Modelling Task Delegation for Human-Centric eGovernment Workflows

Khaled Gaaloul, François Charoy, Andreas Schaad

Modelling Task Delegation for Human-Centric eGovernment Workflows

Khaled Gaaloul
SAP CEC Karlsruhe, Security & Trust Group
Vincenz-Priessnitz-Strasse 1, 76131 Karlsruhe, Germany
khaled.gaaloul@sap.com

François Charoy
LORIA - INRIA - CNRS:UMR7503
BP 239, 54506 Vandœuvre-lès-Nancy, France
charoy@loria.fr

Andreas Schaad
SAP CEC Karlsruhe, Security & Trust Group
Vincenz-Priessnitz-Strasse 1, 76131 Karlsruhe, Germany
andreas.schaad@sap.com

ABSTRACT

The execution of cross-domain eGovernment applications is a challenging topic. eGovernment organisations are facilitated and conducted using workflow management systems. Workflows automates the management and coordination of organisational or business processes. In the context of eGovernment, what is in many cases described as collaboration appears, however, to be coordination and synchronisation of processes ignoring human-centric interactions. In addition, we observed a tendency, moving away from strict enforcement approaches towards mechanisms supporting exceptions that are difficult to foresee when modelling a workflow. One specific set of mechanisms ensuring human centric interactions and supporting collaboration cross-organisations is task delegation.

In this paper we aim to model task delegation for human-centric eGovernment workflows. We do believe that delegation is a solution to support organisational flexibility in human-centric workflow systems, and ensure delegation of authority in access control systems. First, we present a real case study to motivate delegation. Based on this, we define a task delegation model, and analyse its main criteria and requirements within a workflow. In particular, we will focus on the concept of delegation in eGovernment workflows and link it to our case study. Finally, we present delegation protocols to ensure delegation of authority in access control systems based on workflow authorisation constraints.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Workflow management, Miscellaneous

General Terms

Management, Security

Keywords

eGovernment, Workflow, Task, Delegation, Access Control

1. INTRODUCTION

Electronic government (eGovernment) is the civil and political conduct of government, including services provision, using information and communication technologies. The concept of eGovernment 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 [23]. 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.

The R4eGov project consists of inter-organisational collaboration between European administrations [14]. An example domain for such collaboration is Europol[1] (European Police Office) and Eurojust[2] (European Judicial Cooperation Unit). It describes an interagency collaboration within the areas of law enforcement and justice. One of the objectives is to establish a collaboration, including information exchange between both parties based on legal constraints, such as European laws, to which they have to comply to, but sustain effective degrees of freedom for each department to solve their issues in the way they think is the most efficient and effective [1]. Those objectives can be achieved using collaborative workflow systems [19, 4].

Workflow management systems automates the management and coordination of organisational or business processes. In the context of eGovernment, what is in many cases described as collaboration appears, however, to be coordination and synchronisation of processes by ignoring human-centric interactions. In addition, we observed 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 [17].

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 management systems flexible and efficient.

The concept of delegation has not yet been treated in sufficient details in the context of heavily human-centric col-

[1] http://www.europol.europa.eu/
[2] http://www.eurojust.europa.eu/

laborative workflows [7], it is the subject of the paper to support heavily human-centric collaborative workflows according to global policies specified in the European law regulations constraints. Subsequently, we need to define a task delegation model to support organisational flexibility in the human-centric workflow systems, and ensure delegation of authority in access control systems.
The contributions of this paper are threefold. First, we present a real world scenario supporting delegation and illustrate the working of task delegation in human-centric workflows cross European organisations in R4eGov. We then define a task delegation model within a workflow, and analyse additional requirements to support secure task delegation during workflow execution. To this end, we detail delegation protocols with a specific focus on the involved users to ensure delegation of authority in access control systems based on workflow authorisation constraints. We finally define additional requirements for delegation such as revocation and propose a lightweight implementation of our approach.
The remainder of this paper is organised as follows. Section 2 presents a workflow scenario inspired from an R4eGov use case to motivate delegation scenarios. In section 3, we define task delegation model and analyse its main requirements. Section 4 defines workflow authorisations constraints to ensure secure delegation. In section 5, we detail delegation protocols and present revocation as an additional requirement for delegation. Section 6 presents related work. In Section 7, we discuss and conclude our approach, and outline several topics of potential future work.

2. EGOVERNMENTAL WORKFLOW SCENARIO

We introduce an R4eGov 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. They carry out very specific tasks in the context of dialogues, mutual assistance, joint efforts and cooperation between the police, customs, immigration services and justice departments of the EU member states [1]. During their collaboration, Eurojust and Europol are involved and a number of legal instruments are used. A Specific scenario for this collaboration is the Mutual Legal Assistance (MLA)[3].

2.1 Mutual Legal Assistance

Mutual Legal Assistance (MLA) defines a process involving two national authorities of different European countries regarding the execution of measures for protection of a witness in a criminal proceeding. Here we describe MLA process in the 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.

We applied the the Business Process Modeling Notation (BPMN) notation to our MLA process illustrated in Figure 1. BPMN has emerged as a standard notation for capturing business processes, especially at the level of domain analysis and high-level systems design. 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 Notation (UML) [21].

