Injecting RBAC to Secure a Web-based Workflow System

Gail-Joon Ahn,[¶] *Ravi Sandhu*,[¶] *Myong Kang*[||] and *Joon Park*[||]

[¶]ISE Department, MS 4A4
George Mason University
Fairfax, VA 22030
{gahn,sandhu}@isse.gmu.edu

[||]Naval Research Laboratory
Information Technology Division
Washington, DC, 20375
{mkang,jpark}@itd.nrl.navy.mil

Abstract

Web-based workflow systems have recently received much attention because they can support dynamic business processes over heterogeneous computing systems. Most existing web-based workflow systems, however, provide minimal security services such as authentication of users and network security. In this paper we describes an experiment in injecting role-based access control (RBAC) into an existing web-based workflow system. Specifically, we ensure that each task can only be executed by users belonging to a specific role. In order to achieve this, we define a simplified RBAC model to meet our needs and describe the security architecture to be applied to an existing web-based workflow system. We describe our implementation using commercial off-the-shelf (COTS) technology to demonstrate the feasibility of this approach. Our implementation uses X.509v3 certificates with role attribute, and employs a user-pull style where the client requests a client certificate from the role-server and presents it to the workflow system. A major goal of our implementation is to have minimal changes to the existing web server and no changes to the browser. We also discuss alternative architecture such as server-pull with LDAP (Lightweight Directory Access Protocol).

1 INTRODUCTION

Since the earliest work in information security, it has been understood that the greatest threat is from insiders. This fact continues to be confirmed even in the Internet era of ubiquitous connectivity. Security of insider access is achieved by enforcing least privilege, adhering to organizational policy, enforcing separation of duties, separating administration and access, and enforcing access control in terms of application abstractions (such as credit and debit operations) rather than primitive reads and writes. Moreover, insider access is often best determined by a user's roles and job functions in the organization rather than by individual identity. Role-based access control (RBAC) has become widely accepted as the proven technology for this purpose.

Web technology has continued its rapid evolution, most recently towards increasingly distributed applications. The marriage of web and workflow system is one of the results of this trend [Den96]. Several web-based workflow systems have been introduced [SJKB94, EGL97, VW97]. These systems use web technology as a user interface, a gateway to external applications, messaging tool, or workflow specification tool. However, most existing web-based workflow systems provide minimal security services such as authentication of users. While the protection of transmitted data over the network by means of protocols such as SSL has been practiced, access control on workflow activities has not received much attention.

Huang and Atluri [HA99] introduced a web-enabled workflow management system called SecureFlow. This work showed that the security specification and enforcement modules could be placed on top of existing workflow system to provide security with the notion of role-based access control. This system relies on the workflow authorization model which uses the notion of an authorization template to specify the static parameters of the authorization.

Park and Sandhu [PS99] described how we can use role information on the web using *smart certificates.* Even though the application domain was not workflow systems per se, their work showed that role information can be used to authorize web-based transactions between a client and a web server. Unlike [HA99] the authorization was carried out within a web server. The implementation was platform-dependent using built-in Windows NT group mapping mechanism. Here we focus on showing that the authorization can be achieved in a platform-independent manner.

Our objective in this paper is to show how to inject role-based access control into an existing web-based workflow system. We define a simplified RBAC model to meet our needs and describe the security architecture to be applied to an existing web-based workflow system. We demonstrate the feasibility of this approach by implementing it using commercial off-the-shelf (COTS) technology. Our implementation uses X.509v3 certificates with role attribute and employs a user-pull style where the client requests a client certificate from the role-server and presents it to the workflow system. A major goal of our implementation is to have minimal changes to the existing web server and no changes to the browser.

The paper is organized as follows. Section 2 briefly discusses our security objectives. In section 3, we describe the RBAC model customized for our purpose. Section 4 describes the security system architecture for an existing web-based workflow system. Section 5 discusses its implementation using COTS technology. In section 6, we discuss alternative approaches. Section 7 concludes this paper.

2 SECURITY OBJECTIVES

A workflow is an activity involving the coordinated execution of multiple tasks performed by different processing entities [KA95]. These tasks could be manual or automated in nature. A workflow process is an automated organizational process involving both human and automated tasks. A workflow management system (WFMS) is a set of tools that provide support for process definition, workflow enactment, and administration and monitoring of workflow processes [Hol95]. With the emergence of web technology, web-based workflow systems are being deployed over the enterprise computing environment.

