Workflow authorisation in mediator-free environments

Mohamed Shehab*

Department of Electrical and Computer Engineering and
Center for Education and Research in Information
Assurance and Security (CERIAS),
Purdue University,
West Lafayette, IN, USA
E-mail: shehab@ecn.purdue.edu
*Corresponding author

Elisa Bertino

Department of Computer Science,
Electrical and Computer Engineering and CERIAS,
Purdue University,
West Lafayette, IN, USA
E-mail: bertino@cerias.purdue.edu

Arif Ghafoor

Electrical and Computer Engineering and CERIAS,
Purdue University,
West Lafayette, IN, USA
E-mail: ghafoor@ecn.purdue.edu

Abstract: WorkFlow Management Systems (WFMS) coordinate and streamline business processes. Acquiring workflow authorisations and managing workflow authorisation constraints is a challenging problem. Current WFMSs assume a centralised global workflow authorisation model. In this paper, we propose a distributed workflow authorisation model with no central authorisation manager for a mediator-free environment. We provide an on-demand task discovery protocol that enables domains to discover tasks available in other domains. We formulate the workflow authorisation problem as a constraint satisfaction problem to select access paths that satisfy all the workflow authorisation constraints. We propose the Workflow Minimal Authorisation Problem (WMAP), which selects minimal authorisations required to execute the workflow tasks. In addition, we investigate access path overlaps to allow tasks in the same session to share authorisations and we present the Workflow Minimal Authorisation Problem with path Overlaps (WMAPO). Finally, we formulate integer programmes to solve both the WMAP and WMAPO.

Keywords: decentralised workflow authorisation; decentralised secure interoperability; task discovery; role-based access control.

Reference to this paper should be made as follows: Shehab, M., Bertino, E. and Ghafoor, A. (2006) 'Workflow authorisation in mediator-free environments', *Int. J. Security and Networks*, Vol. 1, Nos. 1/2, pp.2–12.

Biographical notes: Mohamed Shehab received a BS degree in Electronic Engineering from United Arab Emirates University, United Arab Emirates, Al-Ain, in 2000. He is a PhD candidate in the School of Electrical and Computer Engineering at Purdue University, West Lafayette, Indiana, and is expected to graduate in May 2007. His research interests include information security, distributed access control, distributed workflow management systems and watermarking of relational databases.

Elisa Bertino is a Professor of Computer Science and Electrical and Computer Engineering at Purdue University and serves as a Research Director of CERIAS. Her main research interests include security and database systems. In those areas, she has published more than 250 papers. She is the coordinating editor-in-chief of the Very Large Database Systems Journal, and serves on the editorial boards of several journals. She is a Fellow of IEEE and ACM. She received the 2002 IEEE Computer Society Technical Achievement Award for outstanding contributions to database systems and database security and advanced data management systems.

Arif Ghafoor is currently a Professor in the School of Electrical and Computer Engineering at Purdue University, West Lafayette, Indiana, and is the Director of Distributed Multimedia Systems Laboratory and Information Infrastructure Security Research Laboratory. He has been actively engaged in research areas related to database security, parallel and distributed computing, and multimedia information systems and has published extensively in these areas. He has served on the editorial boards of various journals. He is a Fellow of the IEEE. He has received the IEEE Computer Society 2000 Technical Achievement Award for his research contributions in the area of multimedia systems.

1 Introduction

Globalisation has removed the barriers between markets, organisations, researchers and societies. In such a connected world, there are immense possibilities of collaboration in distributed environments. In recent years, WorkFlow Management Systems (WFMSs) have gained importance in both business and research fields (Shankar et al., 2005; Stohr and Zhao, 2001). WFMSs are designed to automate processes by efficiently coordinating and controlling the flow of work between participants. The need for WFMSs is essential as enterprises are continuously splitting processes into several tasks and migrating such tasks across organisational boundaries to combine their efforts and become virtual enterprises (Afsarmanesh et al., 1998; Desai and Awad, 2005; Ludwig et al., 1999; Ramnath and Landsbergen, 2005). Furthermore, in recent years, there has been an increasing demand for scientific workflows (Altintas et al., 2004; Franklin and Liu, 2004) to allow scientific institutions to collaborate in the management and analysis of the vast quantities of data generated by their scientific experiments.

Such large-scale workflows will involve the collaboration of several distributed entities across several domains. A domain is a separate, autonomous entity that manages a group of resources, provides services and is capable of performing several tasks. Domains have their own administration and access control polices. Acquiring the authorisations required to access resources and perform tasks is a very important issue in workflow management. Another important issue is handling workflow authorisation constraints between the workflow tasks. Current workflow authorisation models (Atluri et al., 2001; Bertino et al., 1999) assume a centralised workflow management system capable of managing workflow authorisations and handling workflow authorisation constraints. The major limitation in current workflow management systems models (Atluri et al., 2001; Bertino et al., 1999) is the requirement that all the roles and tasks in all the collaborating domains are known in advance by the workflow management system. This inherently requires the WFMS to have a global view of the collaboration environment, which is not feasible in a dynamic mediator-free environment with a large number of collaborating domains, where domains have a limited view of the collaboration environment and where domains dynamically join and leave the collaboration environment. Moreover, current WFMS models assume that the workflow manager is able to access and schedule any task to any role in any domain. With such assumptions the WFMS acts as a central mediator having precedence over all the access control policies of all the domains in the collaboration environment.

Handling workflow authorisation in fully distributed environments where none of the domains has a global view of the collaboration environment is a challenging task. In this paper, we propose a distributed workflow authorisation model where there is no central WFMS. We propose a task discovery protocol, which is used by a WFMS located at any of the collaborating domains, to discover authorised tasks that can be accessed from these domains. We provide an extended workflow authorisation constraints for authorisations in a mediator-free environment. We propose workflow authorisation mechanisms capable of assigning the minimal required authorisation to perform the required tasks. Our proposed framework is built on top of our SEcure Role mApping Technique (SERAT) for mediator-free environments to ensure secure interoperability (Shehab et al., 2005a,b).

1.1 Contributions and paper organisation

The contributions in this paper can be summarised as follows:

- We present a workflow authorisation model for mediator-free multidomain collaboration environments.
- We provide an on-demand task discovery protocol that enables domains to discover tasks available through other domains using a distributed protocol, while respecting the access control policies of the involved domains.
- We define an extended model for workflow authorisation constraints to accommodate constraints involving access paths, tasks and sessions.
- We present three workflow authorisation problems:
  - The Workflow Authorisation Problem (WAP): selects authorisations that satisfy the authorisation constraints.
  - The Workflow Minimal Authorisation Problem (WMAP): selects minimal authorisations required to execute specified workflow tasks while at the same time satisfying the authorisation constraints.
  - The Workflow Minimal Authorisation Problem with Path Overlaps (WMAPO): selects authorisations that can be shared for performing several tasks, and at the same time satisfies the same requirements of WMAP.

