Task Delegation Based Access Control Models for Workflow Systems

Khaled Gaaloul, François Charoy

Task Delegation Based Access Control Models for Workflow Systems

Khaled Gaaloul[1] and François Charoy[2]

[1] SAP Research
Vincenz-Priessnitz-Strasse 1, 76131 Karlsruhe, Germany
[2] LORIA - INRIA - CNRS - UMR 7503
BP 239, F-54506 Vandœuvre-lès-Nancy Cedex, France
khaled.gaaloul@sap.com, charoy@loria.fr

Abstract. e-Government organisations are facilitated and conducted using workflow management systems. Role-based access control (RBAC) is recognised as an efficient access control model for large organisations. The application of RBAC in workflow systems cannot, however, grant permissions to users dynamically while business processes are being executed. We currently observe a move away from predefined strict workflow modelling towards approaches supporting flexibility on the organisational level. One specific approach is that of task delegation. Task delegation is a mechanism that supports organisational flexibility, and ensures delegation of authority in access control systems. In this paper, we propose a Task-oriented Access Control (TAC) model based on RBAC to address these requirements. We aim to reason about task from organisational perspectives and resources perspectives to analyse and specify authorisation constraints. Moreover, we present a fine grained access control protocol to support delegation based on the TAC model.

Keywords: Workflow, Task, Delegation, Access Control, Authorisation.

1 Introduction

The ongoing prosperity of the "e-" trend such as e-Business, e-Government, and e-Services fosters the ever increasing demand for interactions across organisational boundaries. Electronic government (e-Government) is the civil and political conduct of government, including services provision, using information and communication technologies. The concept of e-Government has been gaining ground from initial isolated to extensive research and applications. The prerequisites for an e-Government enactment strategy are the achievement of a technological interoperability of platforms and a deeper cooperation and security at the organisational level. Those requirements are related with the environment in which the public agencies operate, strictly constrained by norms, regulations, and result-oriented at the same time [1]. Actually, most governmental organisations offer electronic services within a collaborative environment. However, inter-organisational collaboration, especially by means of workflow management systems, is not as widespread.

Currently, we observe a tendency moving away from strict enforcement approaches towards mechanisms supporting exceptions that make it difficult to foresee when modelling a workflow. One specific set of mechanisms ensuring human centric interactions and supporting collaboration cross-organisations is that of task delegation [2].
Security is an essential and integral part of workflow management systems. Protecting application data in workflow systems through access control policies has recently been widely discussed. Sandhu proposed a series of access control models [3, 4]: RBAC0, RBAC1, RBAC2, RBAC3, and discussed a variety of constraints and policies including role hierarchy and separation of duties (SoD). These models are called the RBAC96 models. The central idea of this model is that access rights are associated with roles, to which users are assigned in order to get appropriate authorisations. It also involves the role hierarchy that enables the permission heritage. Since the roles in organisations are relatively stable and the number of roles is much smaller then that of users, the work of administrators can be greatly relieved by applying the concept of roles. Thus it is more adaptable to dynamic environments to a certain extent. However, there is no concept of tasks in RBAC, which makes it difficult to satisfy completely the access control requirements in a rapidly-changing dynamic environment [5, 6].
In this paper, we propose a task-oriented access control model based on RBAC, thereby addressing the authorisation requirements in workflow management systems. Permissions are authorised both to roles and to tasks. The idea is to leverage the RBAC features regarding permissions assignments based-roles. In addition, users can get permissions through tasks when they execute a process, thereby supporting tasks dynamic constraints. Moreover, we offer a fine grained access control solution to compute delegated privileges.
The remainder of this paper is organised as follows. Section 2 presents a workflow scenario inspired from a governmental use case to motivate our work. In section 3, we give an overview about delegation and present the delegation scenarios. Section 4 defines workflow authorisations constraints and presents our access control model based-task. In section 5, we define a delegation protocol supporting delegation of authority. In Section 6, we discuss and conclude our approach, and outline several topics of potential future work.

2 e-Government Workflow Scenario

We introduce a govermental workflow scenario related to the European administrations collaboration. Europol and Eurojust are two key elements of the European system of international collaboration within the areas of law enforcement and justice. A specific scenario for this collaboration is the Mutual Legal Assistance (MLA) [7].

