Neighbourhood-Trust Dependency Access Control for WFMS

Jianming Yong
Department of Information Systems, Faculty of Business
University of Southern Queensland, Australia
yongj@usq.edu.au

Abstract

WFMSs are widely used by modern business processes. But so far it is still a challenge to have a simple and effective access control mechanism for WFMSs. This paper contributes an effective and simply access control mechanism, called Neighbourhood-Trust Dependency Access Control (NETDEPAL), for WFMSs. This new notion combines the workflow mechanism and RBAC into Netdepal for WFMSs. The secure access for WFMSs can be efficiently implemented by Netdepal from system dependency and task dependency via their neighbourhood relationships.

Keyword: WFMS, RBAC, Neighbourhood-Trust Dependency Access Control

1. Introduction

Workflow management systems (WFMSs) have been used in business process re-engineering for many years. More and more organisations are using WFMSs to facilitate their daily business processes. Furthermore WFMSs are used to coordinate the intra-organisation processes, for an example, a workflow system can be used to facilitate the operations of an enterprise supply chain which connects customers, venders, material suppliers, and financial agents, etc. Because a workflow system can run across different organisations, it is essential to have a reliable access control mechanism to ensure the security of all transactions among these intra-organisation processes. It is obvious that an intra-organisation workflow includes a series of defined tasks which are run by different users in the different domains. Different users are assigned suitable authorities to execute the designated tasks of a workflow. In order to clearly illustrate the implementation of a workflow, we use an online air-ticket booking system as an example. This online air-ticket booking workflow can be decomposed into the following four tasks:

- Task 1 (T1) customer booking: a customer wants to book air tickets and uses his/her Internet browser to input his/her data and submits to the airline's booking system;
- Task 2 (T2) airline booking processing: Airline booking system gets data from the customer and checking the seat availability and contacting the bank for verification of customer's banking information;
- Task 3 (T3) banking system processing: Bank system gets data and processing and confirming airline booking system;
- Task 4 (T4) confirming: The airline confirms the customer the final processing result.

This example well demonstrates the different users who have different purposes to do different tasks in a streamline workflow. Each task can only be done by the different users or systems associated with relevant authorities. Like T1, only a customer can do it on their demands. T2 can only be executed by airline booking system. T3 can only be implemented by the banking system. T4 needs to be done by airline booking system. Obviously the system which executes T1 needs to access the system which executes T2, in reverse T2 system needs accessing T1 system. The system of T2 needs accessing the system of T3, in reverse T3 system needs accessing T2 system. T3 system needs accessing T4 system, and reversely T4 system needs accessing T3 systems. In order to ensure the security and seamless connection of a workflow, an effective access control policy needs to be developed for WFMSs. This paper intends to introduce an effective access control mechanism for WFMSs, namely *ne*ighbourhood-*t*rust *dep*endency *a*ccess contro*l* (NETDEPAL).

This paper is organised as the follows. Section 2 discusses some basic notations of WFMS and recent developments. Section 3 introduces a current dominant access control technology, namely role-base access control (RBAC). Section 4 identifies constraints of RBAC for WFMSs. Section 5 discusses NETDEPAL in details. Section 6 draws the conclusions.

2. WFMS

Workflow is a key technology for automating business processes that involve access to several applications[1]. This makes workflow technology become one of the most important candidates for integrating [3, 4, 6], automating [3, 4, 7] and monitoring business processes [2, 5, 6, 7]. Internet-

1-4244-0165-8/06/$20.00 ©2006 IEEE.