The rest of this paper is organised as follows. In Section 3, we review the requirements of secure interoperability

and the mediator-free secure collaboration. We introduce the challenges involved in workflow authorisation in mediator-free collaboration environments. In Section 4, we present the on-demand task discovery protocol. The workflow authorisation constraints are presented in Section 5. The workflow authorisation constraint satisfaction and optimisation problems are discussed in Section 6. The related work is presented in Section 7. Concluding remarks are added in Section 8.

2 Preliminaries

In our framework, we assume that all the domains adopt a Role-Based Access Control (RBAC) model (Ferraiolo et al., 2001, 2003) to model their access control policies. The analysis performed in this paper can still be applied when other access control models are adopted. RBAC was chosen because it is suitable for specifying the security requirements for a wide range of commercial, medical, government applications (Bertino et al., 1999; Sandhu et al., 1996) and moreover it is being standardised. A domain that does not use RBAC as its access control model can easily generate an export RBAC policy to join the collaboration.

In RBAC, permissions are associated with roles, and users are granted membership in appropriate roles, thereby acquiring the roles' permissions. The access control policy for domain D_i is modelled as a directed graph $G_i = \langle V_i, A_i \rangle$ where the vertex set V_i represents roles and the arcs set A_i represents the dominance relationship between roles. For example, if role r_1 dominates r_2, $(r_2 \leq r_1)$, then $(r_1, r_2) \in A_i$. Thus, a user acquiring role r_1 can acquire permissions assigned to role r_2 by using the RBAC permission inheritance properties (Crampton, 2003). For $r_x, r_y \in V_i$ an access link (r_x, r_y) is legal if and only if $(r_x, r_y) \in G_i^+$ where G_i^+ is the transitive closure of $G_i = \langle V_i, A_i \rangle$. We denote a legal access link by $(r_x, r_y) \propto A_i$.

2.1 Secure interoperability

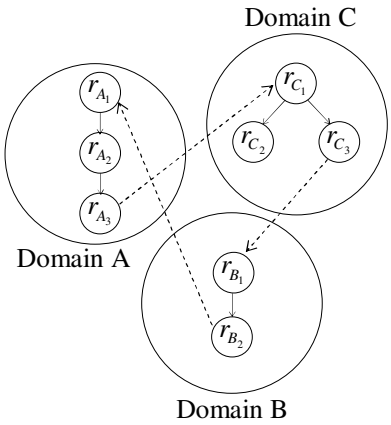
According to Gong and‘Qian (1994, 1996), collaboration among n domains can be achieved by introducing cross domain pairwise mappings between roles in different domains. These mappings relate roles in different domains, and are represented by a set of cross domain arcs referred to as the set F (Figure 1). Solutions developed for schema matching in the area of heterogeneous database systems and more recently approaches based on ontologies (Madhavan and Halevy, 2003; Madhavan et al., 2005) can be used for generating such links. The details of such approaches are outside the scope of this paper. In this study, we assume that the cross domain mappings are selected by the administrators of the domains according to the interoperability requirements of each system. Furthermore, the system administrators agree on a set of restricted accesses which is similar to negative authorisations adopted in several access control models. The restricted access is a binary relation R on $\cup_{i=1}^{n} V_i$ such that $\forall (u, v) \in R, u \in V_i, v \in V_j$ and $i \neq j$, where these edges in R are prohibited to exist during interoperation.

Given n domains $G_i = \langle V_i, A_i \rangle, i = 1, \ldots, n$, set of cross links F and a restricted access relation R, an interoperation $Q = \langle \cup_{i=1}^{n} V_i, A_Q \rangle$, where A_Q is the resulting arc set $A_Q \subseteq \{\cup_{i=1}^{n} A_i \cup F\}$, is secure according to Gong and Qian (1994, 1996) if it satisfies all the following conditions:

1 $A_Q \cap R = \emptyset$ and

2 $\forall u, v \in V_i$, (u, v) is legal in A_i if and only if (u, v) is legal in A_Q.

Figure 1 An example of three collaborating domains. The solid lines show the internal access links, while the dotted lines show the interoperation cross links F



2.2 Mediator-free secure collaboration

In a mediator-free collaboration environment, there is no central mediator or trusted party managing and ensuring secure interoperability among the collaborating domains. The mediator has a global view of all the access control policies of the collaborating domains and the cross links between them.

A mediator-free collaboration is a completely distributed form of collaboration. In this environment, the domains have to collaborate in making access control decisions to avoid violations. In a mediator-free environment, none of the collaborating domains has the global view of all the access control policies; instead the domains view the collaboration environment only through their established cross links. In our previous work (Shehab et al., 2005a,b), we presented a framework for secure collaboration in mediator-free multidomain environments (SERAT). This framework is based on using the access history to dynamically make access control decisions. The access history is referred to as the *access path*, which is the sequence of roles acquired from the home domain to target domains in the collaboration environment. Using the access path to make access control decisions shares ideas with the Chinese Wall security policy (Brewer and Nash, 1989), as the access history controls future accesses. Using access path information when making access control decisions enables domains to prevent security violations as access requests are being made. Furthermore, it enables domains to make localised access control decisions without the need for the global view of the collaboration environment. SERAT provides a path authentication technique that generates signatures to prove the authenticity of access paths.

3 Workflow authorisation

A workflow is composed of a set of tasks, and a set of task dependencies that control the coordination among the tasks. *WFMS* provide facilities to define, manage, schedule and execute business processes by coordinating and controlling the flow of work and information between groups of interacting entities. The tasks in a workflow are carried out by several users in accordance with the organisational rules relevant to the process represented by the workflow. Roles in a domain represent agents that are able to perform certain tasks; we refer to the relationships between roles and tasks as the *role-to-task assignments*, which can easily be derived from the role-permission assignments in the RBAC model. The WFMSs locate a suitable set of roles that are able to execute the different workflow tasks. The *workflow role specification* is the association of roles with tasks in a workflow (Bertino et al., 1999). The selection of roles to be associated with tasks is controlled by several workflow authorisation constraints; a common type of constraints are the separation of duty constraints (Clark and Wilson, 1987).

Current WFMS models (Atluri et al., 2001; Bertino et al., 1999) assume that WFMS is a centralised mediator that has global view of all the roles and tasks available in the collaboration environment. Furthermore, in such WFMS models, collaboration among domains is only through the WFMS which is a very limited form of collaboration when compared to multihop collaboration in mediator-free environments. Figure 2(a) shows the centralised WFMS in a collaboration environment, where there is a single central WFMS controlling the interaction among all the collaborating domains.

