and J. E. Dobson, "Case Study Problem Analysis. The Travel Agency Problem," University of Newcastle upon Tyne, Newcastle upon Tyne, UK 2001.

[7] C. T. Davies, "Spheres of Control," *IBM Systems Journal*, vol. 17, pp. 179-198, 1978.

[8] A. Romanovsky, "Coordinated Atomic Actions: How to Remain ACID in the Modern World," *ACM SIGSOFT Software Engineering Notes*, vol. 26, pp. 66-68, 2001.

[9] A. F. Zorzo, P. Periorellis, and A. Romanovsky, "Using Coordinated Atomic Actions for Building Complex Web Applications: a Learning Experience," presented at 8th IEEE International Workshop on Object-oriented Real-time Dependable Systems (WORDS 2003), Guadalajara, Mexico, 2003.

[10] "Service-Oriented Architecture (SOA) Definition." http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html.

[11] D. F. Ferguson, T. Storey, B. Lovering, and J. Shewchuk, "Secure, Reliable, Transacted Web Services: Architecture and Composition." http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnwebsrv/html/wsoverview.asp, 2003.
[12] H. He, "What is Service-Oriented Architecture." http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html, 2003.
[13] S. Parastatidis, J. Webber, P. Watson, and T. Rischbeck, "WS-GAF: A Grid Application Framework based on Web Services Specifications and Practices." Submitted for publication, 2004.
[14] D. Sprott and L. Wilkes, "Understanding Service-Oriented Architecture." http://msdn.microsoft.com/library/en-us/dnmaj/html/aj1soa.asp, 2004.
[15] "Web Services Secure Conversation Language (WS-SecureConversation)." http://msdn.microsoft.com/ws/2004/04/ws-secure-conversation/, 2004.
[16] "Web Services Coordination (WS-Coordination)." http://msdn.microsoft.com/ws/2003/09/wscoor, 2003.
[17] "Web Services Atomic Transaction (WS-AtomicTransaction)." http://msdn.microsoft.com/ws/2003/09/wsat, 2003.
[18] OASIS(WS-CAF), "Web Services Context (WS-CTX)." http://www.iona.com/devcenter/standards/WS-CAF/WSCTX.pdf.
[19] J. Rosenberg and D. Remy, *Securing Web Services with WS-Security*. Indianapolis: Sams Publishing, 2004.
[20] OASIS, "Web Services Security (WS-Security)." http://www.oasis-open.org/committees/wss.
[21] "Web Services Trust Language (WS-Trust)." http://msdn.microsoft.com/ws/2004/04/ws-trust/, 2004.
[22] "Web Services Federation Language (WS-Federation)." http://msdn.microsoft.com/ws/2003/07/ws-federation/, 2003.
[23] OASIS, "Security Assertion Markup Language (SAML) v2.0." http://www.oasis-open.org/committees/security, 2004.
[24] OASIS, "Extensible Access Control Markup Language (XACML)." http://www.oasis-open.org/committees/xacml.
[25] W3C, "SOAP Version 1.2 Part 1: Messaging Framework," in *W3C Recommendations*, M. Gudgin, M. Hadley, J.-J. Moreau, and H. F. Nielsen, Eds., 2003.