Document access control in organisational workflows

Timon C. Du*

Decision Sciences and Managerial Economics,
The Chinese University of Hong Kong,
Shatin, N.T., Hong Kong
E-mail: timon@cuhk.edu.hk
*Corresponding author

Eldon Y. Li

Department of Management Information Systems,
National Chengchi University,
Taipei 11605, Taiwan
E-mail: eli@calpoly.edu E-mail: eli@nccu.edu.tw

Jacqueline W. Wong

Decision Sciences and Managerial Economics,
The Chinese University of Hong Kong,
Shatin, N.T., Hong Kong
E-mail: jacquelinewong@cuhk.edu.hk

Abstract: A collaborative workflow is a business process with a set of linked tasks. It is important to share knowledge in document format of the workflow to achieve a business objective or policy goal. When an electronic document is shared in a collaborative workflow, appropriate access controls are needed. Access control of documents involves the correlated setting of security at the document and data levels, corresponding to the sequence of workflow activities and organisational role hierarchy. This study proposes an access control mechanism for sharing electronic documents in a document-centric Workflow Management System (WfMS). A mandatory access mechanism is used to manage access control. The mechanism is demonstrated by an example of generating a quotation document using Oracle Workflow and Oracle PL/SQL.

Keywords: document management; security control; Role-Based Access Control; RBAC; workflows; mandatory access mechanism; Oracle PL/SQL; information and computer security.

Reference to this paper should be made as follows: Du, T.C., Li, E.Y. and Wong, J.W. (2007) 'Document access control in organisational workflows', *Int. J. Information and Computer Security*, Vol. 1, No. 4, pp.437–454.

Biographical notes: Timon C. Du is Professor of Decision Sciences and Managerial Economics at the Faculty of Business Administration, The Chinese University of Hong Kong, China. He also serves as the Director of Master of Sciences in e-Business Management. He received MS and PhD Degrees in Industrial Engineering from Arizona State University. Currently, his research interests are business intelligence, RFID privacy and security, e-logistics, culture and e-commerce and semantic web. He has published papers in many

leading international journals. He was the Executive Editor for the *International Journal of Internet and Enterprise Management*. Currently, he is the Executive Editor for the *International Journal of Electronic Business* and the President of the International Consortium for Electronic Business (ICEB).

Eldon Y. Li is University Chair Professor in the College of Commerce at National Chengchi University, Taiwan. He was the Professor and Dean of College of Informatics at Yuan Ze University, Taiwan. He is on leave from the Orfalea College of Business, California Polytechnic State University, San Luis Obispo, California, USA. He was the Founding Director of the Graduate Institute of Information Management at the National Chung Cheng University, the President of the Western Decision Sciences Institute (WDSI), and the Founding Executive Director of the International Consortium for Electronic Business (ICEB). He holds MS and PhD Degrees from Texas Tech University. He has published over 100 papers in the areas of human factors in information technology (IT), strategic IT planning, software engineering, quality assurance, information management, and business management.

Jacqueline W. Wong is Senior Instructor of Decision Sciences and Managerial Economics at the Faculty of Business Administration, The Chinese University of Hong Kong, China. She received MS and PhD Degrees in Computer Science from The Chinese University of Hong Kong. Currently, her research interests are information retrieval, information education and document management. She has published papers in many journals such as *Decision Support Systems, ACM SIGIR Forum, ACM SIGCSE Bulletin, International Journal of Electronic Business* and others.

1 Introduction

Document management is one of the fastest growing areas in knowledge management, where end-users are involved in saving, searching, scanning, routing and revising documents. According to O'Meara, a dynamic and intelligent document system is helpful to resolve the challenging business environment and to support high-quality decision making (O'Meara, 2000). There are two categories of documents: those that contain highly structured information and those that contain more loosely structured information (Eloff et al., 1996). Examples of highly structured information are the information that is contained in contracts, purchase orders, invoices and airline reservations, while the information contained in letters, notes and reports is considered as loosely structured information. Both types of documents are essential to the dissemination of knowledge.

The organisational knowledge systems deliver the right knowledge to the right person at the right time and in the right format to facilitate the right action (Dieng, 2000). An Electronic Document Management System (EDMS) organises information in a manner that can facilitate document retrieval. For example, the major objective of a bibliographic information retrieval system is to obtain bibliographic details that are related to the user's request. The EDMS maintains both documents and data. The document that is retrieved may be an abstract or a full-text document such as a news article, a legal document and so forth. The International Organization for Standardization (ISO) has proposed an Open Document Architecture (ODA) to manage such kinds of document structure (see http://www.iso.org). Normally, the documents are maintained as

files, while the data are stored in the database. The former kind of information system can be regarded as a full-text retrieval system because it enables the user to retrieve the actual documents. Another way to maintain documents is to store document formats as templates and preserve both content and data as objects of templates in the database. This is especially useful in the workflow environment because most documents are standardised. Irrespective of the type of approach adopted, the information (both the document and the data) has to be efficiently retrieved upon request.