We propose a distributed WFMS framework in which each domain D_i in the collaboration environment encloses its own WFMS; furthermore in each domain the WFMS is assigned a specific access role. Figure 2(b) shows the distributed WFMS. In this framework, WFMS do not have a global view of the collaboration environment. Furthermore, the WFMS could only have access to roles in other domains via secure paths from its home domain to the target domain. The design of a WFMS capable of managing workflow authorisations in mediator-free environments has several challenging issues:

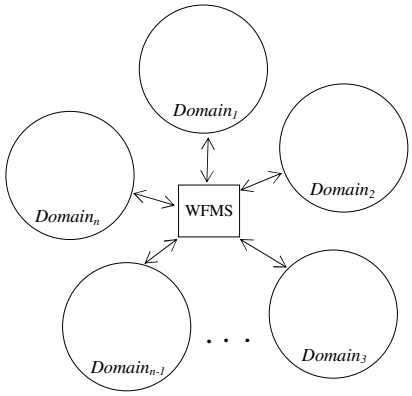
C1 Task discovery: a WFMS in mediator-free environment has a limited view of the collaboration environment, thus a task discovery protocol is needed to discover tasks in other domains.

C2 Enforcing Path Constraints: in a mediator-free environment authorisations are acquired by generating access paths. Authorisation constraints inherently should be applied on access paths, thus WFMS should be extended to handle such constraints.

C3 Minimal Authorisation Assignment: in a mediator-free environment there are multiple paths to perform certain tasks, WFMS should be able to select the set of paths that minimise the required authorisations, or domains involved, in order to minimise the risks of security breaches.

In the following sections, we discuss each of the above challenges in detail, and propose suitable solutions for each challenge.

Figure 2 WFMS and n collaborating domains: (a) centralised WFMS and (b) distributed WFMS



4 Task path discovery

A workflow consists of several tasks that are to be executed according to a predefined sequence. WFMS should be able to discover roles in the collaboration environment that are able to execute the workflow tasks. Each domain D_i in a mediator-free environment has a workflow management system WFMS_i, with respect to which we assume the following:

A1 WFMS_i is assigned an access role r_i^{WF} in domain D_i.

A2 WFMS_i has a limited view of the collaboration environment offered by cross-links $F_i \subseteq F$ that are established with domains neighbouring domain D_i.

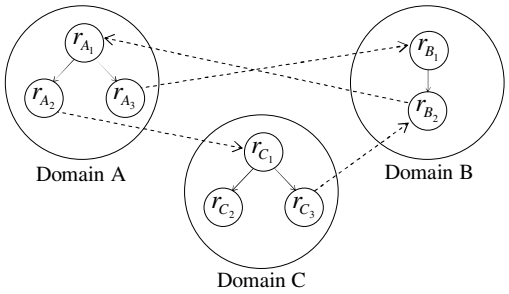
A role r_i^j in domain D_i is capable of executing a set of tasks referred to as $r_i.TS = \{T_1^i, \ldots, T_{n_k}^i\}$. Note that if a role r_i dominates role r_j, $r_j \leq r_i$, then by the semantics of the dominance relationship then r_i can acquire tasks performed by acquiring r_j. A path P is composed of a

set of acquired roles $\{r_1, r_2, \ldots, r_n\}$, the tasks that can be executed by acquiring such a path is defined by the set *PTS* as follows:

$$PTS = \bigcup_{r \in P} rTS$$

According to the above definition, a path P is authorised to execute a task T_i if $T_i \in PTS$. Figure 3(a) shows an example collaborative environment composed of three domains, and Figure 3(b) shows the role-task assignment for each of the roles. For example, path $P = \{r_{A_1}, r_{A_2}, r_{C_1}, r_{C_2}\}$ is able to execute tasks $PTS = \{T_1, T_2, T_7, T_8, T_9\}$.

Figure 3 Examples of a collaborative environment and role-task assignments



(a)

$r_{A_1}.ST$	$\{T_1\}$
$r_{A_2}.ST$	$\{T_2\}$
$r_{A_3}.ST$	$\{T_3, T_4\}$
$r_{B_1}.ST$	$\{T_5\}$
$r_{B_2}.ST$	$\{T_6\}$
$r_{C_1}.ST$	$\{T_7, T_8\}$
$r_{C_2}.ST$	$\{T_9\}$
$r_{C_3}.ST$	$\{T_{10}\}$

(b)

The WFMS in each domain is assigned an access role, this enables the WFMS to discover paths from this role to other roles able to execute all the workflow tasks. Following from the example in Figure 3, we assume that the WFMS in domain A is assigned to role r_{A_1}. Figure 4(a) shows an example workflow to be executed by the WFMS in domain A, and Figure 4(b)–(f) show the paths from r_{A_1} to roles capable of executing tasks T_2, T_9, T_{10}, T_7 and T_6, respectively. This model is very suitable for scientific workflows, as it enables a user operating at a role of a 'scientist' to be able to execute a workflow by discovering paths to roles able to execute the intended tasks.

The paths discovered should respect the access control requirements of their involved domains. To ensure such requirements, we use the path linking rules and path authentication that were used by the SERAT framework for secure collaboration in mediator-free environments the interested reader is referred to Shehab et al. (2005a,b). In this section, we present a task discovery protocol that enables domains to discover access paths to roles in other domains, whether reachable through one or more intermediate domains. The presented protocol is an on-demand path discovery, which is initiated only when domains require access paths to roles able to execute certain tasks.

4.1 On-demand task path discovery

The on-demand task discovery is initiated whenever a WFMS in a domain D_i that is assigned a role r_i^{WF} needs to discover secure access paths to roles in other domains in the collaboration environment that are able to execute task t_k. The task discovery protocol is composed of two types of messages, namely the path request and path reply messages. The domain D_i generates a path request message requesting a path from role r_i^{WF} to a role able to execute t_k and the request is sent on all its neighbouring domains through the outgoing cross links, referred to as $F_i^O \in F_i$, reachable from r_i^{WF}.

On receiving the path requests, the neighbouring domains check the authenticity of the received path signature generated by SERAT's path authentication technique. The path request is checked against SERAT's path linking rules to ensure that the path request does not violate the access control policy of the neighbouring domain. If the path request fails any of the above checks, then the request is dropped. Otherwise, the neighbouring domain appends the entry role to the received path, if the new path is able to execute task, t_k, then a path reply is sent to the home domain D_i, otherwise a path request is generated and forwarded to its neighbours. The path request propagates from domain to domain while obeying the access control requirements of each domain until the requested task is located. The home domain, D_i, waits for a timeout period of $T_{\max}$, if no reply arrives within this period, then there are no secure paths from home domain to roles able to execute the requested task t_k. The value of $T_{\max}$ is assigned based on the number of collaborating domains.