We can consider several security services for web-based workflow systems such as authentication of the user, network security for data transport, and access control. Using authentication services we can identify a user who participates in web-based workflow systems. Once authentication is accomplished we need to enforce that only authorized users can execute appropriate tasks of the workflow.

In this project our workflow platform was the NRL-University of Georgia (NRL-UG) [KFS+99] web-based workflow system built upon METEOR WFMS [MPS+98]. This system implements client-server interaction by means of HTTP, while server-to-server interaction uses CORBA's IIOP protocol. A workflow consists of a number of tasks. Each task of the workflow can be carried out on a different server, and each server can be dedicated to a single task. Some tasks require human intervention; others are carried out automatically. Each human task must be restricted to users belonging to a specific role in an organization. The existing system had a simple authentication functionality for security services but no authorization whatsoever.

Our objective is to inject RBAC into the NRL-UG workflow system in a manner that avoids modification of the underlying workflow system. The central goal is to keep a clear boundary between the RBAC component and the workflow component while integrating these in a seamless manner. In addition our implementation seeks to have minimal changes to the existing web server and no changes to the browser. In summary, we ensure that each task can only be executed by users belonging to a specific role.

3 THE RBAC MODEL

In order to enforce RBAC on our platform we adapt the well-known RBAC96 model defined by Sandhu et al [SCFY96]. Figure 1 shows roles and permissions that regulate access to data and resources. Intuitively, a user is a human being or an autonomous agent, a role is a job function or job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of the role. RBAC96 does not provide any interpretation of permissions. Permissions are simply treated as abstract tokens. For our purpose, we define a permission to be authorization to execute a task in a workflow system. We may consider permissions and tasks with various notions. Each web-based task server is associated with a

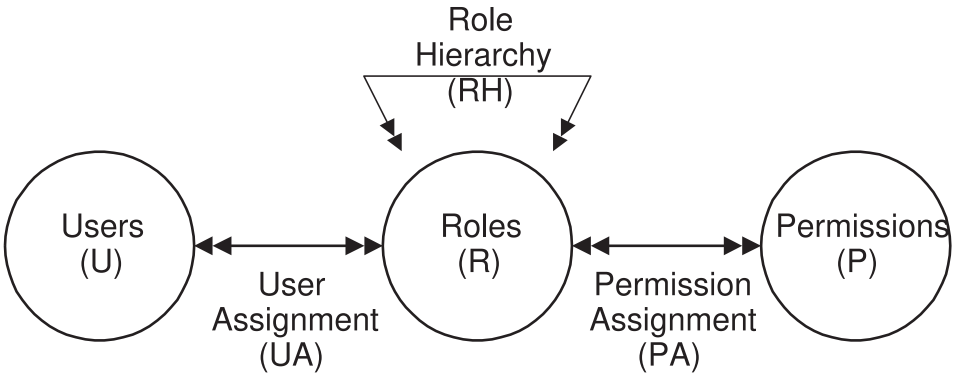

Figure 1: A simplified RBAC model

task and the authorization is enforced in terms of roles. Each task can only be executed by users belonging to a specific role. Roles are organized in a partial order $\geq$, so that if $x \geq y$ then role x inherits the permissions of role y. Members of x are also implicitly members of y. In such cases, we say x is senior to y. This simplified RBAC model has the following components and these components are formalized from the above discussions.

- U is a set of users,
- R is a set of roles,
- P is a set of permissions, $P = \{(exec, t) \mid t \in T\}$, where T is a set of tasks.
- $UA \subseteq U \times R$, is a many-to-many user to role assignment relation,
- $PA \subseteq P \times R$, is a many-to-many permission to role assignment relation,
- $RH \subseteq R \times R$, is partially ordered role hierarchies (written as $\geq$ in infix notation)

A user can be a member of many roles and a role can have many users. Similarly, a role can have many permissions and the same permissions can be assigned to many roles.