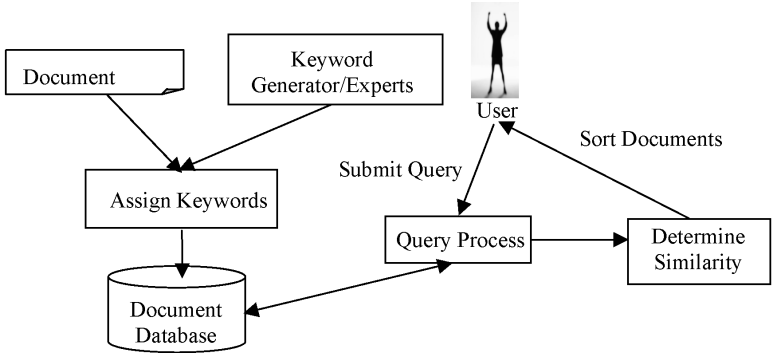
A document can be retrieved according to a keyword search. In an EDMS, search methods are incorporated into a search engine to allow information exploration, as shown in Figure 1. However, locating a document does not mean that the document is accessible. In fact, different levels of security classification protect documents differently. Sameshima and Kirstein (1996) discussed security issues in ODA documents, where documents were exchanged in different scenarios such as between two parties, within a small group or to a large number of people. Various considerations arose during these document exchanges. For example, when the documents were exchanged between two parties, the authenticity of the documents, the confidentiality and integrity of document content, the confidentiality of document flow and the proof of exchange were the major concerns. If the documents were exchanged within a group (such as in a workflow), then the addition, modification and deletion of the documents were the focal points. Moreover, different stages of document usage may also have different degrees of security needs (Eloff et al., 1996). For example, when a document is in its preparation, draft, revision, release and confirmation stages, it needs different security levels (e.g., top secret, secret, strictly confidential, confidential and unclassified).

A workflow comprises a number of business processes that involve multiple roles (or groups of participants). A business process is a set of linked tasks that achieve a business objective, policy or goal (Wu et al., 2002). To manage a workflow-driven business process, the focus points include how to

- streamline the entire process
- define and implement business policies
- monitor route information
- capture exceptions.

Therefore, a WfMS should coordinate and control system participants with the appropriate data resources to achieve defined objectives by the deadline (Kamath and Ramamritham, 1996). As the WfMS has become an important tool for business process engineering, many companies such as IBM, Oracle and HP have commercialised WfMS software. The software can integrate both inter-organisational and intra-organisational processes. There are three different types of WfMS architecture: production architecture, message-based architecture and document-centric architecture (Stohr and Leon, 2001) The production WfMS integrates with the enterprise system such as a database management system and uses a folder to hold all of the documents that are related to a particular workflow process. In contrast, a message-based WfMS is loosely integrated with the enterprise system. It normally implements the workflow process by sending the supporting documents to participants via e-mail. The document-centric WfMS adds the workflow into the document management system. The participants can accomplish the assigned tasks in a workflow with the support of attached documents. In this study, we adopt the production architecture.

Figure 1 Functions and components of a document management system



When sharing a document in a workflow, there are some security issues:

- data security that includes policy issues, security level and legal and ethical issues
- access control that prevents unauthorised access to the system
- control of access to a statistical database that provides statistical information or summaries of values based on various criteria
- data encryption that prevents sensitive data from being transmitted via some type of communications network (Trcek, 1998; Elmasri and Navathe, 2000; Turban et al., 2004).

Most existing workflow security studies focus on the first two issues and the discussions center on work assignment, security sub-processes, inter-workflow security and multilevel secure workflows. Note that the work assignment addresses the security policy concerning the different roles (or participants) involved in performing the tasks necessary for the business process. The access rights should be dynamically provoked and revoked. An example can be seen in Bertino et al. (1999) in which the formal logical authorisation model was developed. Unlike work assignment, security sub-processes focus on the security problems associated with the sequence of the business process (Herrmann and Pernul, 1999). The inter-workflow security focuses on the communication between different units of the same organisation or of different organisations. The application can be seen in Weigand and van den Heuvel (2002), who used a contract specification language to implement, coordinate and control the interaction between business flows. Multiple secure workflows allow tasks to have different levels of security so that they can belong to domains of different levels of classification without compromising security (Atluri et al., 2000).

This study proposes an access control mechanism for document sharing in a WfMS, in which the documents are available for workflow users to carry out their jobs. However, the privilege of access to a document depends on the users' security classifications, i.e., clearances. There are many interdependencies involved in the mechanism. For example, access to a document includes document objects and data objects, and these are co-dependent. Moreover, the activities of a workflow are correlated and the roles within an organisation are hierarchical. The remaining content of this paper is organised as follows. Section 2 will briefly review the access control problems in a workflow, and the control mechanism is proposed in Section 3. Section 4 will use Oracle Workflow to

demonstrate how the mechanism can be implemented, and conclusions are drawn in Section 5.

2 Sharing information in a workflow

A workflow manages the business process of an organisation. In a workflow, the processes are carried out following a specific sequence that determines which tasks need to be performed next. There are four different types of sequences: *sequential*, *parallel*, *selective* and *iterative* routings; sequential routing confines one task to be executed before another task, while the parallel routing allows two tasks to be performed with neither receiving feedback from the other. Similarly, selective routing provides a choice between or among tasks, and iterative routing allows the same task to be performed more than once. During implementation, the process needs to be enacted to perform a task. The enactment is triggered by *events* such as external events (a new order having arrived), resources (an employee making a request) or time signals (at 8 o'clock every morning) (Aslst and Hee, 2002).