The proposed protocol is similar to on-demand routing (Maltz et al., 1999; Perkins and Royer, 1999); however, task discovery is more complicated as it includes several other constraints such as the path linking rules and the task checking. To avoid path loops, which represent repeated authorisations, requests are not forwarded to domains already included in the currently accumulated path. To prevent the path size from increasing indefinitely, the path length is checked against a maximum path length ($P_{\max}$) and if exceeded the request is dropped. Figure 5 shows the algorithm executed when domain D_j receives a path request from a neighbouring domain, where PS and PD represent the path signature and domains included in path P, respectively.

Figure 6 shows the different stages involved in the discovery of task T_6 from role r_{A_1} in domain A. Domain A sends a path requests REQ_1 and REQ_2 to domains B and C requesting task T_6 (see Figure 6(a)). Domain B receives request REQ_1 and realises that r_{B_2} can execute task T_6, so it sends a path reply $REPLY_1$ to A having an accumulated access path $P_1 = \{r_{A_1}, r_{A_3}, r_{B_1}, r_{B_2}\}$ (see Figure 6(b)). Domain C receives request REQ_2 and forwards it to its neighbouring domains, which include only domain B, (see Figure 6(b)). On receiving REQ_2, domain B sends a path reply $REPLY_2$ to A having an accumulated access

Figure 4 Path discovery example, where $r_A^{\mathrm{WF}} = r_{A_1}$: (a) workflow task diagram, (b) path for task T_2, (c) path for task T_9, (d) path for task T_{10}, (e) path for task T_7 and (f) path for task T_6

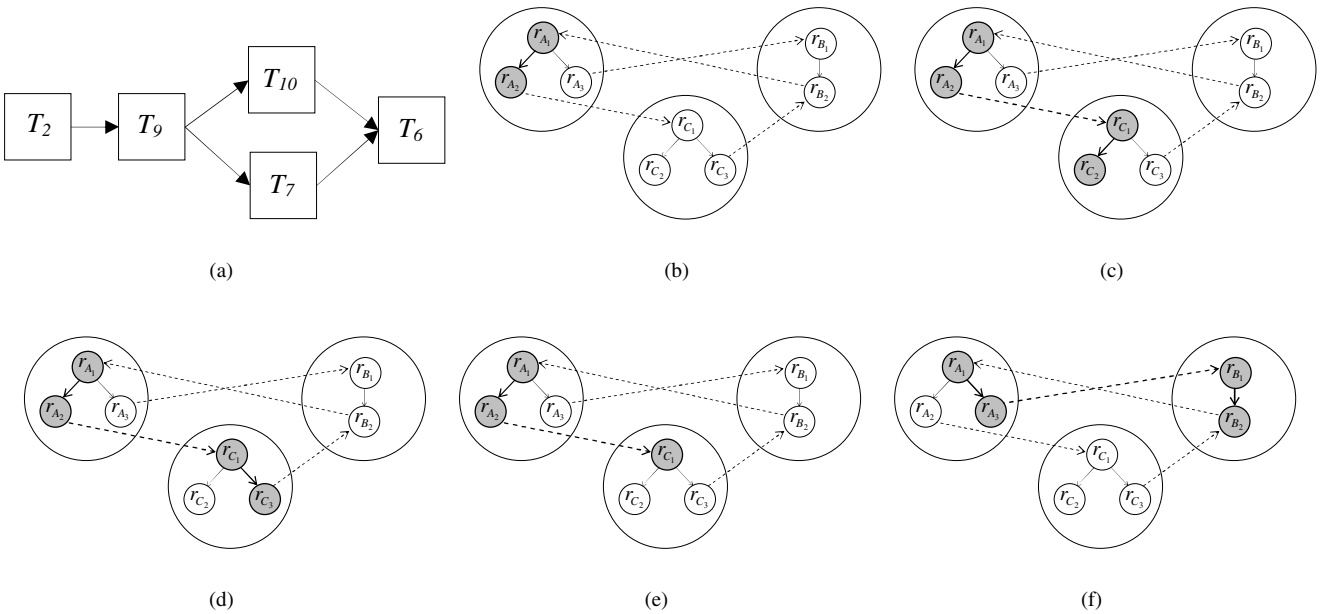


Figure 5 Algorithm executed by domain D_j upon receiving a path request

Input: Request = (Requested Task t, Path P) arriving to domain D_j from domain D_i via cross link (r_i^X, r_j^E).
Algorithm:

1. Check Path signature $P.S$, if invalid drop request, End.
2. Check the Path linking rules on P and link (r_i^X, r_j^E), if insecure path then drop request, End.
3. If $|P| > Pmax$ then drop request, End.
4. Update path $P_{new} = P \circ r_j^E$,where $'\circ'$ is the concatenation operator.
5. If $t \in P_{new}.TS$
   (a) Generate path signature $P_{new}.S$
   (b) Send Path Reply = (P_{new}) to Home domain
6. For all cross links $L = (r_j^X, r_k^E) \in F_j^O$ and $(r_j^E, r_j^X) \propto A_j$ and domain $D_k \notin P.D$,
   (a) Update path $P_{new} = P \circ r_j^E \circ r_j^X$
   (b) Generate path signature $P_{new}.S$
   (c) Send Request = $(t,\ P_{new})$ to domain D_k
7. End.

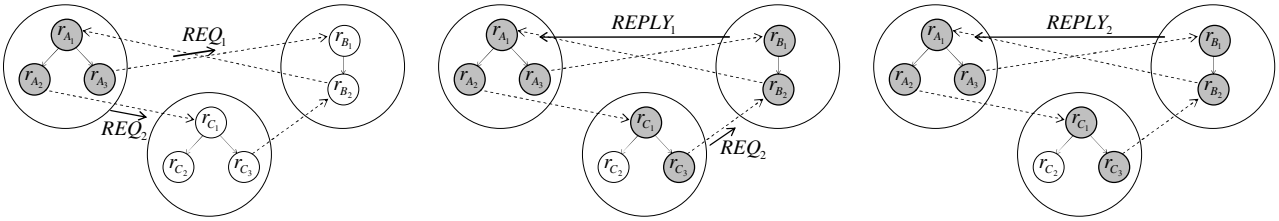
path $P_2 = \{r_{A_1}, r_{A_2}, r_{C_1}, r_{C_3}, r_{B_2}\}$, see Figure 6(c). Note that the presented protocol can be easily updated, with very minor changes, to return paths to a set of tasks instead of a single task. In such a case, a WFMS can discover access paths to all the required tasks by sending a single path request message.