2.1 Mutual Legal Assistance

Mutual Legal Assistance (MLA) defines a collaborative workflow scenario involving national authorities of two European countries regarding the execution

of measures for protection of a witness in a criminal proceeding. Here we describe the MLA process cross Eurojust organisations A and B, and detail the different business actors and resources models involved in the process. Basically, the two organisations work consists of receiving the request of assistance from the Europol member in order to process it and send it the concerned authority in country B. Eurojust infrastructure integrates systems such as MLA service and CMS (Case management System) to process data on the individual cases on which Eurojust national members are working in temporary work files. Eurojust defines an organisational hierarchy working together to achieve common goals. Figure 1 illustrates the organisational role hierarchy and users role memberships in the Eurojust organisation.

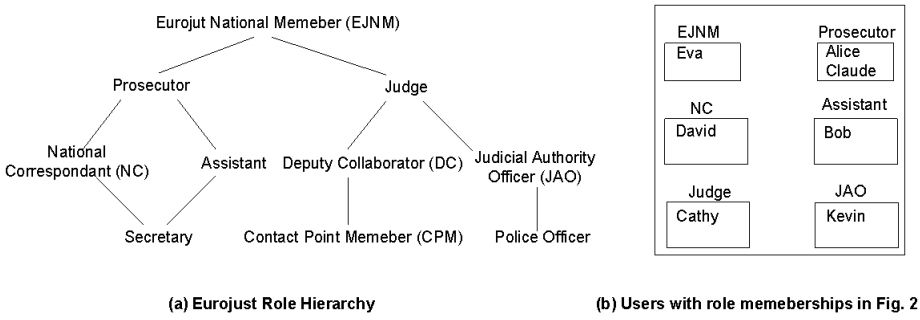

Fig. 1. An example of organisational role hierarchy and users in Eurojust

We applied the Business Process Modeling Notation (BPMN) to the MLA process (see Figure 2). BPMN has emerged as a standard notation for capturing business processes, especially at the level of domain analysis and high-level systems design [8]. The notation inherits and combines elements from a number of other proposed notations for business process modeling, including the XML Process Definition Language (XPDL) and the Activity Diagrams component of the Unified Modeling Language (UML).

In our example, we distinguish Prosecutor as the main responsible that collaborates with internal and external employees (Assistant, National Correspondent (NC), Judge and Judicial Authority Officer (JAO)) to process the MLA request. First, Prosecutor A receives the request and checks it in the MLA information service (tasks 1 and 2). If the provided information are correct, the Prosecutor will continue to process the request by asking for the preparation of the request document (task 4). Note that depending on the request context, the application process will differ in the users involved and data that need to be considered. In fact, the specific type of legal document requested will have a direct effect on the involved controls. For instance, the "Translate Request Document" task might be required to carry out the request preparation when exchanged documents are issued in the local language; therefore we need a national cor-