To implement a workflow involves organisational structures (the role hierarchy and workflow), resources (roles, documents and data) and processes (business processes and their corresponding tasks). In the workflow, it is important that the knowledge can be shared smoothly and securely so that the organisational objective can be accomplished by a due date. Much of that knowledge (which is also called enterprise memory) such as standard business documents, personalised rules, notification and business intelligence is recorded in document format. Most current WfMSs treat the document as an entity and apply a discretionary control to ensure the security of the document. However, to share the work in a workflow efficiently, the document should have a more sophisticated access control mechanism, at least to the level of document's data. As the workflow primarily focuses on capturing the data and controlling flow requirements between the steps that comprise it, it is the duty of the database to handle the data consistency over the workflow (Kamath and Ramamritham, 1996). Moreover, in most commercial software, the workflow engine is embedded in the database server to coordinate the routing of activities for the process. Therefore, access control should mainly be handled via the database management system. This also implies that the workflow designer should consult with the database administrator to create accounts, grant privileges, revoke privileges and assign security levels.

In the literature, it is common to divide database security mechanisms into two types: discretionary security mechanisms and mandatory security mechanisms (Elmasri and Navathe, 2000). Discretionary control specifies the access privileges of users explicitly, for example, granting and revoking the privileges of designated data to users. In contrast, mandatory security mechanisms, also called multilevel security mechanisms, identify the security levels of both subjects and objects, and therefore prevent information flow to unauthorised subjects. The conventional models of discretionary control are the HRU model, Take-Grant model, Action-Entity model; models of mandatory control include the Extended Take-Grant model, Bell-LaPadula model, Biba, Dion, Sea View (see Castano et al. (1995) for further discussion). Both types of security mechanisms provide different degrees of complexity and protection for both the computer and the database. However, neither type of mechanism prevents the information from flowing from authorised to unauthorised users. This function relies on the flow control model such as the lattice

model and the RBAC model. The earliest version of the lattice model was proposed by Denning (1975); in this model, the flow relationships are organised into classes. The data can flow from one class to another class explicitly or implicitly under constraints. Unlike the lattice model, the RBAC model distinguishes the role from the user. In general, a user is a subject such as an individual or a programme who can execute a job. A role is a function involved in executing a job in an organisation with certain authority and responsibilities (Sandhu, 1993). To manage workflow security, it is a good approach to use a role-based access model to control the information flow and use either discretionary mechanisms or mandatory mecahnisms (multilevel security) to manage data access.

Access control for a workflow should involve the $RBAC_3$ model, i.e., the consolidated model in Sandhu's definition (Sandhu et al., 1996). Note that Sandhu divided the access model into four different categories: $RBAC_0$, $RBAC_1$, $RBAC_2$ and $RBAC_3$. $RBAC_0$ is the baseline model, which considers four elements, i.e., users, roles, permission and session. In general, privileged access to data objects is called a permission. Normally, the permission is assigned to a role instead of a user. Users are then assigned to a role/roles. The relationship between users and roles is many-to-many. A session is established by a user when they activate a subset of the roles to which they have been assigned. Owing to the many-to-many relationships between roles and users, the implication of each session has been included, i.e., a user can use the different privileges assigned to the various roles that they have been allocated, to implement a job. Therefore, the session itself will not be taken into consideration in this study. Instead, if multilevel security is adopted, the four elements in this study should be roles, users, objects and privileges. Note that both $RBAC_1$ and $RBAC_2$ add more elements to $RBAC_0$ by including role hierarchy, which structures the roles to reflect an organisation's assignment of authorisation and responsibility in $RBAC_1$, and introduces constraints to limit the privileges assigned to roles in $RBAC_2$. $RBAC_3$ considers both role hierarchy and constraints to $RBAC_0$. It should be noted that both the role hierarchy and the constraints have existed in the workflow of an organisation, and therefore the privileges assigned need to satisfy the constraints.

3 Defining document access control in workflow

Access control to the workflow should consider seven elements – roles, users, data objects, document objects, tasks, privileges and role hierarchies – that are defined as follows:

- A role, r, is a named collection of privileges to perform certain tasks: $r \in R$, where R is the set of roles. Role hierarchy $RH \subseteq R \times R$ is a partial order relationship established among roles R.
- A user, u, is the member of an organisation, and $u \in U$, where U is a set of users. The relationship between users and roles is many-to-many.
- A document, D, is an aggregated object composed of texts to be accessed by the public at large and data to be accessed only by individuals or roles to which this privilege has been granted.

- The data object, d, is for applications' usages. A data object is used and owned by an individual user and is allowed to be shared with other users.
- A task, t, is the process instance of a set of tasks, $t \in T$, where the task indicates the sequence of task implementation, and $t \in T$ means that the process instance is the instance of tasks T at the particular moment of accomplishing a particular job assignment.
- A privilege, p, is the object access mode assigned to a role r.

To allow different roles to have different degrees of privilege for accessing data objects in different workflows, a role-based multilevel security model, which considers different degrees of privilege, security propagation and constraints, is used. In the multilevel security model, a user cannot access data unless the clearance is higher or equal to that of the data. The access modes include *read* and *write*. The *read* privilege grants the authority of reading data objects to a role, while the *write* privilege grants the authorisation of writing data to a role after reading the data. That is, the *write* privilege is given to the role that also owns the *read* privilege.

The authorisation process considers user-role relationship, role-role relationship and user-data relationship. Both the privileges and the ownership belong to the user level. This means that the privilege propagation is at the user level instead of the role level. When assigning users to roles, the *principle of separation of duties* should be observed. When assigning privileges to roles, the *principle of least privilege* should be followed; this involves only assigning the minimum level of permission for a role to perform a task, and two mutually exclusive roles (such as accounting manager and financial controller) should be assigned to two different users.

Six tables are used to implement access control.