The major advantage of on-demand task discovery is that it saves network bandwidth because it limits the amount of bandwidth consumed in the exchange of task discovery information by maintaining paths to only those target domains to which the domains need to collaborate with. The home domain could include constraints on the requested path, to further reduce the path discovery traffic. For example, the request could include a list of domains that should or should not be included in the path discovery. On-demand task discovery also obviates the need for disseminating path discovery information periodically, or flooding such information whenever a cross link changes or when a domain leaves or joins the collaboration environment. The primary problem is the large latency at the beginning of the collaboration caused by propagation of the path request message.

5 Workflow authorisation constraints

WFMS should be able to locate authorisations to execute the tasks included in the workflows to be executed. Furthermore, WFMS must ensure that authorisation constraints between the workflow different tasks are enforced. Authorisation constraints control the interaction between the tasks of a workflow. For example, the role assigned to task T_i should be different from the role assigned to task T_j. Authorisations in a mediator-free environment are acquired by generating access paths to roles in other domains. Thus, the workflow authorisation constraints should be enforced on

Figure 6 Task discovery of task T_6 from role r_{A_1}



(a) (b) (c)

the access paths and not only on roles as in current workflow authorisation systems (Atluri et al., 2001; Bertino et al., 1999).

- *Strict Path Constraints*: a constraint on a path $P = \{r_1, \ldots, r_n\}$ implies a constraint on all the roles $r_i \in P$. For example, give $P_i = \{r_1^i, \ldots, r_n^i\}$ and $P_i\{r_1^j, \ldots, r_n^j\}$, then $P_i \leq_S P_j$ implies that for all $r_u^i \in P_i, r_v^j \in P_j$ and $Domain(r_u^i) = Domain(r_v^j)$ then $r_u^i \leq r_v^j$.

- *Weak Path Constraints*: a constraint on a path $P = \{r_1, \ldots, r_n\}$ implies only a constraint on subset of the roles in P, for example, it can only be on the last role r_n in the path.

Many researchers have highlighted the importance and use of Separation of Duty (SoD) constraints in RBAC models (Ferraiolo et al., 2001, 2003; Li et al., 2004), and in workflow systems (Bertino et al., 1999). In their simplest form, SoD constraints require that for a particular set of tasks, no single role be allowed to execute all tasks within the set. For example, the role authorising a payment should be different from the one issuing it. We extend the SoD constraints from the role level to include other levels such as access paths, tasks and sessions. Sessions allow more than one task to execute and share acquired authorisations, as will be discussed further in Section 6.2. The following presents the different levels that the SoD constraints could be applied:

- *Path Level*: SoD constraints could be of the form that a path P_i should not include roles r_i and r_j. More generally, this constraint could require that the path P_i should not include t or more roles in a set of m roles $\{r_1, \ldots, r_m\}$.

- *Task Level*: task level SoD constraints ensure that paths assigned to different tasks do not overlap. Assuming tasks T_i and T_j are assigned to access paths P_i and P_j, respectively, then the constraints are enforced on the overlap or intersection of both paths, $P_i \cap P_j$. More generally, this constraint could require that no more than t paths overlap in a set of m paths $\{P_1, \ldots, P_m\}$. Furthermore, another form is that if task T_i is executed by domain D_i then task T_j should be executed by a domain other than D_i.

- *Session Level*: session level SoD constraints ensure that certain tasks are not included in the same session. For example tasks T_i and T_j should not be executed in the same session. More generally, this constraint could require that no more that t tasks are selected from a set of m tasks $\{T_1, \ldots, T_m\}$.

Cardinality constraints could also be applied at the above levels. At the path level, path P_i should not have more than t roles from a set of domains. At the task level, if path P_i is assigned to task T_i then the number of roles in P_i should be less than some t. At the session level, session S_k should include at most t. Other types of constraints enforce orderings among roles, paths and tasks. The above constraints are applied to the workflow to constitute the workflow authorisation constraints.

6 Workflow authorisation problems

Authorisations in a mediator-free environment are acquired by establishing access paths. WFMS use the path discovery protocol to discover access paths able to execute the workflow tasks. The workflow constraints define the authorisation constraints among the workflow tasks. The workflow authorisation problem is informally defined as the problem of assigning access paths, which represent authorisations, to the workflow tasks such that all the workflow constraints are satisfied. We define the Workflow Authorisation Problem (WAP) as follows:

Definition 1: *WAP. Given a workflow W, comprised of n tasks $\mathfrak{T} = \{T_1, \ldots, T_n\}$ to be executed sequentially, such that each task T_i has m_i discovered access paths $P^i = \{P_1^i, \ldots, P_{m_i}^i\}$, and a set of workflow constraints $C = \{c_1, \ldots, c_q\}$. Find a workflow task-path labelling $W.PA = \{P^{1*}, \ldots, P^{n*}\}$, where $P^{i*} \in P^i$, that satisfies all the constraints in C.*

The WAP is a Constraint Satisfaction Problem (CSP), which can be solved using CSP techniques (Marriott and Stuckey, 1998; Dechter, 2003). The problem is solved locally by the domain's WFMS after path discovery is completed. The solution of the WAP satisfies all the workflow constraints; however, the solution has no optimisation criteria controlling the number of the acquired authorisations or roles acquired through the established access paths. In other words, the solution to the WAP is not guaranteed to satisfy the *Principle of Least Privilege* (Saltzer and Schroeder, 1975), which requires that each principal be accorded the minimum access

privileges needed to accomplish its task. In the context of the WAP, the principle of least privilege implies choosing access paths that minimise the path authorisations. The following sections provide solutions that ensure that this principle is satisfied.