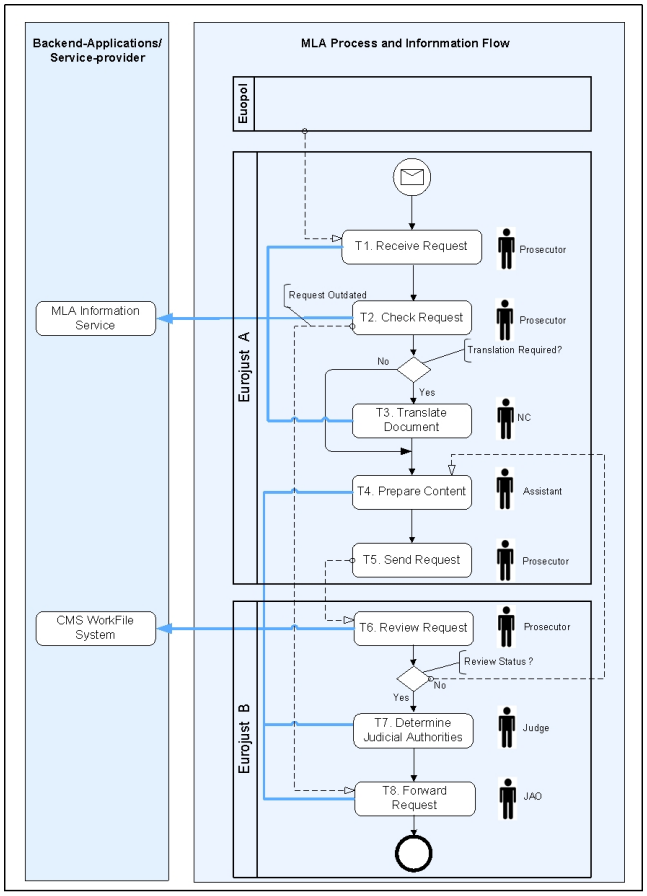

Fig. 2. MLA scenario

respondent (NC) to translate documents (task 3). After the preparation of the required legal documents, the Prosecutor will send the request to his Eurojust colleagues in country B (task 5). The next steps that need to be taken are the review of the request, the determination of the judicial authority in order to forward the request to the concerned authority in country B (Eurojust B) for the final approval (tasks 6, 7 and 8).
The supporting Table 1 summarises the required roles, applications, functions and business objects associated to tasks.

2.2 Problem Statement

Several of the depicted tasks involve human interactions and are possibly time consuming. Tasks taken by the Eurojust organisations can involve several business actors such as Prosecutors, Assistants or Judges. Depending on the current

Table 1. Logistic workflow: Relations between tasks, roles, applications and business objects

Task type	Role	Application	Function	Business Object type
T1. Receive Request	Prosecutor	MLA Information Service	read()	Request Document
T2. Check Request	Prosecutor	MLA Information Service	query(), update()	Request Document
T3. Translate Document	NC	MLA Information Service	translate()	Request Document
T4. Prepare Content	Assistant	CMS WorkFile System	add()	Request Document
T5. Send Request	Prosecutor	CMS WorkFile System	send()	Request File
T6. Review Request	Prosecutor	CMS WorkFile System	read(), update()	Request File
T7. Determine Judicial Authorities	Judge	CMS WorkFile System	add(), modify()	Request File
T8. Forward Request	JAO	CMS WorkFile System	send()	Request File

control-flow sequence, workflow actors can evolve and change from the predefined workflow model. For instance, the absence of translated document (task 3) can lead to a new rearrangement of actors in order to optimise the process execution. In addition, unexpected events can happen without being modelled beforehand. For example, a Prosecutor delegates a part of his work to a subordinate due to emergency situations.

In this scenario, we describe the collaborative work cross-organisations. One of the objectives is to establish a collaboration including information exchange. Those objectives can be achieved using collaborative workflows [9, 10]. However, recent works [11, 9] presented new requirements such as control and transparency in collaborative workflow systems. What is in many cases described as collaboration appears to be coordination and synchronisation of processes by ignoring human-centric interactions. One type of transparency and control supporting mechanism in human-centric collaboration is that of task delegation. Task delegation is a mechanism supporting organisational flexibility in the human-centric workflow systems, and ensuring delegation of authority in access control systems [12].

3 Delegation in Workflow

In this section, we give a brief overview of delegation in workflows systems. We present the main factors that can motivate delegation and link it to the case study.

3.1 Context and Motivations

Task delegation can be very useful for real-world situations where a user who is authorised to perform a task is either unavailable or too overloaded with work to successfully complete it. This can occur, for example, when certain users are sick or on leave. It is frequently the case that delaying these task executions will violate time constraints on the workflow impairing the entire workflow execution. Delegation is a suitable approach to handle such exceptions and to ensure alternative scenarios by making workflow systems flexible and efficient.
Schaad presented a literature review of the different aspects and motivations for delegation [13]. Generally, delegation is motivated by three main factors: organisational, business process management and resources. In the following, we detail specific factors that can motivate delegation:

1. Lack of resources: The task cannot be achieved due to a lack of resources. The user holding the task misses one or several necessary resources. He has to delegate to another user possessing the required medium. Examples for such resources could be a lack of time or equipments.
2. Specialisation: A user might be sufficiently competent to achieve a goal, but it is more efficient to delegate to users in specialist positions, such that the achievement is optimised. Specialisation is a part of the business process management factor.
3. Organisational policies: Goals may conflict and specific organisational policies such as the separation of duties (SoD) may require a user to delegate. SoD constraint defines exclusive relation between tasks. For instance, tasks t_1 and t_2 can not be assigned to the same user. This defines one of the motivation criteria of the organisational factor.

3.2 Link with the Case Study

During the collaboration between Eurojust organisations A and B, several actors are involved in the MLA process (see Figure 2). We define role-based delegation to support human-centric interactions. We are considering a user-to-user delegation supporting role-based access control model (RBAC) defined in [14]. In the following, we present two scenarios describing both local and global delegation.