4 SYSTEM ARCHITECTURE

Park [Par99] identified two different approaches for obtaining a user's attributes on the web called user-pull and server-pull styles. These styles are shown in Figures 2 and 3. In user-pull style, the user pulls appropriate attributes from the attribute server and then presents them to the web servers to gain access. In server-pull style, each user presents only authentication information to web servers, and each web server pulls user's attributes from the attribute server as needed and uses them for authorization. This latter style is more convenient for users but less convenient for web servers than user-pull style.

In this paper we focus on a user-pull style (Figure 2), in which the client requests a client certificate from the role server and present this certificate to the workflow system in order to execute tasks. In section 6 we briefly discuss how we can use server-pull style to meet our demands as an alternative architecture.

The system architectures illustrated in Figures 2 and 3 have three components as follows: *workflow design tool*, *role server*, and *web-based workflow system*.

The workflow design tool assists a workflow designer in designing workflow applications. This encompasses specification of information flow and dependencies among tasks, creating roles and role-hierarchies, and assigning a role to each task. The design tool exports information regarding the role hierarchy to the role server, and information regarding task-role assignment and task dependencies to the workflow system.

The role server has two major components–*user-role assignment* and *certificate server*. The user-role assignment component maintains role hierarchies, assigns users to roles, and generates and maintains the user-role assignment database. Certificate server authenticates clients, retrieves client's role information from user-role assignment database, and issues certificates with client's role information. The user-role assignment component uses the exported role hierarchy from the design tool.

The workflow system consists of web-based task

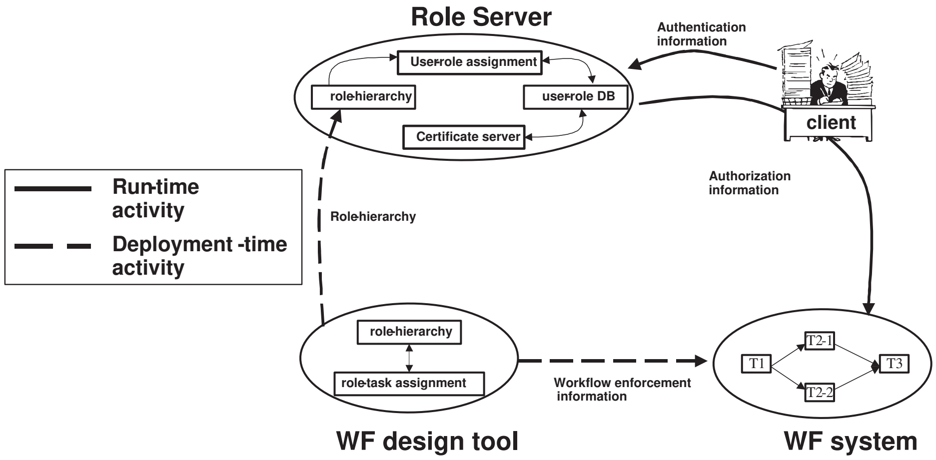

Figure 2: Security Architecture for Secure Workflow (WF) System: User-Pull Style

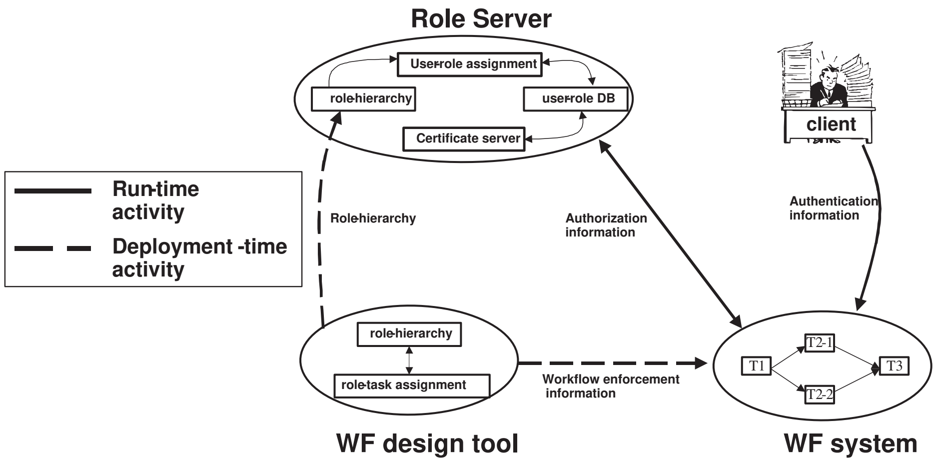

Figure 3: Security Architecture for Secure Workflow (WF) System : Server-Pull Style

servers. The central goal of our work is to enhance the existing NRL-UG web servers to enforce authorization in terms of roles so that a user can execute a task only if the user holds the appropriate role. Once a user requests resources to web servers, he/she presents his/her certificate to the task server. The task server authorizes a user's privileges with respect to the certificate with respect to the role after the user is authenticated. If the authorization check is successful the user is allowed to execute the requested task. The conceptual process of this authorization is described in Figure 4.