6.1 Minimal workflow authorisation

Assume each role r_i is assigned a positive cost $r_i.c$; this cost is assigned by the WFMS solving the optimisation problem. The cost could represent several metrics; for example, it could represent the degree of trust between the WFMS domain and the domain where the role resides, or it could be based on the effort involved in establishing trust with the domain of role r_i. The role costs could be fixed or dynamically assigned based on the reputation of the respective domains in the collaboration environment. An access path is a sequence of roles; we define the cost of an access path as follows:

Definition 2: *Let $P = \{r_1, r_2, \ldots, r_n\}$ be an access path, and $r_i.c$ denote the cost of role r_i. The cost of path P is defined as follows:*

$$\Phi(P) = \sum_{r_i \in P} r_i.c$$

Note that, when all the role costs $r_i c$ are set to a constant β then the path cost $\Phi(P) = \beta|P|$, which is proportional to the path length $|P|$, with equality when $\beta = 1$. To satisfy the principle of least privilege the optimisation problem formulated should minimise the cost of the access paths selected. We define the WMAP as follows:

Definition 3: *WMAP. Given a workflow W, comprised of n tasks $\mathfrak{T} = \{T_1, \ldots, T_n\}$ to be executed sequentially, such that each task T_i has m_i discovered access paths $P^i = \{P^i_1, \ldots, P^i_{m_i}\}$, and a set of workflow constraints $C = \{c_1, \ldots, c_q\}$. Find the workflow task path labelling $WPA = \{P^{1*}, \ldots, P^{n*}\}$, where $P^{i*} \in P^i$, that minimises $\sum_{i=1}^{n} \Phi(P^{i*})$ and satisfies all the constraints in $\mathcal{C}$.*

The WMAP is an optimisation problem that chooses access paths having low costs; in other words, if all the role costs are set to 1, then the optimisation problem would choose the shortest access paths. Shortest paths are paths having the least number of roles, which implies the paths with the least privileges. Also note that the solution of the optimisation problem is affected by the assigned role costs $r_i.c$, this enables the WFMS to indicate its role preferences. For example, if the WFMS favour certain roles then it should assign to them low costs and high costs if otherwise. In what follows we formulate the WMAP as an integer program. For each path $P^i_j \in P^i$ for task T_i, we introduce a decision variable $x_{ij} \in \{0, 1\}$, where $x_{ij} = 1$ if path P^i_j is assigned to task T_i and $x_{ij} = 0$ otherwise. Figure 7 shows the association of decision variables with paths and tasks. The problem is formulated as follows:

$$\min_{x_{ij}} \quad \sum_{i=1}^{n} \sum_{j=1}^{m_i} x_{ij} \Phi(P^i_j) \tag{1}$$

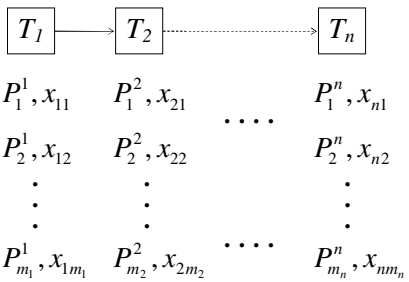
$$\text{s.t.} \quad \sum_{j=1}^{m_i} x_{ij} = 1 \quad \text{for } i = 1, \ldots, n \tag{2}$$

$$c_l \in C \qquad \text{for } l = 1, \ldots, q \tag{3}$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \ldots, n, \quad j = 1, \ldots, m_i \tag{4}$$

The objective function we are minimising is the sum of all the paths assigned to the tasks, note that x_{ij} is multiplied by $\Phi(P^i_j)$ to account only for the assigned paths. The first constraint ensures that each task is assigned only a single path. The remaining constraints ensure that all the workflow constraints $c_l \in \mathcal{C}$ are also satisfied. The formulated problem is not only a constraint satisfaction problem; instead it seeks to satisfy the constraints as well as to find paths with minimal costs. To solve this optimisation problem, techniques for constrained integer programming (Nemhauser and Wolsey, 1988) and resource allocation (Ibaraki and Katoh, 1988) could be used. Note that the presented optimisation problem is solved at a localised level by the WFMS in each domain without the need for a global view of the collaboration environment.

Figure 7 Tasks, paths and decision variables



6.2 WMAP with path overlaps

The WMAP provides a minimal path assignment to tasks in the workflow; however, it assumes that tasks are assigned paths without investigating savings that could be gained by the sharing established paths among different tasks. To investigate further possible optimisations, we introduce sessions in which multiple tasks can share authorisations by making use of path overlaps among paths assigned to the tasks in the same session. A session is composed of a Set of Tasks (ST), which is a subset of the workflow tasks, and a session lifecycle that indicates the sequence in which the tasks in ST should be executed and the sequence in which their access paths are to be generated. A session S_k is composed of tasks $\{T^k_1, \ldots, T^k_{m_k}\}$, where task T^k_{i+1} is to be executed after task T^k_i, then the path P_{i+1} assigned to task T^k_{i+1} should make use of the overlap with path P_i, which was assigned to task T^k_i to make use of already acquired roles. We define the path overlap as follows:

Definition 4: *Given two paths $P_u = \{r^u_1, \ldots, r^u_{m_u}\}$ and $P_v = \{r^v_1, \ldots, r^v_{m_v}\}$. The path overlap of P_u and P_v is their left matching subsequence defined as follows:*

$$\sigma(P_u, P_v) = \{r^O_i : (r^O_i = r^u_i \in P_u) \wedge (r^u_j = r^v_j, \forall j = 1, \ldots, i)\}$$

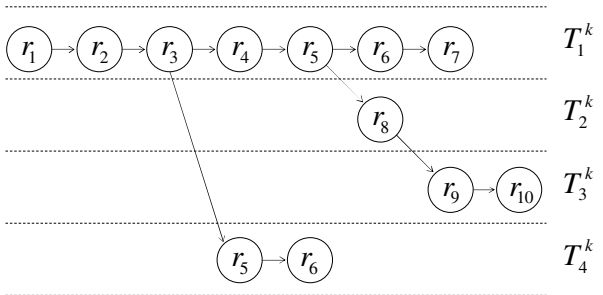
For example, if $P_u = \{r_1, r_2, r_3, r_4, r_5, r_6\}$ and $P_v = \{r_1, r_2, r_3, r_7, r_8, r_5, r_6\}$, then $\sigma(P_u, P_v) = \{r_1, r_2, r_3\}$. Note that, if P_v is to be established after P_u then the overlap $\sigma(P_u, P_v)$ could be used as it was already established for P_u, and only $P_v \setminus \sigma(P_u, P_v)$ need to be established to generate P_v. Exploiting such overlap among paths of consecutive tasks in the same session makes it possible to use already acquired roles and obviates the need for reestablishing acquired authorisations. Figure 8 shows a session composed of four tasks $\{T_1^k, T_2^k, T_3^k, T_4^k\}$, where path for T_i^k is used to generate the path for T_{i+1}^k. For a session S_k composed of tasks $\{T_1^k, \ldots, T_{m_k}^k\}$, where P_i^k is the access path assigned to task T_i^k, we define the session overlap cost as follows:

$$\Upsilon(S_k) = \sum_{i=1}^{m_k - 1} \Phi(\sigma(P_i, P_{i+1}))$$

Path sharing is possible if multiple tasks share the same session while respecting any constraints enforced on these sessions, such as SoD constraints related to sessions, for example, Task T_i and T_j should not be performed in the same session. We define the Workflow Minimal Authorisation Problem with path Overlaps (WMAPO) problem as follows.

