SUPPORTING AUTHENTICATION REQUIREMENTS IN WORKFLOWS

Ricardo Martinho
School of Technology and Management, Polytechnic Institute of Leiria
Morro do Lena, Alto do Vieiro - 2411-901 Leiria, Portugal

Dulce Domingos
Faculty of Sciences, University of Lisbon
Campo Grande, Edifício CS - 1749-016 Lisboa, Portugal

António Rito-Silva
INESC-ID Software Engineering Group, Technical University of Lisbon
Rua Alves Redol 9, 6.° - 1000-029 Lisboa, Portugal

Keywords: Workflow authentication requirements, authorization constraints, Role-based access control.

Abstract: Workflow technology represents nowadays significant added value to organizations that use information systems to support their business processes. By their nature, workflows support the integration of different information systems. As organizations use workflows increasingly, workflows manipulate more valuable and sensitive data. Either by interoperability issues or by the value of data manipulated, a workflow may present several and distinct authentication requirements. Typically, information systems deal with their authentication requirements once, within their authentication process. This strategy cannot be easily applied to workflows since each workflow activity may present its own authentication requirements. In this paper we identify authentication requirements that workflows present and we propose to meet these requirements by incorporating authentication constraints into workflow authorization definitions. With this purpose, we extend a generic Role-Based Access Control (RBAC) model and we define an access control algorithm that supports and enforces authorization decisions constrained by authentication information.

1 INTRODUCTION

Due to the increasing need of being digitally enabled, organizations often adopt workflows and Workflow Management Systems (WfMSs) to model and execute their main business processes. Digital workflows have reduced the cost of operations in many companies by displacing paper and related manual routines. By using workflow technology many corporations not only cut costs significantly but also improve customer service.

In (Workflow Management Coalition, 1999) the WfMC defines workflow as a concept that is usually associated with the automation of a business process where documents, information and activities are passed among participants according to some set of rules. The business process is modeled under a process definition, also called a workflow definition. A workflow definition may be composed of other sub-workflow definitions, activities (manual or automated), rules and control data.

A WfMS is an information system that interprets workflow definitions and creates and manages workflow instances as they are executed. The enactment of a workflow definition may derive in one or more workflow instances, which in turn include one or more activity instances. Activity instances are the runtime representation of the activities identified in the workflow definition, and may include work items (activities allocated to a workflow user) or invoked applications (information systems used to support activity execution). A workflow user is a resource which performs the work represented by a workflow activity instance. This user is usually associated with a human being, but it may also be a group of human beings, an automated process or a computer system.

Typically, authentication is the initial security step for users who want to interact with information systems, in general, as well as WfMSs and their workflow activities. The goal of authentication is to confirm a user's asserted principal identity with a specified, or understood, level of confidence. Different

Martinho R., Domingos D. and Rito-Silva A. (2006).
SUPPORTING AUTHENTICATION REQUIREMENTS IN WORKFLOWS.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - ISAS*, pages 181-188
DOI: 10.5220/0002465701810188

authentication mechanisms provide different levels of confidence. For instance, an authentication mechanism that uses fingerprints provides typically a higher level of confidence than another one that uses a password (Kent and Millett, 2003).

However, for some workflow scenarios, users need to have several principal identities with different levels of confidence and/or authenticated by different means. Indeed, workflow activities may present different levels of confidence requirements depending on their functionality and/or the sensitiveness of the data they manipulate.

On the other hand, a workflow may model within its activities the invocation of different information systems (Hung and Karlapalem, 2003) and those systems may require distinct authentication mechanisms and protocols. For example, a workflow activity may invoke a web service that requires a user principal to be authenticated under the WSSecurity protocol.

Additionally, if authentication requires workflow history information, this information cannot yet exist when the initial authentication process occurs. For instance, if a workflow has an activity that requires a user principal to have a higher level of confidence than the user principal that has executed a previous activity, this constraint can only be evaluated if the previous activity has already been executed.