- *Employee table (E).* This table maintains the information such as the roles and security levels of employees as users in an organisation.
- *Role Hierarchy table (RH).* This table records the hierarchy of roles in an organisation, and the clearance of roles is assigned in the table.
- *Role Assignment table (RA).* This table records the assignment of users (U) to roles (R) in an organisation. The clearance of users is inherited from the *RH* when a user is assigned to a role. The user is the key to this table, and both the role and workflow ID are the foreign keys associated with the Access Matrix.
- *Data and Document table (H).* This table maintains the relationship between document (D) and data (d). The clearance of data objects is assigned in this table. When a document is created by a user, the clearance level of the document is set to that of the user, i.e., owner. However, some data belonging to the document may have higher clearance if they are modified by other users with higher security levels. Therefore, data within the same document may have different clearance levels.

- *Access Matrix (M).* The access matrix describes the static access mode of roles in relation to the data. The access privileges are determined when the workflow is designed, but the instances are inserted when an electronic document is created.
- *Operations Matrix (OM)*. The dynamism of assigning access privilege is maintained in this table. It should be noted that the object granularity is set to the level of data rather than the document, although the access privilege to the document must be acquired before granting the privilege. In this case, it is possible that some data in a document would not be allowed to be accessed by a user; even the privilege of the carrier document is granted.

The mechanism of generating instances in *OM* is that of rolling forward. That is, an instance is generated only when the precedent task is completed or it is the first task in a new workflow. This approach can prevent a user in an OR-split parallel route to access the document in a case where the actual route does not flow through the role. The roll forward mechanism is implemented as if the current access set c in the *OM* is composed of (u, d, t and p), the state of the system is described by the association of (c, f, M, RA, H and Wf), where f is the level function, M is the access matrix, RA is role assignment, H is the current data object hierarchy and Wf is the task definition of the workflow. The level function associates the security level, called clearance s, with roles and data objects, f: $R \cup d \rightarrow L$, L is the set of security level $L = (s)$. A security level L_1 is higher than or equal to L_2 if and only if $s_1 \geq s_2$. The condition for $s_1 \leq s_2$ is analogous. A user can access (write or read) a data object only if their security level is higher than or equal to the security level of data objects.

The five axioms upheld are:

- *Simple security property.* A system state (c, f, M, RA, H and Wf) satisfies the simple security property if and only if operations matrix *OM* (u and d) contains *read* or *write* access mode, tuple (d and D) $\in$ H, and $f(u) \geq f(d)$. It is also called no read-up secrecy. That is, a user, u, can only read a data object, d, whose security level is not higher than the level of the user.
- *Dependent security property*. A data object, d, can be accessed if tuple (d and D) $\in$ H and $f(u) \geq f(D)$. That is, the data objects cannot be accessed unless the corresponding document can be accessed.
- *Star property*. This is also called no write-down secrecy. That is, a user, u, can only write a data object, d, whose security level is equal to the level of the user. This also means that when a piece of data is written by a user, the security level will be set to the level of the user.
- *Discretionary security property.* A system state satisfies the discretionary security property if and only if (u, d, t and p) $\in$ $c \Rightarrow p \in OM[u, t, d]$. That is, every current access set of a task must be explicitly stated in the *OM*.
- *Non-accessibility of inactive objects*. A system state (c, f, M, RA, H, Wf) satisfies the non-accessibility of inactive object property if and only if $(u, d, t, p) \in c \Rightarrow p \neq read \wedge p \neq write$ where d is an inactive object. That is, the data object is not accessible if it does not appear in the object hierarchy.

In the workflow, the access privilege is temporarily granted to a user. The privilege is normally granted to a user either when a previous task is completed (roll forward) or when a higher ranked user propagates the privilege, for example, if a sales manager thinks that their subordinates need to know some information and grants permission for accessing related documents to the subordinate. Another example is if a preceding user thinks that a subsequent user should refer to their note before issuing an order. Similarly, the privilege is revoked once the task is completed. The revoking process may remove a user from the *OM* or change the access privilege from write to read after the activities have been completed. A user can refer back to the old task he/she processed before if the read privilege is still maintained in *OM*.

It should be noted that implementing access control has both static and dynamic properties. The static aspect determines the access privileges of subjects (roles) to objects (data) once a workflow instance is initiated, while the dynamic aspect refers to the actual and historical privilege granted to the users. The static privileges are defined in the Access Matrix, *M*, and the dynamics are managed by Operations Matrix, *OM*. This means that the workflow system will use *OM* to determine whether or not the access request of users can be granted. *OM* is updated on several occasions:

- new instances for the initiated users are inserted when a new document is created
- new instances for the subsequent users are inserted when the precedent task is completed
- a new instance of a user is inserted when the new privilege is granted to the user
- an instance is removed when a higher-security-level user changes the authorised data
- all privileges are changed into read when the workflow instance has been completed.

The third occasion is considered as the *privilege propagation* where whether or not the granted privilege of a user can be propagated to another user is taken into consideration. In general, a workflow user is allowed to grant access privilege to subsequent users if they own the document and data objects. However, if more than one user were to assign the privilege to a latter user but the privileges are not the same, then it is considered as *confliction* in compatibility checking. This study adopts the *pessimistic resolution* of the confliction. That is, if the confliction appears at the access mode (i.e., read or write) of a data object, the lower privilege one (i.e., read) would be adopted. If the confliction appears at the security level (i.e., clearance confliction of the grantors), the lower clearance object would be adopted (i.e., data objects with higher clearance cannot be accessed).