Definition 1. *We define a task delegation relation* RD = (T,u_1,u_2,DC)*, where* T *is the delegated task,* u_1 *the delegator,* u_2 *the delegatee, and* DC *the delegation constraints. Constraints refer to the condition of delegating accordingly to the global policy.*

Local delegation (DS1): We consider an instance of the process MLA where no intervention is required from the NC member. We denote user *Alice* member of role Prosecutor and user *Bob* member of role Assistant. T5 is assigned to Alice, where Alice needs to send the MLA request to authority B. Alice is overloaded (lack of resources) and need to delegate T5 to one of his assistant.

Delegation criteria is based on the role hierarchy (RH) of Eurojust A, where the Assistant Bob is a subordinate to the Prosecutor Alice based on the global policy definition.
The delegation relation (T5,Alice,Bob,(RH,2days)) $\in$ *RD*, where (RH, 2 days) defines the delegation constraints *DC* regarding the time validity (2 days) and the organisational constraint (RH).

Global delegation (DS2): It defines a delegation cross-organisations. We consider an instance of the process MLA where the MLA request is outdated and exists already in the CMS system of country B. The specialisation of Prosecutor Claude will motivate the prosecutor Alice to delegate T2 for his colleague. Task delegation is defined based on a role mapping (RM) constraint, where distributed resources with external roles are defined in the global policy. The delegation relation (T2,Alice,Claude,RM) $\in$ *RD*.

The next step will be to consider the propagation of authority during delegation. We need to define authorisation requirements with regards to workflow invariants such as task, users and data [6]. To this end, we propose an access control model based on the workflow specifications and user authorisation information (see Section 4.2).

4 Workflow Authorisation Constraints

A workflow comprises various activities that are involved in a business process. Activities that are part of a process are represented as tasks. Authorisation information is given which authorises users to perform tasks. Such authorisation information may be specified using a simple access control list or more complex role-based structures [15].

4.1 Task Execution Model

We define a task execution model using a UML activity diagram composed of three main activities: *Initialisation*, *Processing* and *Finalisation*. During the initialisation of the task, a task instance is created and then assigned to a user. During task processing, the assigned user can start or delegate the task which gathers all operations and rights over the business objects related to task resources. Finally the task finalisation would notice the workflow management system that the task is terminated, where termination defines completeness, failure or cancellation.

Seeing the task as a block that needs protection against undesired accesses, the activity diagram includes an access control (AC) transition that is in charge with granting or not the access to the task. AC checks defines the transition from the creation of a task to its assignment to a user. This assignment will lead to the processing or the cancellation of the task. Cancellation can be triggered when the assigned user doesn't fulfil the required authorisation to proceed the

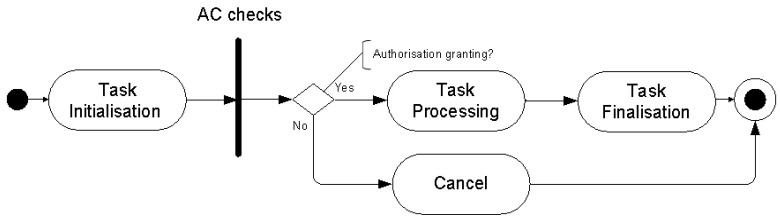

Fig. 3. Task execution model

task instance.
The AC transition defines the on time authorisation supporting task execution. It defines a relationship between user, task instance and authorisation instance. An authorisation instance defines the permission needed to execute operations on business objects to carry out a task.

Definition 2. P *is a set of permissions.* P *defines the right to execute an operation on a resource type. A permission* p *is a pair* (f,o) *where* f *is a function and* o *is a business object:* $p \subseteq f \times o$.

For instance, the task T7 "Determine Judicial Authorities" requires a permission that defines functions *add()* and *modify()* to access the MLA business object (see Table 1). Therefore, the assigned user Cathy member of role Judge needs to be authorised to access T7 task resources.

4.2 Task-Oriented Access Control Model

We propose a Task-oriented Access Control (TAC) model to support authorisation requirements in workflow systems (see Figure 4). Authorisation information will be inferred from access control data structures, such as user-role assignment and task-role assignment relations [16]. We leverage the different task requirements regarding human and material resources and model it in a set of relationships building our model.