In this paper we firstly systematize authentication requirements that workflows present: workflows present authentication requirements involving user principals' context authentication; information that is already used for authorization purposes; and workflow history authentication information. To meet these authentication requirements, we propose to include authentication constraints into the definition of workflow authorizations. As WfMSs often use a Role-based Access Control (RBAC) model to protect workflow activities, we extend it to support authorizations with authentication-based constraints. Finally, to evaluate authorizations with authentication-based constraints, we also propose an algorithm for an access decision function. By implementing this algorithm, the access decision function is capable of informing the WfMS that it needs additional information to support an authorization decision, upon a request made to a workflow activity with authentication requirements.

The remainder of this paper is organized as follows: in section 2 we present related work. In section 3 we describe a workflow example with authentication requirements. In section 4 we systematize workflow authentication requirements, we define an RBAC extension that supports authorizations with authentication-based constraints and an algorithm to enforce these authorizations. Finally, in section 5 we present some conclusions and future work.

2 RELATED WORK

The idea of meeting authentication requirements within access control models is not new. In (Wang et al., 2004), the authors propose an RBAC model that includes authentication information in order to dynamically assign users to roles in an operating system environment. Access rights are determined through the concept of a user's authentication trustworthiness, which is bounded to the strength of the authentication mechanism used. In (Moodahi et al., 2004) the authors present a workflow access control model that also uses authentication information to constrain the assignment of users to roles (UA relationship). These approaches handle authentication information in the initial authentication process, not solving the problem of meeting various workflow authentication requirements that may emerge as workflows are loaded and instantiated after that initial process. Another disadvantage is that constraining the UA relationship based on authentication information may lead to role proliferation (like having "password manager" and "certificate-based manager" roles).

In (Tzelepi and Pangalos, 2001) the authors propose an extension to the RBAC model to protect medical imaging files stored on a database. They support the definition of authorizations that include constraints with domain and location authentication attributes. In our work we go one step forward by supporting generic authentication attributes as well as workflow specific authentication information (the history information).

As RBAC models are widely used in WfMSs, our purpose is to incorporate authentication-based constraints into the definition of RBAC authorizations. In the following, we overview the RBAC model and how some authors have extended this model to support constraints.

The basic notion of RBAC is controlling access to resources accounting for the roles that users undertake, rather than on an individual user basis. This way complexity is reduced because roles are usually less than users in an organization. Furthermore, revoking and re-granting authorizations are avoided when users change their roles within the organization.

Important contributions to standardize a generic RBAC model may be found in (Ferraiolo et al., 2001). The generic architecture of this model is presented in Figure 1.

The main components depicted are users, sessions, roles, roles hierarchy (RH), permissions, user-assignment (UA) relationships and permission assignment (PA) relationships. It also comprises separation of duty constraints, either being static (SSD) between users and roles, or dynamic (DSD) between sessions and roles.

In general, to increase authorizations' expressive-

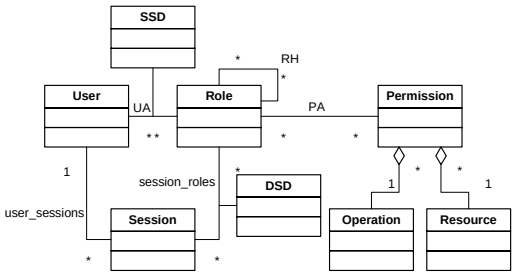

Figure 1: Generic RBAC model.

ness, access control models support the definition of authorizations with constraints to restrict their validity (Samarati and di Vimercati, 2000). Constraints may refer to different kinds of information, such as (Beznosov, 1998):

- User attributes, like age or nationality;
- Object attributes (content-based access control);
- External conditions like access time and location (context-based access control);
- Access history (history-based access control); and
- Component relationships.

In (Kandala and Sandhu, 2002) the authors extend the RBAC model to WfMSs. They interpret the permission concept by identifying the objects of the system that need protection (activities and activity instances) and the operations that can be executed on them. The main purpose of workflow access control models is to increase authorization expressiveness with constraints that are workflow specific: constraints that support dynamic separation and junction of duty policies. These constraints define security policies that prevent or force a certain user to execute two or more correlated activities (Casati et al., 1998; Bertino et al., 1999; Casati et al., 2001). However, these approaches do not address authentication-based constraints inferred from workflow authentication requirements.