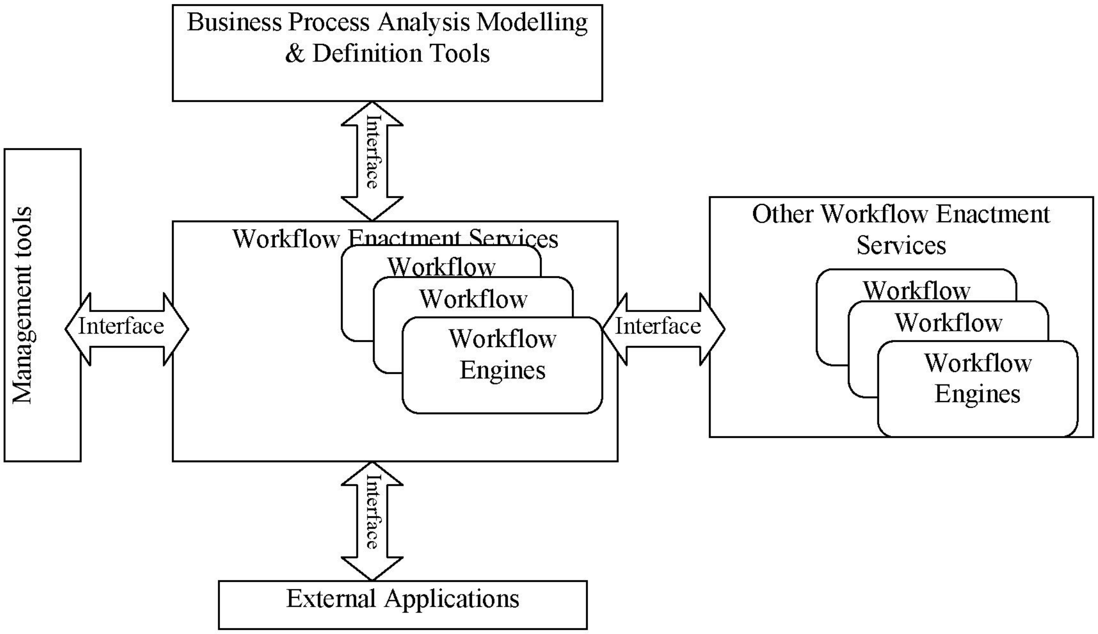

Figure 1. General architecture of workflow

based e-commerce utilises workflow technology to effectively conduct inter & intra-organisational business processes. The Workflow Management Coalition (WfMC) [8] has been formed by vendors, users, consultants, and researchers to develop the primary standards in workflow management arena. The first most important standard is the workflow reference model [9], which appeared in January 1995. In the document of the workflow reference model, workflow is defined as the computerised facilitation or automation of a business process, in whole or part. It is concerned with the automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules to achieve, or contribute to, an overall business goal. Whilst workflow may be manually organised, in practice most workflow is normally organised within the context of an IT system to provide computerised support for the procedural automation. Workflow is often associated with business process re-engineering, which is concerned with the assessment, analysis, modelling, definition and subsequent operational implementation of the core business processes of an organisation or other business entities. While a workflow management system is defined as a system that completely defines, manages and executes "workflows" through the execution of software whose order of execution is driven by a computer representation of the workflow logic. All workflow management systems exhibit certain common characteristics, which provide a basis for developing integration and interoperability capability between different products. At the highest level, all workflow management systems are characterised as providing support in three functional areas:

- the build-time functions, concerned with defining, and possibly modelling, the workflow process and its constituent activities;
- the run-time control functions, concerned with managing the workflow processes in an operational environment and sequencing the various activities to be handled as part of each process;
- the run-time interactions with human users and IT application tools for processing the various activity steps.

Figure 1 illustrates the general workflow architecture. The major components and interfaces are clearly identified through the architecture.

After the workflow reference model, several standards have been proposed at the interface level for business process integration, such as Wf-XML [10], MIME [11].

3. Role-based Access Control (RBAC)

In February 2004, ANSI INCITS 359-2004 [12] was approved. It is a milestone for RBAC research and development. It has a significant impact on the access control. So far RBAC has been accepted as a primary technology for system access control. It is a result of co-efforts by both academia and industry.

3.1. In academia

From 1995 to the present, there are 10 ACM symposiums held on Access Control Models and Technologies. Most scholarly papers published at the symposiums are more or less related to RBAC. If any search is done on ACM or IEEE digital libraries for RBAC, there will bring hundreds of papers which are relevant to RBAC. The fundamental concepts of RBAC were addressed by Sandhu, et al [13]. In [13], RBAC was categorized as four sub-components, $RBAC_0$, $RBAC_1$, $RBAC_2$, and $RBAC_3$.

$RBAC_0$ is base model of RBAC. In ANSI INCIT 359-2004, $RBAC_0$ is called Core RBAC. Core RBAC has four fundamental entities: Users (U), Permissions (P), Sessions (S), and Roles (R). Users are human beings or their agents who have ability to use information systems. Roles are duties and responsibilities in an organisation or a system. Permissions are concerning access to roles with suitable users. Sessions are responsible for establishing dynamical relationships between users and roles. The set of relationships of U, P, S, and R is defined at Core RBAC as $U \times R$, $P \times R$, $S \rightarrow U$, and $S \rightarrow Set(R)$. $U \times R$ means the relationship mapping between users and roles. $P \times R$ means the relationship mapping between permissions and roles. $S \rightarrow U$ means that a specific user gets a required session or sessions. $S \rightarrow Set(R)$ means that a session is associated with a set of roles.