In summary,

- new instances are inserted into *RH* when a new workflow is defined
- new instances are inserted into *RA* when a new workflow instance is initiated
- new instances are inserted into tables *H*, *M* and *OM* when a new document is created
- new instances are inserted into *OM* when precedent activities are completed
- the access privileges of all instances in *OM* of the same workflow instance are changed into read when the workflow instance is completed.

To illustrate the procedure, the following section uses Oracle Workflow for demonstration purposes.

4 Implementation and demonstration

To illustrate how the access control mechanism of documents can be implemented in the workflow, this section uses an example of electronic quotation approval workflow. Oracle Workflow, which is one Oracle Applications product, is a system that implements workflow processes, which consist of tasks (activities in Oracle terminology) and routes (transactions in Oracle terminology). The activities are implemented in the form of notifications, PL/SQL, stored procedures, or other sub-processes, while the transactions include decision points (business branched processes), parallel flows (business processes that flow simultaneously) and loops (business processes that flow back to an activity that has been completed earlier). Oracle workflow contains six major components (Allen and Chow, 2000).

- *Workflow builder*. The Workflow Builder provides graphical interfaces for users to create, review and maintain workflow definitions. The definitions include processes, attributes, notifications, messages and functions.
- *Workflow engine*. The Workflow Engine executes the defined workflow process. It changes the state of an activity, operates function activities, performs notification activities, executes process activities and transactions, handles errors and maintains workflow history.
- *Workflow monitor*. The Workflow Monitor is used to review the status of an item in a workflow process.
- *Workflow definitions loader*. The Workflow Definition Loader is the exchange interface between Oracle Workflow and database or text files.
- *Workflow directory services*. The Workflow Directory Services tells Oracle Workflow how to find the users, interprets the roles of each user and sends notification to the users.
- *Notification system*. The Notification System is responsible for sending messages to users.

Figure 2 shows an example of the quotation generation built into Oracle Workflow. The workflow consists of eight tasks (activities), as explained in Table 1:

- Workflow designer builds a workflow for an electronic document
- An electronic form of quotation is prepared by Clerk 1
- System selects an employee from the predefined roles to act as Manager 1
- Manager 1 can either approve the quotation, reject the quotation or return to Clerk 1 if more information is needed
- Manager 1 returns the quotation to Clerk 1 for revision

- Manager 1 examines the document to determine whether any further revision is needed
- Clerk 2 processes the rejected quotation
- Clerk 3 matches the quotations with a purchase order if the quotation is approved.

Figure 2 An example of preparing quotation document in oracle workflow

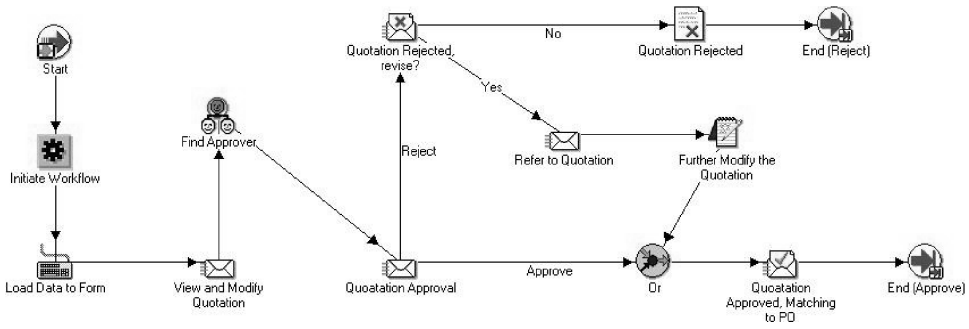


Table 1 Activities and performers of the sample workflow process

Activity	*Name*	*Performer*	*Description*
1	Initiate workflow	Workflow designer	Define the roles and workflow for electronic quotation
2	Prepare quotation document	Clerk 1	Load the quotation and prepare the quotation
3	Find Manager 1 form roles definition	Workflow system	Randomly select a manager from pre-defined roles
4	Approve, reject, or revise quotation	Manager 1	Review and decide whether or not to approve, revise, or reject the quotation
5	Revise and refer to other information	Clerk 1	Clerk 1 revised according to the comment of Manager 1. Referring to more information may be needed at this stage
6	Review the quotation	Manager 1	Manager 1 further reviews and updates the quotation if needed
7	Process rejected quotation	Clerk 2	Clerk 2 prepares the rejection letter by referring to the quotation
8	Process approved quotation and match to Purchase Order	Clerk 3	Clerk 3 reviews the quotation and matches to Purchase Order.

The workflow process includes transactions of branching and looping; it is triggered when a new quotation document is prepared by Clerk 1, and ends at either acceptance or rejection of the quotation.

Figure 3 shows that six tables – Employee table, Role Hierarchy table, Role Assignment table, Document table, Access Matrix table and Operations Management table – are used to illustrate the access control mechanism. The instances of tables are generated and updated following the discussion in the previous section. There are several scenarios that can be used to illustrate the implementation details:

- *Scenario 0.* When a workflow is defined, roles relating to the activities in the Role Hierarchy are defined and the security levels of the roles are assigned. When a clerk, e.g., Mary Anderson, would like to prepare a quotation, which is a task of Clerk 1 in workflow WF01, an instance is inserted to the Role Assignment table for her and the clearance is set to 1 for WF01 version 1 (please also refer to Figure 3). The clearance is determined by referring to the instances defined in both the Employee table and Role Hierarchy table. Figure 4 shows an illustration in which the PL SQL program code is used.
- *Scenario 1.* When a quotation document is created by Mary Anderson, the access modes of the document to the data level for the workflow are also specified in the Access Matrix table. The matrix allows users to have different access modes to different data within the same document. Also, two instances are inserted into the OM table, where the write privilege is granted to Mary Anderson and allows her to change DATA1 and DATA2. If Mary Anderson needs to refer to DATA1 in the old quotation QUO edition 0 which she prepared earlier, the system will check the instance in the OM. As Mary Anderson has read privilege for the data, the request is granted. It should be noted that all privileges are set to read after a workflow instance is completed and therefore the old data can no longer be modified. After creating the document QUO edition 1, two instances are inserted into Document table for QUO edition 1. The security level of both data objects is set at default to the clearance of Mary Anderson (document owner) (see Figure 5 for program illustration). When Mary Anderson completes her job, Ben White, who was assigned as Manager 1 based on the role definition in Employee table, reviews the quotation for workflow WF01 version 1. An instance is inserted into the Role Assignment table for Ben White, who has been assigned security level 2, and two instances are inserted into the OM, where the access mode to DATA1 is read and to DATA2 is write.
- *Scenario 2.* Ben White wants Mary Anderson to refer to DATA3 in the previous purchase order PO and to revise DATA1 accordingly. A new instance is inserted into the OM table to allow Mary Anderson to refer to DATA3. Note that DATA4 of the same document cannot be seen by Mary Anderson, because its security level is higher.
- *Scenario 3.* Ben White modifies the content of DATA2 and approves the quotation. The clearance of DATA2 becomes level 2, because the security level of Ben White is 2. When this occurs, the instance for granting the write privilege of DATA2 to Mary Anderson will be removed. Following the workflow, John Karlson is selected to process the order matching, which is the job of Clerk 3. (Note that no one will be designated as Clerk 2 in the workflow instance, and, therefore, no Clerk 2 user can access data.)

- *Scenario 4.* As the DATA2 of quotation edition 1 has been set to clearance 2, John Karlson will not be able to do his job. There are two ways of resolving the problem: bring down the security level of data or upgrade the security level of the user. (This is called *attenuation of privilege*.) The former approach runs the risk of making sensitive data available to other users, while the latter approach risks other sensitive data being made available to the user. This study will adopt the second approach, by limiting the clearance to a certain workflow instance, since the privilege is maintained in the OM and only the clearance of workflow WF01 version 1 for John Karlson is enhanced.
- *Scenario 5*. After John Karlson has completed the job, all the privileges of the workflow instances will be changed to read in the OM table, and, therefore, no further modifications are allowed.