Definition 5: *WMAPO. Given a workflow W, comprised of n tasks $\mathfrak{T} = \{T_1, \ldots, T_n\}$ to be executed sequentially, such that each task T_i has m_i discovered access paths $P^i = \{P_1^i, \ldots, P_{m_i}^i\}$, n sessions $S_1, \ldots, S_n$ and a set of workflow constraints $\mathcal{C} = \{c_1, \ldots, c_q\}$. Find the workflow task path labelling $WPA = \{P^{1*}, \ldots, P^{n*}\}$ and a task-session assignment $S_k.T = \{T_1^k, \ldots, T_{m_k}^k\}, k = 1, \ldots, n$, where $P^{i*} \in P^i$ and T_j^k is a task assigned to session k, that minimises $\sum_{i=1}^{n} \Phi(P^{i*}) - \alpha \sum_{k=1}^{n} \Upsilon(S_k)$ and satisfies all the constraints in $\mathcal{C}$, where $\alpha > 0$.*

Figure 8 Path sharing in session S_k, $S_k T = \{T_1^k, T_2^k, T_3^k, T_4^k\}$



In what follows, we formulate the WMAPO as an integer program which accommodates both path selection and task-session assignment. We assign each task T_i a decision variable y_{ik}, where $y_{ik} = 1$ if task T_i is assigned to session k and $y_{ik} = 0$ otherwise. The problem is formulated as follows:

$$F(X) = \sum_{i=1}^{n} \sum_{j=1}^{m_i} x_{ij} \Phi(P_j^i)$$

$$G(X, Y) = \sum_{k=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{\substack{u=1 \\ u>i}}^{n} \sum_{v=1}^{m_u} x_{ij} x_{uv} y_{ik} y_{uk} \Phi(\sigma(P_j^i, P_v^u)) 1\left\{ \sum_{t=i}^{u} y_{tk} \stackrel{?}{=} 2 \right\}$$

$$\begin{aligned}
\min_{x_{ij}} \quad & F(X) - \alpha G(X, Y) \\
\text{s.t.} \quad & \sum_{j=1}^{m_i} x_{ij} = 1 \quad \text{for } i = 1, \ldots, n \\
& \sum_{k=1}^{n} y_{ik} = 1 \quad \text{for } i = 1, \ldots, n \\
& c_l \in \mathcal{C} \quad \text{for } l = 1, \ldots, q \\
& x_{ij} \in \{0, 1\} \quad i = 1, \ldots, n, \quad j = 1, \ldots, m_i \\
& y_{ik} \in \{0, 1\} \quad i = 1, \ldots, n, \quad k = 1, \ldots, n
\end{aligned}$$

where $1\{\}$ is the indicator function defined as follows:

$$1\{\text{condition}\} = \begin{cases} 1 & \text{if condition} = \text{TRUE} \\ 0 & \text{otherwise} \end{cases}$$

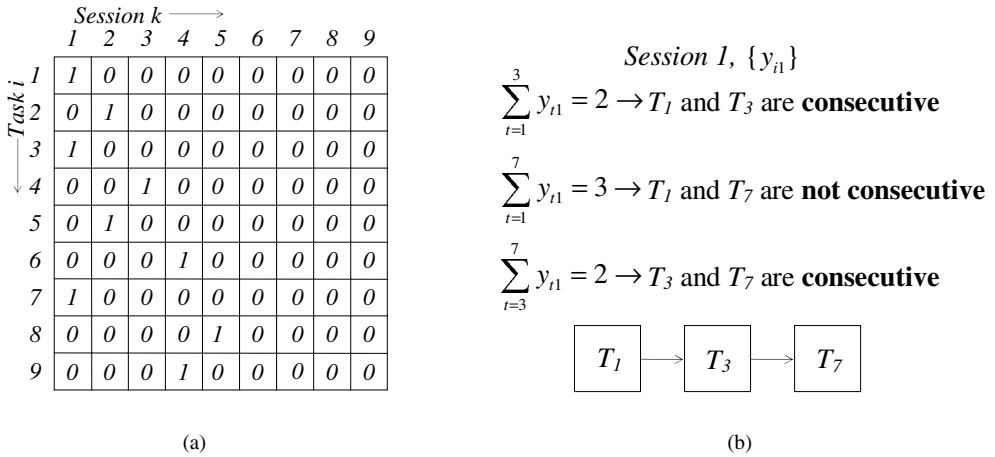
The objective function is composed of two parts. The first part, $F(X)$, is the objective function of the original WMAP, which targets choosing paths with low costs. The second part, $G(X, Y)$, accounts for path sharing in each session; the objective function rewards path overlaps in sessions by subtracting $\alpha G(X, Y)$ from $F(X)$, to give overlapping paths in the same session a boost over non-overlapping paths. $G(X, Y)$ computes the overlap cost in each session and indicator function $1\{\sum_{t=i}^{u} y_{tk} \stackrel{?}{=} 2\}$ ensures that the overlap is computed only for consecutive tasks in similar sessions. Figure 9(a) shows an example session-task assignment, and Figure 9(b) shows the evaluation of the indicator function for session S_1. The multiplier $\alpha > 0$ allows the WFMS to steer the objective of the optimisation problem. For example, assigning high $\alpha > 1$ will generate paths with high overlap but no necessarily path with minimal cost. This, α controls the effect of the overlap on the optimisation problem. The second constraint ensures that each task is assigned to only a single session. All the other constraints are similar to the constraints of WMAP.

Note that the session life cycle for a session S_k can be extracted from the solution vector $y_{ik}, i = 1, \ldots, n$, because the sequence in which the task assigned to session S_k indicated in the vector, for example from Figure 9(a), the solution vector of session S_1, is $[y_{11}, \ldots, y_{91}] = [1, 0, 1, 0, 0, 0, 1, 0, 0]$. Thus, the session life cycle is $\{T_1, T_3, T_7\}$ which is shown in Figure 9(b).

7 Related work

Bertino et al. (1999) proposed a workflow authorisation model capable of specifying and enforcing workflow authorisation constraints, the model assumes a central WFMS having full knowledge of all the users, roles and tasks in the collaboration environment. Altintas et al. (2004) proposed a

Figure 9 Example task session assignment and session life-cycle: (a) task session assignment and (b) session life-cycle



distributed workflow execution model based on the Chinese Wall security model; however, the workflow authorisation is controlled by a central system, which has a global view of the collaboration environment. Similarly, Muth et al. (1998) proposed a WFMS, where the workflow specification is centralised and the workflow execution is distributed.