$RBAC_1$ is defined as role hierarchies. In ANSI INCIT 359-2004, $RBAC_1$ is called Hierarchical RBAC. Hierarchical RBAC is actually defined the relationships among different roles, especially role inheritance. The main purpose of hierarchical RBAC is to simplify the system administration and management.

$RBAC_2$ is defined as Constraints Model. In ANSI INCIT 359-2004, $RBAC_2$ is called Constrained RBAC. Constrained RBAC actually puts constrains on users what roles and permissions can be assignment to. It can prevent some users from abusing the system. Furthermore static constraint and dynamic constraint are introduced by Ferraiolo, et al [14].

$RBAC_3$ is actually a combination of $RBAC_1$ and $RBAC_2$. It has not been listed as a separate component in ANSI INCIT 359-2004.

3.2. In Industry

Many popular operating systems, like, Windows, Linux, Unix, are widely using roles to manage system resources. Major DBMS products, like Oracle, Sybase, etc, all support RBAC. Major enterprise integration solutions, like SAP, PeopleSoft, widely use the concept and mechanism of roles which have the similar functions as being defined in RBAC.

4. Constraints on RBAC for WFMS

Though RBAC has been widely used by current information systems, the role is only meaningful to a defined system and is meaningless to any other systems. Continuously we still use the online air ticket booking system as an illustration. There are three systems which are involved in this workflow (w), customer system (S1), airline system (S2) and banking system (S3). Prior to the workflow, S1, S2 and S3 are definitely separate systems. In order to simplify the analysis, we assume that all systems have used RBAC as their access control. S1 has a set of roles, S1.r, a set of users, S1.u, a set of permissions, S1.p, a set of sessions, S1.s. S2 has a set of roles, S2.r, a set of users, S2.u, a set of permissions, S2.p, a set of session, S2.s. S3 has a set of roles, S3.r, a set of users, S3.u, a set of permissions, S3.p, and a set of sessions, S3.s.

1 $S1 = S1.r \cup S1.u \cup S1.p \cup S1.s$

2 $S2 = S2.r \cup S2.u \cup S2.p \cup S2.s$

3 $S3 = S3.r \cup S3.u \cup S3.p \cup S3.s$

4 $S1.r \cap S2.r \cap S3.r = \phi$

5 $S1.u \cap S2.u \cap S3.u = \phi$

6 $S1.p \cap S2.p \cap S3.p = \phi$

7 $S1.s \cap S2.s \cap S3.s = \phi$

The previous equations (1-7) well represent the independency of S1, S2 and S3 prior to forming the online air ticket booking workflow. Because the online air ticket booking workflow needs a full cooperation of S1, S2 and S3. We will have the following equations.

8 $w = (\text{subset of } S1) \cup (\text{subset of } S2) \cup (\text{subset of } S3)$

9 $S1.r \cap S2.r \cap S3.r \neq \phi$

10 $S1.u \cap S2.u \cap S3.u \neq \phi$

11 $S1.p \cap S2.p \cap S3.p \neq \phi$

12 $S1.s \cap S2.s \cap S3.s \neq \phi$

The formulas (8-12) present the cooperation relationships among S1, S2 and S3. There have to have some common sets to seamlessly connect S1, S2 and S3 to form an effective booking workflow. The online air ticket booking system is shown in Figure 2. With 8 steps, the booking workflow finishes its procedures. S1 needs an access to S2 at step 1-2. S2 needs an access to S3 as step 3-4. S3 needs an access to S2 as step 5-6. S2 needs an access to S1 as step 7-8.

Definition 1. Access order is defined as a 2-tupe <Si, Sj>, ($Si \in w$ and $Sj \in w$), where Si executes its task first then needs an access to Sj for the follow-up task. As Figure 2, S1 executes T1 then needs an access to S2 for T2. This can be expressed as <S1, S2> .