3 WORKFLOW AUTHENTICATION REQUIREMENTS: AN EXAMPLE

In this section we exemplify different types of authentication requirements that may be presented by a workflow and related activities, by using a simplified Loan Approval workflow (Figure 2).

Authorizations are defined taking into account the roles that users play in the bank. Each role has the following authorizations: *Branch clerks* may a_1) receive

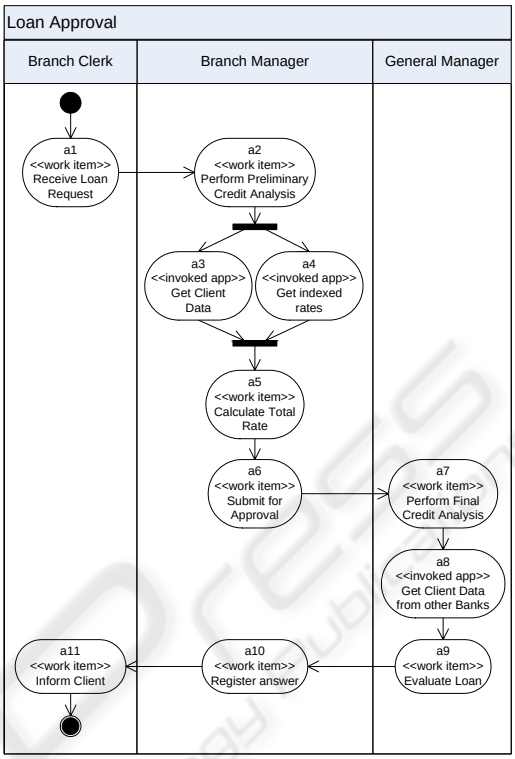

Figure 2: Loan Approval workflow example.

loan requests from clients and check the corresponding data; and a_{11}) inform clients about the result of their loan requests. *Branch managers* may a_2) perform preliminary credit analysis; launch invoked applications to a_3) retrieve data from clients and a_4) get indexed international rates; a_5) calculate the total loan rate; a_6) submit loans for approval; and a_{10}) register loan approval answers. *General managers* may a_7) perform final credit analysis on clients; a_8) invoke an application to retrieve data from other financial institutions; and a_9) evaluate the loan viability.

Now lets consider the following authentication requirements for the workflow shown in Figure 2:

- R_1 - All workflow activities require that users possess an authenticated principal identity belonging to a *bank.org* domain;
- R_2 - Activity a_3 requires a user principal identity authenticated by an identity provider *iDP*, with an authentication service at *http://www.idp.org/authnService*, and with a stronger mechanism than the one of the user principal identity which executed activity a_1.
- R_3 - Activity a_4 requires a user principal identity authenticated with a password over an Secure Sockets Layer (SSL) protected session;
- R_4 - If the loan value involved in the workflow in-

stance is higher than $100,000 then activity a_9 requires a user principal identity authenticated under a two-factor authentication mechanism, being composed by a smartcard and an activation PIN;

- R_5 - Activity a_{11} requires the same user principal identity on which behalf was granted access to activity a_1.

As illustrated by this example, workflows may present authentication requirements based on different types of information. In the next section we systematize workflow authentication requirements and propose an access control model that supports the definition of authorizations with authentication-based constraints. We also present an access control algorithm that is capable of identifying additional needed information, in order to evaluate authorization decisions that include authentication requirements.

4 DEFINING AND SUPPORTING WORKFLOW AUTHENTICATION REQUIREMENTS

Our approach to support workflow authentication requirements includes three correlated issues:

1. Systematizing workflow authentication requirements - this includes identifying and categorizing requirements by the type of information that they can use;
2. Defining an extended RBAC model to support authorizations that include authentication-based constraints; and
3. Enforcing authorization decisions that include authentication-based constraints.

In the next sections we expand and propose solutions for each of these items.