5 IMPLEMENTATION DETAILS

In our implementation of the user-pull architecture we make use of previously existing components, modifying them as appropriate to meet our objectives. The workflow design tool and workflow system were previously implemented by NRL and University of Georgia, respectively. For role server component, we use the URA97 implementation previously done at GMU [SP98]. Thus our implementation includes the preexisting components as follows: Workflow design tool (from Naval Research Laboratory), Role server (from Laboratory for Information Security Technology at GMU), and Workflow management system (from University of Georgia).

5.1 Workflow Design Tool

The NRL workflow design tool assists a workflow designer in performing the following functions.

- Create a workflow domain,
- design workflow applications specifying information flow and dependence among tasks,
- create roles and role hierarchies, and
- assign a role to each task.

All these functions are performed through graphical interfaces. Based on this workflow design, a specification for workflow runtime is generated. The design tool exports information regarding the role hierarchy to the role server, and information regarding task-role assignment and task dependencies to the workflow system. In the NRL implementation role hierarchies are expressed in XML (eXtensible Markup Language). The design tool is integrated with GMU's role server as described in next section.

5.2 Role Server

The role server has two major components: *user-role assignment* and *certificate server*. The user-role assignment component maintains role hierarchies, assigns users to roles, and generates and maintains the user-role assignment database. Certificate server authenticates clients, retrieves client's role information from user-role assignment database, and issues certificates with client's role information.

5.2.1 User-Role Assignment

The user-role assignment component was previously implemented at GMU using a model called URA97 (user-role assignment '97) [SB97]. This implementation [SP98] has its own interface for defining a role hierarchy. However, in this project we have integrated the NRL design tool with GMU's URA97 implementation. As mentioned earlier, the graphical input from the workflow designer is converted to an XML representation of the role hierarchy. This XML representation is then translated into the internal format of the GMU's URA97 implementation. This role hierarchy is used to populate a user-role database for use by the certificate server. In the current implementation all roles of a user are explicitly assigned in the user-role database. In other words the role hierarchy is simulated be explicit user-role assignment. This simplifies the task of the certificate server.

5.2.2 Certificate Server

The workflow system contains web-based task servers. Each task server authorizes client's privileges with respect to the certificate in terms of role. This authorization occurs during SSL establishment between client and server. To facilitate this we have developed an enhanced certificate server built around COTS certificate engines (specifically, Microsoft engines).

We use standard X.509v3 digital certificates. Existing COTS certificate engines do not support roles. We developed a novel approach to insert role information in the certificate, as shown in Figure 5. The client accesses the GMU-developed certificate server and provides authentication information (user name and password). After successful authentication the certificate server sends username to role server, which maintains client's information including roles assigned to client. The role server picks up the proper information for the client and generates a certificate enrollment form, which is sent to certificate server. For the moment, we are using the organization unit (OU) attribute in X.509v3