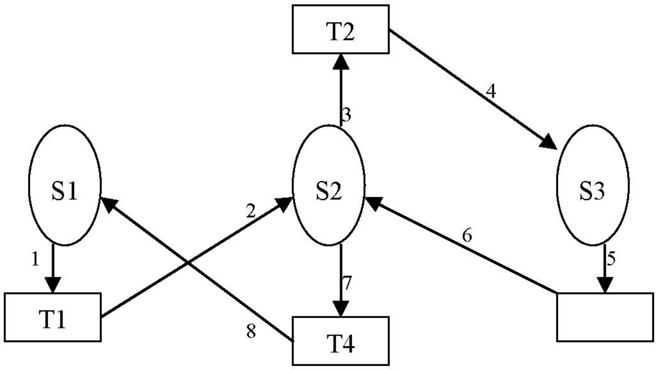

Figure 2. workflow of online air ticket booking

The access orders of this online air ticket booking workflow are <S1, S2>, <S2, S3>, <S3, S2>, <S2, S1>. Prior to this workflow, S1, S2 and S3 have their own defined RBAC sets respectively. When S1, S2 and S3 form a workflow, these RBAC sets need keeping its original independency so that S1, S2 and S3 can be used separately. In the other hand, they need a common RBAC set so that two neighbour systems can work together to finish the defined tasks of a workflow. This common RBAC will ensure the secure access to streamline systems. Like in Figure 2, S1 has its RBAC sets and S2 also has its own RBAC sets. In order to execute T1 by S1 and T2 by S2 securely and seamlessly, an effective mechanism needs to be work out between S1 and S2. So far to the best of our knowledge, there is no much literature about this research. We propose a solution, called Neighbourhood-Trust Dependency Access Control (NETDEPAL).

5. Neighbourhood-Trust Dependency Access Control (NETDEPAL)

A workflow has a series of well defined tasks which are executed by different/same systems. Like in Figure 2, T1 is executed by S1, T2 and T4 are executed by S2, while T3 is executed by S3. In order to clearly demonstrate NETDEPAL, we need the following definitions.

Definition 2 Workflow task sequence and workflow system sequence: a workflow task sequence is defined as a 2-tupe, *<Ti, n>, Ti* is a defined task of a workflow (*w*), *Ti* ∈ *w, n* is the order number of executed tasks. For an example, in the Figure 2, T1 is executed first. We denote it as *<T1, 1>*. Similarly a workflow system sequence is defined as a 2-tupe, *<Si, m>*, *Si* is a system which executes tasks for a workflow. The *m* is the order number of *Si* in the streamline systems for the workflow. *<S1, 1 >* denotes that the system, *S1,* is firstly used to execute a defined task for the workflow. *<S2, 2>* denotes that S2 is secondly used to execute a defined task.

Definition 3 Neighbourhood task: a neighbourhood task is defined as a 2-tupe, [Ti, Tj], a task which is executed immediately before or after it. Like there is <Ti, n>, its neighbourhood tasks are Tj, Tk, while there exists a workflow task sequence <Tj, n-1> or <Tk, n+1>. For an example, in Figure 2 the streamline of tasks is <T1, 1>, <T2, 2>, <T3, 3>, and <T4, 4>. T1 and T3 are the neighbourhood tasks of T2, [T1, T2] and [T2, T3].

Definition 4 Neighbourhood system: a neighbourhood system is defined as a 2-tupe, {Sy, Sz}, a system which is used immediately before or after it. Like there is <St, m>, its neighbourhood systems are Sy and Sz, which there exists a workflow system sequence <Sy, m-1> or <Sz, m+1>. For an example, in Figure 2 the streamline of systems is <S1, 1>, <S2, 2>, <S3, 3>, <S2, 4>, and <S1, 5>. S1 and S3 are S2's neighbourhood systems, {S1, S2} and {S2, S3}.

Definition 5 Task dependency: Task dependency is defined as a relationship, Ti→Tj, a task, Tj is to be executed depending on the successful execution of its prior task, Tj. Like in Figure 2, Task 2 can only be executed after Task 1 successfully finishes, T1→T2. It is obvious that task dependency comes from neighbourhood task, [Ti, Tj] ↔ Ti→Tj

Definition 6 System dependency: System dependency is defined as a relationship, Si→Sj, a system, Sj, only needs to be used depending on the usage of its prior system, Si. Like in Figure 2, the online air ticket booking workflow utilise S1 first then S2, S1→S2. Similarly system dependency comes from neighbourhood systems, {Sy, Sz}↔Sy→Sz.

Again a workflow consists of a series of well defined tasks [15]. A general workflow can be modelled as:

$$w\text{=T(T1, T2, T3, …, T}_{n\text{-}1}\text{, Tn)}$$

All systems involved in the execution of the workflow are modelled as:

$$\text{S=(S1, S2, S3, …, S}_{m\text{-}1}\text{, Sm)}$$

According to the system requirements, all the tasks are assigned to relevant systems (as a mapping function, *f*). The system sequence (*Q*) for the workflow is modelled as:

Q=f(T)=*S'* *S'* is the streamline of involved systems for the workflow.