Figure 3 Tables used in the quotation preparation

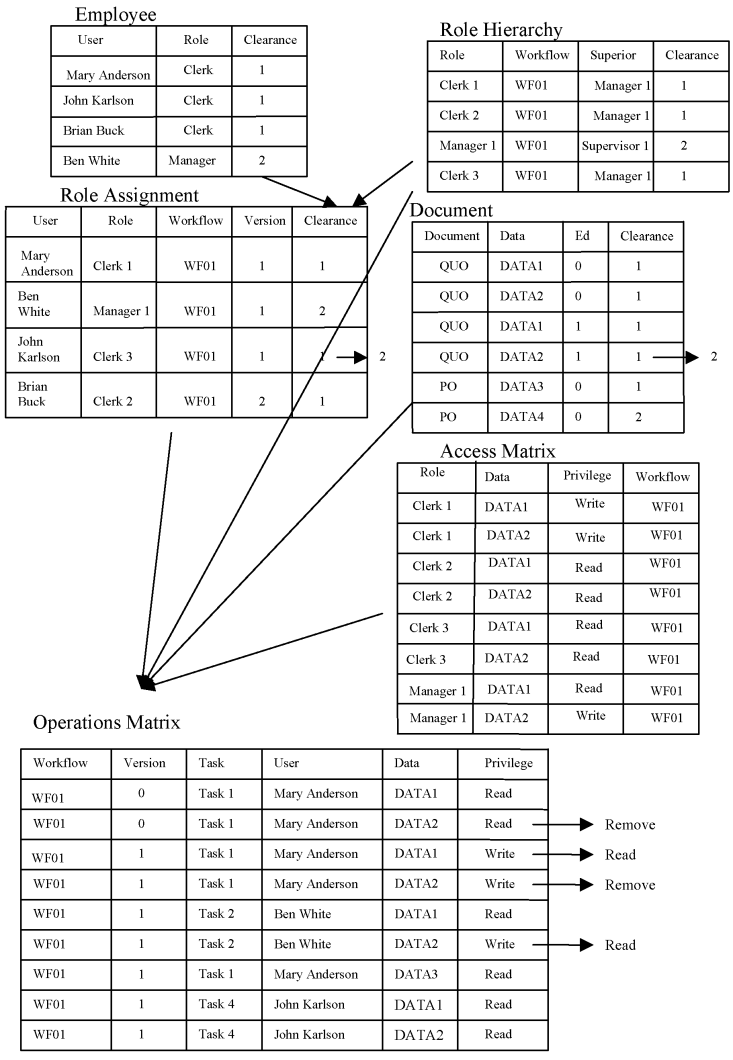


Figure 4 Illustration of programs for defining workflow instances with security setting

```
-- This function initiates the workflow instance and builds the role hierarchy of staff that work on it
procedure INITIATE_DOC    //define procedure and variables
   (itemtype  in varchar2,
   itemkey   in varchar2,
   actid     in number,
   funcmode  in varchar2,
   resultout    in out varchar2)
is
   lwork_name varchar2(10);  //workflow name
   lwork_version varchar2(3);  //workflow version
   lreviewer_app varchar2(30);
   lreviewer_rej varchar2(30);
   lcreator           varchar2(30); //document creator
   lapprover varchar2(30);
   lcreator_cla number(1);  //document clearance obtaining from the document creator

begin
-- Run Mode
 if (funcmode = 'RUN') then
  lwork_name := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
          'WORK_NAME');
  lwork_version := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
          'WORK_VERSION');
  lreviewer_app := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
          'REVIEWER_APP');
  lreviewer_rej := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
          'REVIEWER_REJ');
  lcreator := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname => 'CREATOR');
   -- Assign supervisors and clearances to different roles of the workflow
   insert into role_hierachy values ('clerk 1', lwork_name, 'manager 1', 1);
   insert into role_hierachy values ('clerk 2', lwork_name, 'manager 1', 1);
   insert into role_hierachy values ('manager 1', lwork_name, 'supervisor1', 2);
   insert into role_hierachy values ('clerk 3', lwork_name, 'manager 1', 1);

   -- The workflow designer initializes the electronic document and workflow
   -- Clerk 1 initiates a workflow instance and the assignment is save into Role Assignment table
   Select clearance into lcreator_cla from staff where staff_name = lcreator;
   insert into role_assignment values (lcreator, 'clerk 1', lwork_name, lwork_version, lcreator_cla);

 -- no result needed
   resultout := wf_engine.eng_completed||':'||wf_engine.eng_null;
   return;
 end if;

-- Cancel Mode
 if (funcmode = 'CANCEL') then
   -- no result needed
   resultout := wf_engine.eng_completed||':'||wf_engine.eng_null;
   return;
 end if;

-- Other
 resultout := wf_engine.eng_null;
 return;

exception
 when others then
   -- The line below records this function call if errors occur
   wf_core.context('WFDOC', 'INITIATE_DOC', itemtype, itemkey, to_char(actid), funcmode);
   raise;
end INITIATE_DOC;
```

5 Conclusions and recommendations

Sharing electronic documents among workflow activities involves several dependent relationships concerning the document, the data, the activities of the workflow, the role hierarchy of the organisation and the access mode. Having appropriate setting of an access control mechanism is important for sharing documents efficiently. This study has proposed a mandatory access control mechanism for managing the document access privileges of users. The mechanism was clearly demonstrated in Oracle Workflow and Oracle PL/SQL through the example of generating a quotation document. It gives the workflow managers a plausible option of controlling the access of shared electronic documents in organisational workflows.

Figure 5 Illustrated programs for adding instances into access matrix and operations matrix

```
-- This function loads the quotation to be processed and builds the access matrix of the data items
procedure LOAD_DATA   //define procedure and variables
    (itemtype  in varchar2,
    itemkey   in varchar2,
    actid     in number,
    funcmode  in varchar2,
    resultout    in out varchar2)
is
  lcreatorvarchar2(30);
  lapprover          varchar2(30);
  lreviewer_app varchar2(30);
  lreviewer_rej varchar2(30);
  lquotation_id varchar2(8);  //document
  ldataA_id varchar2(8);  //data 1 in the document
  ldataB_id varchar2(8);  //data 2 in the document
  ldataA_clearance number(1);  //clearance for data 1
  ldataB_clearance number(1);  //clearance for data 2
  lwork_name varchar2(10);
  lclearance number(1);  //variable for clearance
  lwork_version varchar2(3);

begin
-- Run Mode
  if (funcmode = 'RUN') then
    lcreator  := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
          'CREATOR');
    lapprover := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
          'APPROVER');
    lreviewer_app := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
          'REVIEWER_APP');
    lreviewer_rej := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
          'REVIEWER_REJ');
    lwork_version  := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
          'WORK_VERSION');
```

Figure 5 Illustrated programs for adding instances into access matrix and operations matrix (continued)

```
  lquotation_id  := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
        'QUOTATION_ID');
  ldataA_id  := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
        'DATAA_ID');
  ldataB_id  := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
        'DATAB_ID');
  ldataA_clearance  := wf_engine.GetItemAttrNumber(itemtype => itemtype, itemkey => itemkey, aname =>
        'DATAA_CLEARANCE');
  ldataB_clearance  := wf_engine.GetItemAttrNumber(itemtype => itemtype, itemkey => itemkey, aname =>
        'DATAB_CLEARANCE');
  lwork_name  := wf_engine.GetItemAttrText(itemtype => itemtype, itemkey => itemkey, aname =>
        'WORK_NAME');

  -- Find the clearance of clerk 1
    select clearance into lclearance from staff where staff_name = lcreator;

  -- The clearance of clerk 1 will used as the default clearances of data items as they are initialized by clerk 1
    insert into document values (lquotation_id, ldataA_id, lclearance);
    insert into document values (lquotation_id, ldataB_id, lclearance);

 -- Insert instances into Access Matrix and assign different privileges to different roles. It is noted that some
data are hard-coded here for illustration. The data can retrieve from related tables in the real applications.
    insert into daccess values ('clerk 1', ldataA_id, 'write', lwork_name);
    insert into daccess values ('clerk 1', ldataB_id, 'write', lwork_name);
    insert into daccess values ('clerk 2', ldataA_id, 'read', lwork_name);
    insert into daccess values ('clerk 2', ldataB_id, 'read', lwork_name);
    insert into daccess values ('clerk 3', ldataA_id, 'read', lwork_name);
    insert into daccess values ('clerk 3', ldataB_id, 'read', lwork_name);
    insert into daccess values ('manager 1', ldataA_id, 'read', lwork_name);
    insert into daccess values ('manager 1', ldataB_id, 'write', lwork_name);

  --  Update Operations Matrix to allow clerk 1 to read and modify the data items in quotation
    insert into operation values (lwork_name, lwork_version, 1, lcreator, ldataA_id, 'write');
    insert into operation values (lwork_name, lwork_version, 1, lcreator, ldataB_id, 'write');

  -- no result needed
    resultout  := wf_engine.eng_completed||':'||wf_engine.eng_null;
    return;
 end if;

-- Cancel Mode
 if (funcmode = 'CANCEL') then
        resultout := wf_engine.eng_null;
 return;
 end if;

-- Others
exception
 when others then
  -- The line below records this function call if errors occur
  wf_core.context('WFDOC', 'LOAD_DATA', itemtype, itemkey, to_char(actid), funcmode);
  raise;
end LOAD_DATA;
```