4.1 Systematizing Workflow Authentication Requirements

To systematize workflow authentication requirements we take the following approach:

1. **We identify generic authentication context information** - this refers to pure technical authentication context information that a user principal identity must meet in order to access a workflow activity. This information may expand on (OASIS, 2005):
   - User principal information - information about the principal identifier (e.g. *foo*), domain (e.g. *bar.net*) and group membership from where a user's digital entity is registered (e.g. an LDAP-enabled identity provider). Examples include a workflow activity a_1 with an authentication requirement stating that authorization permissions are only granted to a principal with the identifier *foo@bar.net*; or a workflow w_1 that requires users to possess principal identities from domain *bar.net* (like the R_1 requirement defined in section 3);
   - Identity provider information - information about the identity provider and corresponding authentication services that are supposed to have authenticated the user's principal identity. This can include the identity provider's name, location (e.g. an URL), and authentication service bindings and locations;
   - Authentication context information - characteristics about the processes, procedures and mechanisms by which a principal identity must have been authenticated by the identity provider, in order to access a workflow activity. These characteristics may be categorized as follows:
     - Identification mechanism - characteristics of the processes and mechanisms used by the identity provider to initially create an association between a user and the identity by which he will be known. Examples include face-to-face, online and shared secret identification;
     - Credential protection - characteristics that describe how the "secret" (the knowledge or possession of which allows the user to authenticate to the authentication authority) is kept secure;
     - Authentication method - characteristics that define the mechanisms by which the user authenticates to the authentication authority (for example, a password versus a smartcard, or even multifactor authentication, requiring more than one mechanism). The R_3 requirement defined in section 3 is an example of an authentication method requirement.
2. **We identify access control context information** - this includes information that can be derived from the access request context, and that is already used to constrain authorizations to workflow activities, namely:
   - Generic access control information:
     - User attributes, like age or nationality;
     - Object attributes (content-based access control);
     - External conditions like access time and location (context-based access control);
     - Access history (history-based access control); and
     - Component relationships.
       For example, an authentication requirement

may be defined stating that, if a workflow activity is accessed from a workstation in a public location, then users are required to present an extra authenticator (for example, a Social Security Number identifier);

- Workflow access control information - this refers to information used to define an authentication requirement that may be based on:
  - Workflow history, i.e., information about activity instances already executed;
  - Values of their input data - the R_4 requirement defined in section 3 defines a condition for the authentication mechanism to use if the loan request to approve surpasses a determined value.

3. **We identify workflow specific authentication information** - this refers to authentication information used in the execution of workflow activities. The R_2 and R_5 requirements defined in section 3 are examples of authentication requirements that refer to authentication contexts used on previous activities. The R_5 requirement presents also a kind of a junction of duty constraint, as it forces the same user principal identity to be used in accessing both a_1 and a_{11} workflow activities.

4.2 Defining RBAC Authorizations with Authentication Constraints

Defining an authentication requirement for a workflow activity may be accomplished by extending corresponding authorization definitions in the workflow definition, adding it authentication-specific constraints.

WfMSs have limited advance planning, i.e., workflows can be loaded and instantiated at any moment, and users cannot foresee with which activities are they supposed to interact. Thus, in an RBAC model, previous authorizations are given by assigning roles to users, based on principal identities authenticated in a reference domain (like the *bank.org* domain in our workflow example described in section 3). For example, even if a user u_1 does not possess a principal identity that satisfies a workflow activity authentication requirement, the roles associated with his reference domain principal identity may allow him to see and begin an interaction with his role-based assigned activities. This can be accomplished by manipulating visibility properties of workflow activities (Muehlen, 2004). After picking up an activity for interaction, the access control manager may then evaluate if that user has principals that can satisfy a pre-defined activity authentication requirement.

This is a kind of lazy authorization decision-making evaluation that complies with most WfMSs' poor advanced planning. Therefore, we present in Figure 3 our proposed extended workflow RBAC model for supporting authentication requirements as authorization decision constraints that restrict the permission assignment (PA) relationship.