From the definition 5 & 6, we know that system dependency is decided by task dependency because each task can only be executed by a specific system. We can model a workflow as the follows.

w→T →Neighbourhood task→task dependency→Users→roles→*S'*→system dependency →system neighbourhood→access control policy

Neighbourhood-trust dependency access control for a workflow can be implemented effectively. For an example, if there exists Si→Sj, Sj allows Si's roles which are involved in the workflow process to access its relevant resources for the workflow tasks. Through this mechanism, the access control of a workflow becomes simple, effective and efficient.

6. Conclusions

More and more WFMSs are used in current business processes. It is very important to have a good mechanism to implement an effective access control policy for a WFMS. So far, RBAC is a dominant access technology used in many systems. This paper discusses RBAC and its constraints for WFMSs. Furthermore, a neighbourhood-trust dependency access control is proposed and analysed for WFMSs. The major contribution of this paper is to identify NETDEPAL as an effective access control for WFMSs. The minor contribution is to extend RBAC and its mechanism into NETDEPAL for WFMSs.

Acknowledgement

This paper is finished during my academic stay at Purdue University. Prof. Elisa Bertino gives a valuable support and advice. The paper is also partially supported by USQ's early career research grant.

References

[1] B. B. Medjahed , A. Bouguettaya , A. H. H. Ngu , A. K. Elmagarmid, "Business-to-business interactions: issues and enabling technologies," *The VLDB Journal — The International Journal on Very Large Data Bases*, vol. 12, pp. 59-85, 2003.

[2] P. Hartmann, R. Studt, and T. Wewers, "A Framework for Classifying Interorganizational Workflow-Controlled Business Processes Focusing on Quality Management," *the 34th Hawaii International Conference on System Sciences*, Maui, Hawaii, USA, 2001.

[3] M. Kwak, D. Han, and J. Shim, "A Framework Supporting Dynamic Workflow Interoperation and Enterprise Application Integration," *the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)*, 2002.

[4] J. A. Espinosa and A. S. Pulido, "IB(integrated Business): a Workflow Based Integration Approach," *the 35th annual Hawaii international conference on system sciences (HICSS-35'02)*, Big Island, Hawaii, USA, 2002.

[5] M. Rohloff, "Reference Model and Object Oriented Approach for Business Process Design and Workflow Management," *the 1996 Information Systmes Conference of New Zealand*, Palmerston North, NEW ZEALAND, 1996.

[6] M. Gillmann, G. Weikum, and W. Wonner, "Workflow Management with Service Quality Guarantees," *the International Conference on Management of Data and Symposium on Principles of Database Systems*, Madison, Wisconsin, USA, 2002.

[7] P. Xu and B. Ramesh, "Supporting Workflow Management Systems with Traceability," *the 35th Hawaii International Conference on System Science*, Big Island, Hawaii, USA, 2002.

[8] WfMC, "Workflow Mangement Coalition," http://www.wfmc.org, 2003.

[9] D. Hollingsworth, "The Workflow Reference Model," vol. http://www.aiim.org/wfmc/standards/docs/tc003v11.pdf: WfMC-TC-1003, Version 1.1, 1995.

[10] WfMC, "Workflow Management Coalition Workflow Standard – Interoperability Wf-XML Binding," http://www.wfmc.org/standards/docs/Wf-XML-11.pdf, 2003.

[11] WfMC, "Workflow Standard – Interoperability Internet e-mail MIME Binding," http://www.wfmc.org/standards/docs/Mime11f.pdf, 2003.

[12] ANSI, American National Standard for Information Technology – Role Based Access Control, *ANSI INCITS 359-2004*, February 2004.

[13] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, Role-Based Access Control Models, *IEEE Computer*, 29 (2): 38-47, February 1996.

[14] D. F. Ferraiolo, R. S. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, Proposed NIST standard for role-based access control, *ACM Transactions on Information and Systems Security*, 4(3):224-274, August 2001.

[15] E. Bertino, E. Ferrari, and V. Atluri, A Flexible Model Supporting the Specification and Enforcement of Role-based Authorizations in Workflow Management Systems, *RBAC' 97*, Fairfax, VA, USA, pp1-12.