Formally, we define sets U, R, OU, T, P, S and TI as a set of users, roles, organisations units, tasks, permissions, subjects and task instances respectively. We define RH (Role Hierarchy), where RH is a partial order on R. $(r_i, r_j) \in R$, RH denotes that r_i is a role superior to r_j, as a result, r_i automatically inherits the permissions of r_j.

We define RM (Role Mapping), where RM is a partial order on R belonging to a set of roles defined in the involved organisations hierarchies (OU). RM defines external roles accessing distributed resources cross-organisations [17]. It provide a decentralised access control mechanism where externally known roles are publicly available, where:

$r_k \in OU_k$ and $r_l \in OU_l$, RM denotes that r_l is a role mapped to r_k, as a result, r_l automatically inherits the permissions of r_k.

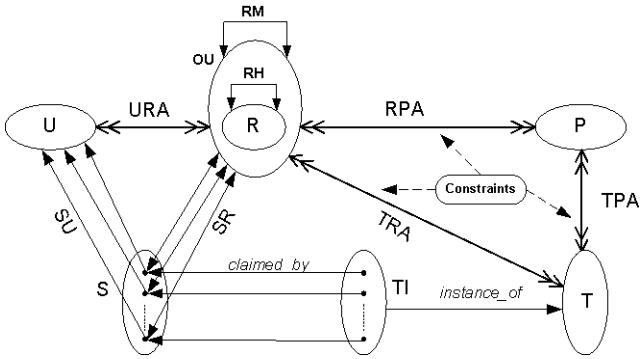

Fig. 4. Task-oriented access control model

Definitions of Map Relations:

- $URA \subseteq U \times R$, the user role assignment relation mapping users to roles they are member of.
- $RPA \subseteq P \times R$, the permission role assignment relation mapping roles to permissions they are authorised to.
- $TPA \subseteq T \times P$, the task permission relation mapping tasks to permissions. This defines the set of permission required to execute a task.
- $TRA \subseteq T \times R$ the task role assignment relation mapping roles to tasks they are assigned to.

Definitions of Functions:

- $SU: S \rightarrow U$ a function mapping a subject to the corresponding user.
- $SR: S \rightarrow 2^R$, a function mapping each subject to a set of roles, where $SR(s_i) \subseteq \{r|(SU(s_i), r) \in URA\}$ and subject s_i has the permissions; $\cup_{\{r \in SR(s_i)\}}\{p|(p, r) \in RPA\}$.
- $instance_of: TI \rightarrow T$, a function mapping a task instance to its task type.
- $claimed_by: TI \rightarrow S$, a function mapping a task instance to a subject to execute it, where:
  $claimed_by(t_i, s_i) = \{t_i|instance_of(t_i, t), (r, u) \in URA|(SR(s_i) = r \bigwedge SU(s_i) = u), (t, r) \in TRA\}$.

Definitions of Constraints:

Here we discuss Separation of duty (SoD) and Binding of duty (BoD) constraints. We define exclusive relation between tasks for SoD, and binding relation

between tasks for BoD as follows:

$TT_{SOD} = \{(t_i, t_j) \in T | t_i \textit{ is Exclusive with } t_j\} \subseteq \text{TxT}$

$TT_{BOD} = \{(t_i, t_j) \in T | t_i \textit{ is Binding with } t_j\} \subseteq \text{TxT}$, where $t_i \leq t_j$.

If $(t_1, t_2) \in TT_{SOD}$, then t_1 and t_2 can not be assigned to the same subject. For instance, T4 and T6 $\in TT_{SOD}$, where subjects with role Prosecutor must be different.
If $(t_1, t_2) \in TT_{BOD}$, then t_1 and t_2 must be assigned to the same subject.

Contributions and Motivations:

We model permission assignment relations for task and role in order to support both human and material resources. The tuple (P,T,R) specifies TRA, TPA and RPA many-to-many relationships which are specifics to the task execution context. The remaining relations are generic relations based on RBAC model [3].

Definition 3. *A task can only be assigned to a role if and only if* $(t, r) \in TRA \Rightarrow \{p \in P | (t, p) \in TPA\} \subset \{p | (p, r) \in RPA\}$.