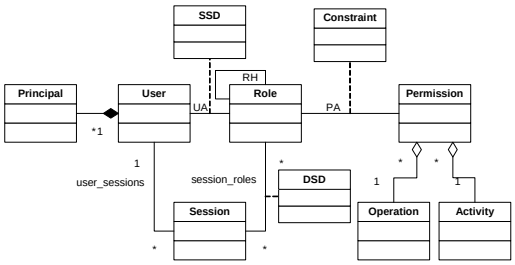

Figure 3: An extended RBAC workflow model for supporting authentication constraints.

Differences between this model and the generic one referred in Figure 1 include:

- `Principal` - represents information about a user principal identity. Users may have several principals in order to interact with workflow activities, and authentication constraints may refer to specific principals or principal authentication contexts;
- `Constraint` - represents constraints that restricts a role-permission (PA) relationship and can also include authentication information.

An authentication constraint may be modeled into a workflow RBAC authorization rule as a tuple *(r, a, execute, cn)* that states that a user principal u playing role *r* can execute activity *a* if the constraint *cn* is met.

We present in Appendix A an XML[1] definition for the workflow example of Figure 2, including formal definitions of the authentication requirements identified. Activity definition details were omitted. We include access control rules to specify role-permission assignments (XML `<actACRule>` elements), and authorization constraints for those assignments (XML `<actACConstraint>` elements). We define authentication requirements as authorization constraints under the child XML `<authConstraint>` element. For size reasons, only activities with authentication constraints are defined.

4.3 Enforcing Authorization Decisions that Include Authentication Constraints

Enforcing authorization decisions that include authentication constraints present two correlated challenges:

[1]We use the XML language to enhance reading and comprehension. The specification or adoption of a standard schema to define authentication requirements is recommendable, yet out of the scope of this paper.

1. What to do when authentication requirements are not met by lack of information - we consider that authorization decision results may have three different values: $REJECT$, $ACCEPT$ and $ADDITIONAL$. The $ADDITIONAL$ value means that the access control decision function could not resolve the authentication constraints, and needs more information to do so. How to handle this $ADDITIONAL$ result is somehow a particular security policy decision that can be solved by the WfMS opting from:
   - Rejecting or accepting the access request; or
   - Getting the additional information required to support the authorization decision.
2. How to request for additional authentication information to support authorization decisions - when a user requests access to a workflow activity with authentication requirements, the context information available in the request might not be sufficient to make an authorization decision. Regarding a user's authentication context, the WfMS must identify the required additional information and communicate it to the requesting user agent (e.g. a web-based application). For the specific workflow history information, the WfMS must save the authentication context with which each workflow activity is executed, in order to consult and compare that information when evaluating an authentication constraint for a subsequent workflow activity access request.

In Algorithm 1 we present an algorithm for an access decision function that processes an access request made on behalf of a user *u* with a role *r* to uphold permission *p* on a given workflow activity *a* with a list of authentication constraints *cntList*.

We consider the $ADDITIONAL$ return state as a set of attribute identifiers gathered along the evaluation of each authentication constraint, corresponding to additional information needed to evaluate the access request. For example, when a user's principal identity u, authenticated with a password authenticator through a web-based client application and authorized to play the *Branch manager* role, wants to execute activity a_4 from our workflow scenario, the WfMS receives an access request containing the user's principal authentication context information. That information is then compared with the authentication constraint defined for activity a_4 (see element `<authConstraint>` for a_4 in Appendix A). If the user's principal authentication context information does not refer the authenticator's transport protocol used in the authentication process, the access control decision function returns $ADDITIONAL$. This $ADDITIONAL$ state refers to a set of attribute identifiers containing, in this case, the `<authenticatorTransportProtocol>`

Algorithm: Access control decision function
Input: an access request from user u with role r to uphold permission p over activity a with a list of constraints $cntList$
Output: [1] $REJECT$, [2] $ACCEPT$, [3] $ADDITIONAL$ - Set of attribute identifiers

```
begin
  if userRole(u,r) and
       rolePermission(r,a,p) then
    foreach cnt in cntList do
      result ← evalCnt(u,cnt)
      switch result do
        case 0 return REJECT
        case 1 continue
        otherwise
          put result in
          ADDITIONAL
        end
      end
    end
    if ADDITIONAL is empty then
      return ACCEPT
    else return ADDITIONAL
  else return REJECT
end
```

`userRole(u,r)` returns TRUE if a user u is authorized to play role r, else returns FALSE

`rolePermission(r,a,p)` returns TRUE if role r has permission p over activity a, else returns FALSE

`evalCnt(u,cnt)` returns 0 if the constraint was not satisfied, 1 if the constraint was satisfied, and an attribute identifier if an additional user attribute is needed to evaluate the constraint.