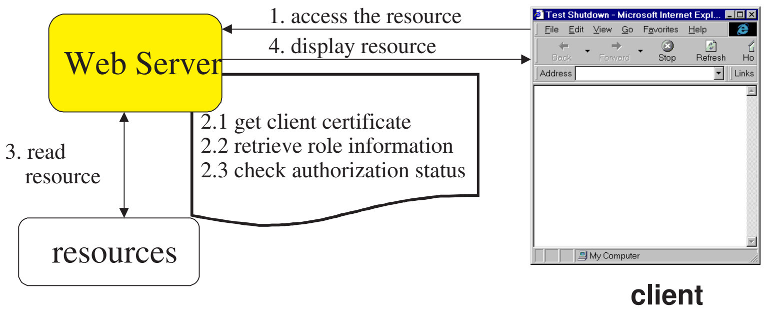

Figure 4: Authorization on Workflow servers

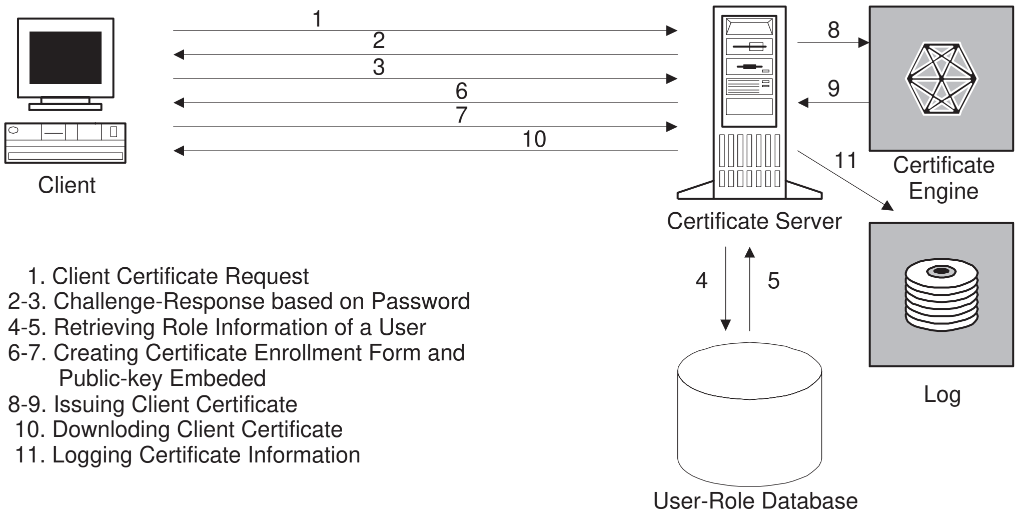

Figure 5: Certificate Issuing Procedures

certificate as the role information unit. According to certificate engine configuration, certificate server issues the client certificate based on the given enrollment form from role server. The certificate enrollment form which is modified for using role attribute is illustrated in Figure 6. This enrollment form embeds Javascript codes which trigger certificate engine in Microsoft certificate server.

5.3 Workflow Server

So far we have discussed the design tool and role server, describing how we can put role information in the client certificates. Now we look at how we can use those certificates in task servers. In the NRL-UG workflow system, each task scheduler has a web server—a simple Java implementation—as a task server.

Our goal is to show how we could use the client certificates with role information in the web server (task server). The web server requests a proper certificate from the client (for authentication during SSL negotiation) and then retrieves role information from the certificate to check if the client has required privileges. In order to support this, we need a secure connection (SSL) between the clients and task servers. Since we do not want to change the web servers at this point, we decided to use a reverse proxy server between the clients and task servers (UG's web servers)[1]. A usual proxy web server has client behind a firewall and server outside the firewall. Reverse proxy reverses this relationship. It can be used outside the firewall to represent a secure content server to outside client, preventing unmonitored access to web server's data from outside an organization. In order to achieve this, we selected IAIK's Jigsaw-SSL, an implementation of SSL on Jigsaw web server (developed by W3C) as our reverse proxy server to support security services between the clients and task servers [WWWC99, faipcI99].

To use client certificates with role information in Jigsaw-SSL, we modified several classes in the original package as below.

1. Get client certificate
   `X509Certificate[] certChain = Certificate.getCertificateChain()`

2. Get certificates content
   `certChain[0].getSubjectDN().toString()`

3. Retrieve role information
   `getOUinfo(String s)`

4. Compare this role with the required role
   `Role.equalsIgnoreCase()`

We also modified Jigsaw-SSL's source codes to build proxy server supporting authorization functionality. The proxy server retrieves client's role information and compares it with expected roles. We have small modifications of web server to monitor incoming network traffic because there is an insecure regular channel between a proxy server and a task server while our implementation establishes a secure channel for communication between a client and a proxy server. We developed new Java class which can support IP (Internet Protocol) filtering functionality so that we can monitor whether or not all incoming requests are from our proxy server. Figure 7 summarizes our implementation based on the security architecture mentioned in section 4.