The main contribution is to specify the task assignment conditions based on the RPA and TPA requirements (see Definition 3). Actually, two conditions have to be verified to satisfy TRA relation. The first condition is related to task resources requirements. The user's permissions defined in RPA need to satisfy the permissions defined in TPA. If this condition is satisfied, the task is executed if and only if the user is assigned to it. Basically, having permissions to execute a task but not being in the task worklist will not satisfy those conditions and, therefore, deny the access to task resources.
Returning to the example, T2 "Check Request" is assigned a set of permissions (*query(), update()*) via TPA in order to carry out this task. Once T2 is claimed, TRA is assigned to roles that are authorised to claim it. On one hand, user Bob with role Assistant is not allowed to claim T2. Bobs permissions (*add()*) do not fulfill T3 requirements. Therefore, the relation for the instance of T2: TRA (T2, Bob) returns false and no authorisation is granted for Bob. On the other hand, user Kevin member of role JAO has the required permissions to carry out T5, however, Kevin does not have the right to claim T2 since the user-task assignments is not defined in the global policy (see Figure 2, Table 1).

5 Context-Aware Delegation for TAC

Delegation is a mechanism that permits a user to assign a subset of his assigned authorisations (privileges) to other users who currently do not possess it. The user who performs a delegation is referred to as a "delegator" and the user who receives a delegation is referred to as a "delegatee". We provide an optimised

method to compute the delegated privileges based on the current requirements of the task instances (resources requirements). The TAC model defines the list of potential delegatees (RPA) that may satisfy the delegated task requirements (TPA). For instance, u_1 and u_2 are members of roles r_1 and r_2 respectively; (t,u_1,u_2,DC) $\in RD$ *iff* $(t, r_2) \in TRA \Rightarrow \{p \in P | (t, p) \in TPA\} \subset \{p | (p, r_2) \in RPA\}$.

5.1 Delegation Protocol

We present a fine grained access control protocol to support delegation. Delegation protocol depicts the dialogue between a delegator and a delegatee during a delegation request. We model the protocol using UML sequence diagrams (see Figure 5).

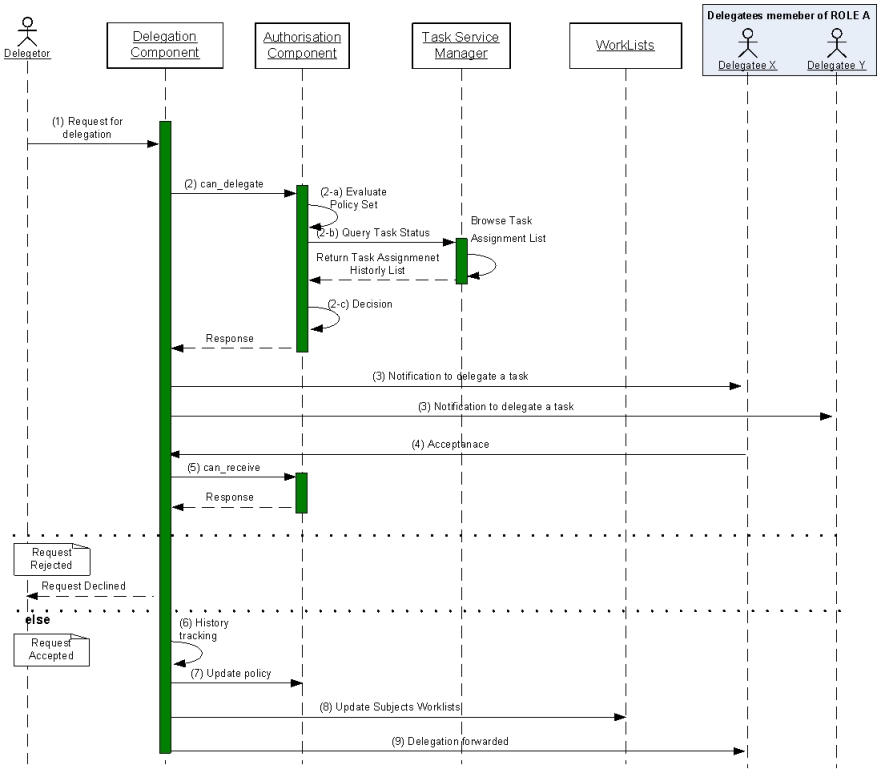

Fig. 5. Task delegation protocol

The Authorisation Component supports the access control mechanism to make a policy decision. An access control policy specifies a level of defining

access to task resources when starting or delegating a task. The Task Service Manager returns the current task state (started, cancelled, etc.). The Worklist component maintains the user-task assignments during runtime. We detail the basic steps as follows:

1. First the delegator is sending a request for delegation to the Delegation Component (DC) for a specific task and a specific role (Role A).
2. The DC checks with the help of the Authorisation Component (AC) if the delegator can actually delegate based on his policy attributes, then with the Task Service Manager regarding the delegated task status.
3. The DC notifies all the delegatees belonging to the role (Role A) of the availability of the task.
4. The first one to respond is allocated with the task.
5. The DC checks with the help of the AC if the delegatee can actually receive the task.
6. The DC then keeps track of the current delegation within internal history records.
7. The DC updates the appropriate policy in the policy repository.
8. The DC updates the appropriate worklists (delegator and delegatee) if the delegation is related to a task instance.
9. Then the delegation is forwarded to the designated delegatee.

Returning to the example, the scenario DS2 can be satisfied. Actually, Prosecutor Alice can delegate T2 "Check Request" to Prosecutor Claude based on their role mapping constraint. In addition, global policy constraints (SoD, BoD) are not specified for T2. Subsequently, Prosecutor Claude will inherit permissions that will authorise him to claim T2 and access its resources afterwards.

5.2 Revocation

Revocation is an important process that must accompany the delegation. It is the subsequent withdrawal of previously delegated objects such as a role or a task. A vast amount of different views on the topic can be found in literature [18] where each author having their own assumptions and opinions on how to model revocation. For simplification, our model of revocation is closely related to the delegation model based user-to-user. Actually, the decision of revocation is issued from the delegator in order to take away the delegated privileges (permissions), or the desire to go back to the state before privileges were delegated.

6 Related Work

Role-based access control (RBAC) is recognised as an efficient access control model for large organisations. Most organisations have some business rules related to access control policy. Delegation of authority is among these rules [19]. In [4], authors extend the RBAC96 model by defining some delegations rules.

Barka and Sandhu proposed a role-based delegation model. They deal with user-to-user delegation. The unit of delegation in them is a role. However, users may want to delegate a piece of permission from a role.

Zhang and *al.* propose a flexible delegation model named Permission-based Delegation Model (PBDM) [20]. PBDM supports user-to-user and role-to-role delegations with features of multi-step delegation and multi-option revocation. It also supports both role and permission level delegation, which provides great flexibility in authority management. However, neither RBAC nor PBDM support the task-based delegation criteria described in the motivated delegation scenarios.

The eXtensible Access Control Markup Language is an XML-based, declarative access control policy language that lets policy editors to specify the rules about who can do what and when. As an OASIS standard, its greatest strength lies in interoperability [21]. Unlike other application-specific, proprietary access-control mechanisms, this standard can be specified once and deployed beyond the boundaries of organisations and countries. The current XACML standard does not provide explicit support for task delegation.

In [22], Rissanen and Firozabadi add new structured data-types to express chains of delegation and constraints on delegation. The main result of their research is an administrative delegation. It is about creating new long-term access control policies by means of delegation in a decentralised organisation. However, this approach does not cover ad-hoc interactions and is not suitable to not support decentralized delegation in the context of heavily human centric collaboration.

7 Conclusion and Future Work

Enormous amounts of data flow cross-organisations along processes and are shared by many different users. Their security must be assured. In this paper, we firstly analyse the relevant authorisation requirements in workflow management systems. Then, based on RBAC model, we propose the task-oriented access control (TAC) model. This model can grant authorisations based on workflow specifications and user authorisation information. The motivation of this direction is inspired from an e-government case study supporting dynamic authorisation changes during delegation. In this context, we proposed a fine grained access control protocol to support delegation based on TAC constraints, thereby ensuring dynamic delegation of authority.

The next stage of our work is the implementation of our approach using the eXtensible Access Control Markup Language (XACML) standard. We propose an extension to XACML specifications supporting task delegation constraints. Future work will look also at enriching our approach with additional delegation constraints supporting historical records. Delegation history will be used to record delegation that have been made to address administrative requirements such as auditing.