Algorithm 1: Algorithm for processing a constrained access decision on a workflow activity.

identifier, as the constraint demands for the use of the SSL protocol and there is no way for the access control decision function to know which one was used to obtain the user's principal identity. Otherwise, the access control decision function returns $REJECT$ on the first constraint not met by the user's authentication context, or $ACCEPT$ if all constraints are met.

5 CONCLUSIONS AND FUTURE WORK

As organizations use workflow technology increasingly, many problems cannot be ignored anymore neither solved with ad-hoc solutions. In this paper we

identify and systematize workflow authentication requirements. We propose to meet these requirements by supporting authorizations with authentication-based constraints. With this purpose we define an extension to the RBAC model and propose an algorithm that enforces these authorizations. With our approach, users do not need to satisfy, in the initial authentication process, all workflow authentication requirements, since they are only evaluated when they request access to activities. Moreover, when a user does not possess all the required authentication information, the access control decision function notifies the WfMS, that can choose how to proceed.

We are implementing workflow authentication requirements in a WfMS called WorkSCo (Workflow with Separation of Concerns) (INESC Lisboa Software Engineering Group, 2004). WorkSCo already uses an RBAC model that needs to be extended in order to provide evaluation and enforcement of authentication constraints. As to communications between WorkSCo and exterior identity providers, we intend to adopt a standard called Security Assertion Markup Language (SAML)(OASIS, 2005). SAML is an XML-based framework for communicating user authentication, entitlement and attribute information. SAML-enabled systems may exchange authentication information on users, independently of their heterogeneous software and hardware environments. Messages exchanged must obey to SAML defined XML schemas and protocols that will enable the WorkSCo WfMS to query different identity providers and consume produced SAML authentication assertions, in order to produce and enforce authorization decisions that include authentication constraints.

REFERENCES

Bertino, E., Ferrari, E., and Atluri, V. (1999). The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. *ACM Trans. Inf. Syst. Secur.*, 2(1):65–104.

Beznosov, K. (1998). Requirements for access control: Us healthcare domain. In *RBAC '98: Proceedings of the third ACM workshop on Role-based access control*, page 43, New York, NY, USA. ACM Press.

Casati, F., Castano, S., and Fugini, M. (2001). Managing workflow authorization constraints through active database technology. *Information Systems Frontiers*, 3(3):319–338.

Casati, F., Castano, S., and Fugini, M. G. (1998). Enforcing workflow authorization constraints using triggers. *Journal of Computer Security*, 6(4):257–285.

Ferraiolo, D. F., Sandhu, R. S., Gavrila, S. I., Kuhn, D. R., and Chandramouli, R. (2001). Proposed NIST Standard for Role-based Access Control. *Information and System Security*, 4(3):224–274.

Hung, P. C. K. and Karlapalem, K. (2003). A secure workflow model. In *CRPITS '03: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003*, pages 33–41, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

INESC Lisboa Software Engineering Group (2004). WorkSCo: Workflow with Separation of Concerns. http://sourceforge.net/projects/worksco/.

Kandala, S. and Sandhu, R. (2002). Secure Role-Based Workflow Models. In *DAS'01: Proceedings of the fifteenth Annual Working Conference on Database and Application Security*, pages 45–58, Niagara, Ontario, Canada. Kluwer Academic Publishers.

Kent, S. T. and Millett, L. I., editors (2003). *Who goes There? Authentication Through the Lens of Privacy*. National Academies Press, Washington, DC, USA.

Moodahi, I., Gudes, E., Lavee, O., and Meisels, A. (2004). A Secure Workflow Model Based on Distributed Constrained Role and Task Assignment for the Internet. In *ICICS'04: Proceedings of the sixth International Conference on Information and Communications Security*, pages 171–186, Malaga, Spain. Springer-Verlag.