6 DISCUSSION

We first defined the security objectives of this work. Based on this security objective, we formulated the simplified RBAC model to meet our needs. Given RBAC model, we developed a security architecture. In order to demonstrate feasibility of this architecture, we implemented the prototype using COTS. Thus our research followed four steps such as *objectives*, *model*, *architecture*, and *mechanism* corresponding to the OM-AM framework [San00]. In this way we were able to successfully demonstrate the feasibility of injecting RBAC into an existing web-based workflow system using COTS technology and minimal changes to a web server.

So far this project has developed a successful prototype, based on user-pull style, to show how role certificates can be used by Java-based and SSL-enabled web-servers to verify authorization of users for tasks by means of roles. Instead of exchanging role attribute between clients and role server, we may consider alternative architecture such as server-pull style wherein the user goes directly to the workflow system and presents authentication credentials. The authorization credentials for that user are then obtained by the workflow-server from the role-server. User information such as a user's digital certificate is often fragmented across the enterprise, leading to data that is redundant, inconsistent, and expensive to manage. Directories are being viewed as the one of best mechanisms to make enterprise information available to multiple different systems within an organization. Directories also make it possible for organizations to access information over the Internet. The trend towards directories has been accelerated by the recent growth of the LDAP (Lightweight Directory Access Protocol). We can re-

[1] There may exist other approaches. Our approach is one of possible alternatives.

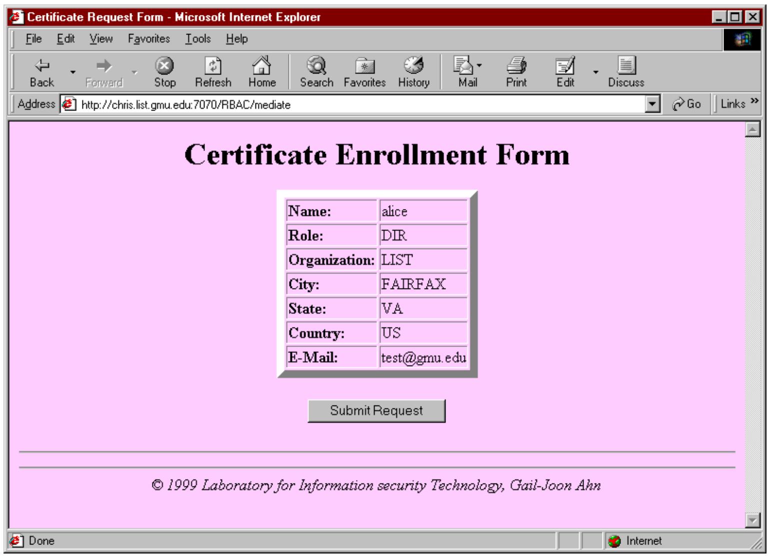

Figure 6: Certificate Enrollment Form

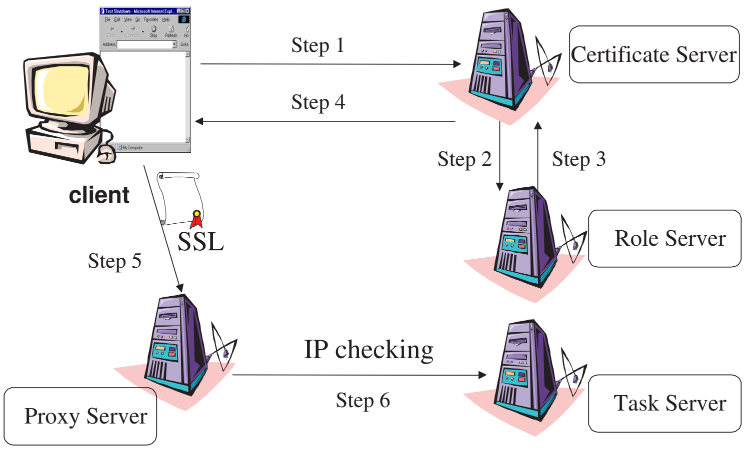

Figure 7: Implementation Scenario

place role-server with directory-oriented role-server using LDAP for task-server to role-server communication. This server-pull style is currently under study.

In addition, we need to investigate the security architecture that can support more sophisticated security objectives such as dynamic separation of duty. Also the role information in an organization can be changed.