References

1. Roland Traunmüller and Maria Wimmer, editors. *e-Government at a Decisive Moment: Sketching a Roadmap to Excellence*, volume 3183 of *Lecture Notes in Computer Science*. Springer, 2004.
2. Andreas Schaad. A framework for evidence lifecycle management. In *Web Information Systems Engineering, Proceedings of the WISE 2007 International Workshops, Nancy, France*, Lecture Notes in Computer Science, pages 191–200. Springer, 2007.
3. Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.
4. E. Barka and R. Sandhu. Framework for role-based delegation models. In *Proceedings of the 16th Annual Computer Security Applications Conference*, pages 168–176, Washington, DC, USA, 2000. IEEE Computer Society.
5. Li Zhang Xu Liao and Stephen Chi fai Chan. A task-oriented access control model for wfms. pages 168–177. Information Security Practice and Experience, First International Conference, ISPEC 2005, Singapore, April 11-14, 2005, Proceedings, Springer, 2005.
6. Khaled Gaaloul, Andreas Schaad, Ulrich Flegel, and Franois Charoy. A secure task delegation model for workflows. In *SECURWARE '08: Proceedings of the 2008 Second International Conference on Emerging Security Information, Systems and Technologies*, pages 10–15, Washington, DC, USA, 2008. IEEE Computer Society.
7. R4eGov Technical Annex 1. Towards e-Administration in the large. Sixth Framework Programme, Information Society Technologies, March 2006. `http://www.r4egov.info`.
8. The Workflow Management Coalition. Process Definition Interface XML Process Definition Language (2005). `http://www.wfmc.org`.
9. Karsten A. Schulz and Maria E. Orlowska. Facilitating cross-organisational workflows with a workflow view approach. *Data Knowl. Eng.*, 51(1):109–147, 2004.
10. Mariangela Contenti, Massimo Mecella, Alessandro Termini, and Roberto Baldoni. A Distributed Architecture for Supporting e-Government Cooperative Processes. In *TCGOV*, pages 181–192, 2005.
11. Chris Jensen and Walt Scacchi. Collaboration, Leadership, Control, and Conflict Negotiation in the NetBeans.org Community. In *26th International Software Engineering Conference*, 2004.
12. Khaled Gaaloul, Franois Charoy, and Andreas Schaad. Modelling Task Delegation for Human-Centric eGovernment Workflows. To appear in the 10th International Digital Government Research Conference (dg.o 2009).
13. Andreas Schaad. A Framework for Organisational Control Principles. PhD thesis, The University of York, York, England, 2003.
14. Longhua Zhang, Gail-Joon Ahn, and Bei-Tseng Chu. A rule-based framework for role-based delegation and revocation. *ACM Transactions on Information and System Security*, 6(3):404–441, 2003.
15. Jason Crampton and Hemanth Khambhammettu. Delegation and satisfiability in workflow systems. In *SACMAT '08: Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 31–40, New York, NY, USA, 2008. ACM.
16. Savith Kandala, Ravi Sandhu, Savith K, Savith K, Ravi S, and Ravi S. Secure role-based workflow models. In *Metal Detection, Volume II, Technical Proposal, FETC Contract DE-AR2195MC32089*, pages 45–58. Kluwer, 2002.

17. Eric Freudenthal, Tracy Pesin, Lawrence Port, Edward Keenan, and Vijay Karamcheti. drbac: Distributed role-based access control for dynamic coalition environments. In *ICDCS '02: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, page 411, Washington, DC, USA, 2002. IEEE Computer Society.
18. Asa Hagstrom, Sushil Jajodia, Francesco Parisi-Presicce, and Duminda Wijesekera. Revocations-A Classification. In *CSFW '01: Proceedings of the 14th IEEE workshop on Computer Security Foundations*, page 44, Washington, DC, USA, 2001. IEEE Computer Society.
19. András Belokosztolszki, David M. Eyers, and Ken Moody. Policy Contexts: Controlling Information Flow in Parameterised RBAC. In *POLICY '03: Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks*, page 99, Washington, DC, USA, 2003. IEEE Computer Society.
20. Xinwen Zhang, Sejong Oh, and Ravi Sandhu. PBDM: a flexible delegation model in RBAC. In *SACMAT '03: Proceedings of the eighth ACM symposium on Access control models and technologies*, pages 149–157, New York, NY, USA, 2003. ACM Press.
21. eXtensible Access Control Markup Language (XACML v2.0). Standard, Organization for the Advancement of Structured Information Standards (OASIS), February 2005. `http://docs.oasis-open.org/xacml/2.0/access-control-xacml-2.0-core-spec-os.pdf`.
22. Erik Rissanen and Babak Sadighi Firozabadi. Administrative Delegation in XACML. Swedish Institute of Computer Science, Kista-Sweden.