The access control mechanism proposed by this study is very simple but effective. To implement this mechanism, it is recommended that new instances be inserted into *Role Hierarchy* table when a new workflow is defined, that new instances be inserted into *Role Assignment* table when a new workflow instance is initiated, that new instances be

inserted into tables *Data and Document* table, *Access Matrix* and *OM* when a new document is created, and that new instances be inserted into *OM* when precedent activities are completed. Furthermore, the access privileges of all instances in *OM* of the same workflow instance should be changed into *read* when the workflow instance is completed. The general rules are:

- if a conflict of access appears at the access mode (i.e., read or write) of a data object, the lower privilege one (i.e., read) should be adopted
- if a conflict of access appears at the security level (i.e., clearance confliction of the grantors), the lower clearance object would be adopted (i.e., data objects with higher clearance cannot be accessed).

Through this mechanism of dynamically granting and revoking access rights, documents can be shared in an organisation effectively and efficiently. This control mechanism could be directly applied to the documents available on a web-based collaborative intranet, such as supply chain, healthcare network or strategic alliance network.

References

Allen, C. and Chow, V. (2000) *Oracle Certified Professional: Financial Applications Consultant Exam Guide*, Osborne McGraw-Hill, Berkeley.

Aslst, W. and Hee, K. (2002) *Workflow Management: Models, Methods Systems*, MIT Press, Cambridge, MA.

Atluri, V., Huang, W-K. and Bertino, E. (2000) 'A semantic-based execution model for multilevel secure workflows', *Journal of Computer Security*, Vol. 8, No. 1, pp.3–41.

Bertino, E., Ferrari, E. and Atluri, V. (1999) 'The specification and enforcement of authorization constraints in workflow management systems', *ACM Transactions on Information and System Security*, Vol. 2, No. 1, pp.65–104.

Castano, S., Fugini, M., Martella, G. and Samarati, P. (1995) *Database Security*, ACM Press and Harlow, Addison-Wesley, England.

Denning, D.E. (1975) 'A lattice model of secure information flow', *Communications of the ACM*, Vol. 19, No. 5, pp.236–243.

Dieng, R. (2000) 'Knowledge management and the internet', *IEEE Intelligent Systems*, Vol. 15, No. 3, pp.14–17.

Elmasri, R. and Navathe, S. (2000) *Fundamentals of Database Systems*, 3rd ed., Addison-Wesley, Reading MA.

Eloff, J.H.P., Holbein, R. and Teufel, S. (1996) 'Security classification for documents', *Computers and Security*, Vol. 15, No. 1, pp.55–71.

Herrmann, G. and Pernul, G. (1999) 'Viewing business-process security from different perspectives', *International Journal of Electronic Commerce*, Vol. 3, No. 3, pp.89–103.

Kamath, M. and Ramamritham, K. (1996) 'Correctness issues in workflow management', *Distributed Systems Engineering*, Vol. 3, No. 4, pp.213–221.

O'Meara, D. (2000) 'Buried in documents?', *Engineering Management Journal*, Vol. 10, No. 5, pp.241–243.

Sameshima, Y. and Kirstein, P. (1996) 'Secure document interchange: a secure user agent', *Computer Networks and ISDN Systems*, Vol. 28, pp.513–523.

Sandhu, R.S. (1993) 'Lattice-based access control models', *IEEE Computer*, Vol. 26, pp.9–19.

Sandhu, R.S., Coyne, E.J., Feinstein, H.L. and Youman, C.E. (1996) 'Role-based access control models', *IEEE Computer*, Vol. 29, pp.38–47.

Stohr, E.A. and Leon, Z.J. (2001) 'Workflow automation: overview and research issues', *Information Systems Frontiers*, Vol. 3, No. 3, pp.281–296.

Trcek, D. (1998) 'Minimizing the risk of electronic document forgery', *Computer Standards and Interfaces*, Vol. 19, No. 2, pp.161–167.

Turban, E., King, D., Lee, J.K. and Viehland, D. (2004) *Electronic Commerce 2004: A Managerial Perspective*, 3rd ed., Upper Saddle River, Prentice-Hall, New Jersey.

Weigand, H. and van den Heuvel, W-J. (2002) 'Cross-organizational workflow integration using contracts', *Decision Support Systems*, Vol. 33, No. 3, pp.247–265.

Wu, S., Sheth, A., Miller, J. and Luo, Z. (2002) 'Authorization and access control of application data in workflow systems', *Journal of Intelligent Information Systems*, Vol. 18, No. 1, pp.71–94.