7 CONCLUSION

In this paper we developed models, architectures and mechanisms for specifying and enforcing role-based authorization models for web-based workflow systems. We also developed a proof-of-concept implementation to demonstrate the practical feasibility of this technology. Also another research direction was discussed. Even though this work is applied to an existing web-based workflow system, we believe that this architecture can be deployed into several application domains such as large-scale collaboratory environments and electronic commerce systems.

Acknowledgment

The work of Gail Ahn and Ravi Sandhu was partially supported at GMU by NRL.

References

[Den96] Peter J. Denning. Workflow in the WEB. In Layna Fischer, editor, *New tools for New Times: Electronic Commerce*. Future Strategies, Inc., 1996.

[EGL97] Johann Eder, Herbert Groiss, and Walter Liebhart. The workflow management system Panta Rhei. In Asuman Dogac, Leonid Kalinichenko, M. Tamer Ozsu, and Amit Sheth, editors, *Advances in Workflow Management Systems and Interoperability*, pages 129–144. NATO Advanced Study Institute, 1997.

[faipcI99] Institute for applied information processing and communications (IAIK). Jigsaw SSL. In *http://jcewww.iaik.tu-graz.ac.at/Applications/jigsaw.htm*, 1999.

[HA99] Wei-Kuang Huang and Vijayalakshmi Atluri. SecureFlow: A secure web-based workflow management system. In *Proceedings of 4th ACM Workshop on Role-Based Access Control*, pages 83–94, Fairfax, VA, October 1999. ACM.

[Hol95] D. Hollingsworth. The workflow reference model. Technical Report TC00-1003, The Workflow Management Coalition, Hampshire, UK, January 1995.

[KA95] N. Krishnakumar and A. Aheth. Managing heterogeneous multi-system tasks to support enterprose-wide operations. *Distributed and Parallel Databases*, 3(2):155–186, April 1995.

[KFS+99] Myong H. Kang, Judith N. Froscher, Amit P. Sheth, Krys J. Kochut, and John A. Miller. A multilevel secure workflow management system. In *Proceedings of the 11th Conference on Advanced Information Systems Engineering (CAiSE'99)*, pages 271–285, Heidelberg, Germany, June 1999.

[MPS+98] J. Miller, D. Palaniswani, A. Sheth, K. Kochut, and H. Singh. WebWork: METEOR's web-based workflow management system. *Journal of Intelligent Information Systems*, 10(2):185–215, March/April 1998.

[Par99] Joon S. Park. *Secure Attribute Services on the Web*. PhD Thesis, George Mason University (Adviser: Ravi Sandhu), August 1999.

[PS99] Joon S. Park and Ravi Sandhu. RBAC on the web by smart certificates. In *Proceedings of 4th ACM Workshop on Role-Based Access Control*, pages 1–9, Fairfax, VA, October 1999. ACM.

[San00] Ravi Sandhu. Engineering authority and trust in cyberspace: The OM-AM and RBAC way. In *Proceedings of 5th ACM Workshop on Role-Based Access Control*, 2000.

[SB97] Ravi Sandhu and Venkata Bhamidipati. The URA97 model for role-based administration of user-role assignment. In T. Y. Lin and Xiaolei Qian, editors, *Database Security XI: Status and Prospects*. North-Holland, 1997.

[SCFY96] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, February 1996.

[SJKB94] H. Schuster, S. Jablonski, T. Kirsche, and C. Bussler. A client/server architecture for distributed workflow management systems. In *Proc. Parallel and Distributed Information Systems Conf*, Austin, TX, 1994.

[SP98] Ravi Sandhu and Joon S. Park. Decentralized user-role assignment for web-based intranets. In *Proceedings of 3rd ACM Workshop on Role-Based Access Control*, pages 1–12, Fairfax, VA, October 1998. ACM.

[VW97] Gottfried Vossen and Mathias Weske. The WASA approach to workflow management for scientific applications. In Asuman Dogac, Leonid Kalinichenko, M. Tamer Ozsu, and Amit Sheth, editors, *Advances in Workflow Management Systems and Interoperability*, pages 145–165. NATO Advanced Study Institute, 1997.

[WWWC99] The World Wide Web Consortium. Jigsaw–the W3C's Web Server. In *http://www.w3c.org/jigsaw*, 1999.