Muehlen, M. Z. (2004). Organizational Management in Workflow Applications – Issues and Perspectives. *Inf. Tech. and Management*, 5(3-4):271–291.

OASIS (2005). SAML V2.0 Executive Overview. Technical report, Organization for Advancement of Structured Information Standards.

Samarati, P. and di Vimercati, S. D. C. (2000). Access Control: Policies, Models, and Mechanisms. In *FOSAD '00: Revised versions of lectures given during the IFIP WG 1.7 International School on Foundations of Security Analysis and Design*, pages 137–196, Bertinoro, Italy. Springer-Verlag.

Tzelepi, S. and Pangalos, G. (2001). A flexible access control model for multimedia medical image security. In *PCM '01: Proceedings of the Second IEEE Pacific Rim Conference on Multimedia*, pages 1030–1035, Beijing, China. Springer-Verlag.

Wang, L., Wei, L., Liao, X., and Wang, H. (2004). AT-RBAC: An Authentication Trustworthiness-Based RBAC Model. In *GCC Workshops*, pages 343–350, Wuhan, China. Springer-Verlag.

Workflow Management Coalition (1999). Terminology & Glossary. Technical report, Workflow Management Coalition.

APPENDIX A

An XML Workflow Definition Example with Authentication Constraints

```
<wfDefinition workflowID="w1" name="Loan Approval">...
    <wfACConstraint>
      <authConstraint>
        <user>
          <principalID>
            <domain>bank.org</domain>
          </principalID>
        </user>
      </authConstraint>
    </wfACConstraint>...
    <actDefinition activityID="a3" type="invoked app" name="Get Client Data">...
      <actACRule>
        <actRole>Branch manager</actRole>
        <actOperation>execute</actOperation>
      </actACRule>
      <actACConstraint>
        <authConstraint>
          <provider>
            <name>iDP</name>
            <url>http://www.idp.org</url>
            <authenticationService>
              <location>http://www.idp.org/authnService</location>
              <binding>HTTP-binding</binding>
            </authenticationService>
          </provider>
          <authMethod comparison="stronger">
            <principalAuthnMech>
              <reference type="activityID">a1</reference>
            </principalAuthnMech>
          <authMethod>
        </authConstraint>
      </actACConstraint>
    </actDefinition>
    <actDefinition activityID="a4" type="invoked app" name="Get Indexed Rates">...
      <actACRule>
        <actRole>Branch manager</actRole>
        <actOperation>execute</actOperation>
      </actACRule>
      <actACConstraint>
        <authConstraint>
          <authMethod>
            <authenticatorType>password</authenticatorType>
            <authenticatorTransportProtocol>SSL</authenticatorTransportProtocol>
          </authMethod>
        </authConstraint>
      </actACConstraint>
    </actDefinition>...
    <actDefinition activityID="a9" type="work item" name="Evaluate Loan">...
      <actACRule>
        <actRole>General manager</actRole>
        <actOperation>execute</actOperation>
      </actACRule>
      <actACContraint>
        <authnConstraint comparison="greater">
          <reference type="actInputValue">100000</reference>
          <techProtection>
            <privateKeyProtection>
              <keyActivation>ActivationPin</keyActivation>
              <keyStorage medium="smartcard"/>
            </privateKeyProtection>
          </techProtection>
          <authMethod>
            <principalAuthnMech>smartcard</principalAuthnMech>
            <authenticator>asymmetricKeyDecryption</authenticator>
          <authMethod>
        </authnConstraint>
      </actACConstraint>
    </actDefinition>...
    <actDefinition activityID="a11" type="work item" name="Inform Client">...
      <actACRule>
        <actRole>Branch clerk</actRole>
        <actOperation>execute</actOperation>
      </actACRule>
      <actACContraint>
        <authConstraint>
          <user>
            <principalID>
              <reference type="activityID">a1</reference>
            </principalID>
          </user>
        <authConstraint>
      </actACConstraint>
    </actDefinition>
<wfDefinition>
```