8 Conclusions

In this paper, we have presented a mediator-free workflow authorisation model where each domain in the collaboration environment encloses its own WFMS. Furthermore, we have presented a task discovery protocol that would enable domains with a limited view of the collaboration environment to discover secure paths to tasks in other domains in the collaboration environment. We have also defined an extended model for workflow authorisation constraints. The model would be able to accommodate constraints involving access paths, tasks and sessions.

Our workflow authorisation problem was formulated as a constraint satisfaction problem that selects paths that satisfy all the authorisation constraints. Specifically, we proposed the WMAP that utilises the minimal authorisations required to execute the workflow tasks, while at the same time satisfying all the workflow authorisation constraints. We showed that by selecting the paths at minimal costs, the WMAP is consistent with the principle of least privilege. Furthermore, we investigated path overlaps among access paths, which allowed us to add path overlaps to our WMAP and formulated the WMAP with overlaps. The WMAPO targeted assigning tasks with overlapping paths to the same sessions and at the same time ensuring minimal paths. Finally, we formulated integer programs to solve the WMAP as well as the WMAPO.

Acknowledgement

The work reported in this paper has been partially supported by the National Science Foundation under the ITR Grant No. 0428554 'The Design and Use of Digital Identities', NSF Grant IIS-0209111, NSF Grant IIS-0242419 and by the sponsors of Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University.

References

Afsarmanesh, H., Garita, C. and Hertzberger, L. (1998) 'Virtual enterprises and federated information sharing', *DEXA'98 : Proceedings of the International Conference Database and Expert Systems Applications*.

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludscher, B. and Mock, S. (2004) 'Kepler: an extensible system for design and execution of scientific workflows', *SSDBM'04 : Proceedings of the 16th International Conference on Scientific and Statistical Database Management*, pp.423–424.

Atluri, V., Chun, S. and Mazzoleni, P. (2001) 'A Chinese wall security model for decentralized workflow systems', *CCS '01: Proceedings of the Eighth ACM Conference on Computer and Communications Security*, ACM Press, pp.48–57.

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

Brewer, D. and Nash, M. (1989) 'The Chinese wall security policy', *Proceedings of IEEE Symposium on Security and Privacy*, pp.206–214.

Clark, D. and Wilson, D. (1987) 'A comparison of commercial and military computer security policies', *Proceedings of IEEE Symposium on Security and Privacy*, pp.184–194.

Crampton, J. (2003) 'On permissions, inheritance and role hierarchies', *CCS '03: Proceedings of the Tenth ACM Conference on Computer and Communications Security*, ACM Press, pp.85–92.

Dechter, R. (2003) *Constraint Processing*, Morgan Kaufmann.

Desai, A. and Awad, N. (2005) 'Special issue on adaptive complex enterprises', *Communications of ACM*, Vol. 48, No. 5.

Ferraiolo, D. and Kuhn, D. and Chandramouli, R. (2003) 'Role-based access control', *Artech House*.

Ferraiolo, D., Sandhu, R., Gavrila, S., Kuhn, D. and Chandramouli, R. (2001) 'Proposed NIST standard for role-based access control', *ACM Transactions on Information and Systems Security*, Vol. 4, No. 3, pp.224–274.

Franklin, M. and Liu, D. (2004) 'The design of GridDB: a data-centric overlay for the scientific grid', *VLDB'04: Proceedings of the 13th International Conference on Very Large Data Bases*.

Gong, L. and Qian, X. (1994) 'The complexity and composability of secure interoperation', *Proceedings of IEEE Symposium on Security and Privacy*, Washington, DC: IEEE Computer Society, pp.190–200.

Gong, L. and Qian, X. (1996) 'Computational issues in secure interoperation', *IEEE Transaction on Software and Engineering.*, Vol. 22, No. 1.

Ibaraki, T. and Katoh, N. (1988) *Resource Allocation Problems: Algorithmic Approaches*, Cambridge, MA: MIT Press.

Li, N., Bizri, Z. and Tripunitara, M. (2004) 'On mutually exclusive roles and separation of duty', *CCS '04: Proceedings of ACM Conference on Computer and Communications Security*.

Ludwig, H., Bussler, C., Shan, M. and Grefen, P. (1999) 'Cross-organisational workflow management and co-ordination WACC', *99 Workshop Report*, Vol. 20, No. 1.

Madhavan, J., Bernstein, P., Doan, A. and Halevy, A. (2005) 'Corpus-based schema matching', *ICDE'2005: Proceedings of the 21st International Conference on Data Engineering*.

Madhavan, J. and Halevy, A. (2003) 'Composing mappings among data sources', *VLDB' 2003 : Proceedings of the 29th International Conference on Very Large Databases*.

Maltz, D., Broch, J., Jetcheva, J. and Johnson, D. (1999) 'The effects of on-demand behavior in routing protocols for multi-hop wireless ad hoc networks', *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp.1439–1453.

Marriott, K. and Stuckey, P. (1998) *Programming with Constraints: An Introduction*, Cambridge, MA: MIT Press.

Muth, P., Wodtke, D., Weissenfels, J., Dittrich, A. and Weikum, G. (1998) 'From centralized workflow specification to distributed workflow execution', *Journal of Intelligent Information Systems*, Vol. 10, No. 2, pp.159–184.

Nemhauser, G. and Wolsey, L. (1988) *Integer and Combinatorial Optimization*, John Wiley and Sons Inc.

Perkins, C. and Royer, E. (1999) 'Ad-hoc on-demand distance vector routing', *WMCSA'99 : Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, pp.90–100.

Ramnath, R. and Landsbergen, D. (2005) 'IT-enabled sense-and-respond strategies in complex public organizations', *Communications of ACM*, Vol. 48, No. 5, pp.58–64.

Saltzer, J. and Schroeder, M. (1975) 'The protection of information in computer systems', *Proceedings of the IEEE*, Vol. 63, No. 9, pp.1278–1308.

Sandhu, R., Coyne, E., Feinstein, H. and Youman, C. (1996) 'Role-based access control models', *IEEE Computer*, Vol. 29, No. 2, pp.38–47.

Shankar, S., Kini, A., DeWitt, D. and Naughton, J. (2005) 'Integrating databases and workflow systems', *SIGMOD Record*, Vol. 34, No. 3, pp.5–11.

Shehab, M., Bertino, E. and Ghafoor, A. (2005a) 'Secure collaboration in mediator-free environments', *CCS '05: Proceedings of the 12th ACM Conference on Computer and Communications Security*, ACM Press.

Shehab, M., Bertino, E. and Ghafoor, A. (2005b) 'SERAT: secure role mapping technique for decentralized secure interoperability', *SACMAT '05: Proceedings of the ACM Symposium on Access Control Models and Technologies*, ACM Press.

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