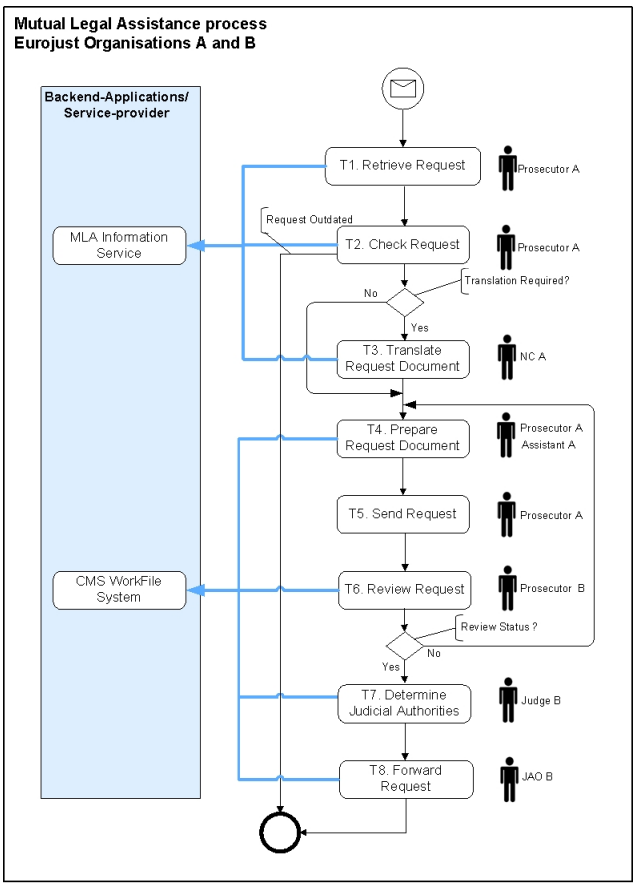

Figure 1: Mutual Legal Assistance scenario

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 users. For instance, the "Trans-

[3]This case study has been performed in joint collaboration between Eurpol, Eurojust and Unisys in the context of the EU FP6 IST Integrated Project R4eGov.

late 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 correspondent 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 define the collaboration inter-organisations Eurojust A and B. Activities 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 for the final approval (tasks 6, 7 and 8).

2.2 Problem Statement

Several of the depicted tasks involve human interactions and are possibly time consuming. Tasks taken by the Eurojust organisation A to define the requested assistance might involve several business actors such as a Prosecutor, an Assistant and a National Correspondent. Depending on the current 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 work of Eurojust collaboration cross organisations. One of the objectives is to establish a collaboration including information exchange between both parties. Those objectives can be achieved using collaborative workflows [19, 4]. However, recent works [12, 24] presented new requirements such as control and transparency in collaborative workflows. 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. In the next section, we motivate the concept of delegation as a support for transparency and control within a human-centric collaborative workflow.

3. DELEGATION IN WORKFLOWS

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 our case study. We then define our delegation model that will monitor the delegation control flow within a workflow.

3.1 Context and Motivations

Within workflow management systems we observed 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 is that of task delegation. In recent work we presented task delegation as a mechanism supporting organisational flexibility in the human-centric workflow systems, and ensuring delegation of authority in access control systems [18, 8].
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 [18]. 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 resource. 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.

In the next section, we will identify scenarios taken from MLA case study to illustrate the requirements and criteria for each delegation scenario defined.

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 Fig.1). Users role memberships are identified to the system as having one or more roles. We define role-based delegation to support human-centric interactions. We are considering a user-to-user delegation where task execution is atomic and delegation is fully assigned to the delegatee with his defined permissions. Therefore, delegation criteria such as cascaded and/or partial are not considered in this paper [25]. In the following, we present two scenarios describing local and global delegation.

Local delegation (DS1): Our interest is related to the Eurojust organisation A. The involved actors are responsible for the reception of the request of assistance and the preparation of the required legal document to be sent to the concerned authority in country B. The role Prosecutor is a senior role and has at his disposal subordinates such as NC and Assistant.

Definition 1. We define a task delegation relation *RD* including the delegated task T, the delegator u_1, the delegatee

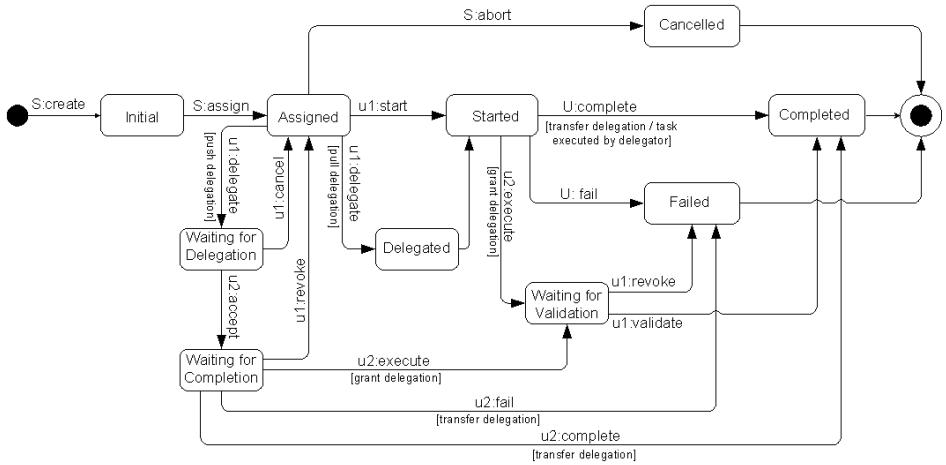

Figure 2: Task delegation model

u_2, the role R and constraints C where constraints refer to the right of delegating accordingly to the global policy.

We consider an instance of the process MLA where no intervention is required from the NC member. We denote Prosecutor A by user *Alice* member of role Prosecutor and Assistant A by user *Bob* member of role Assistant. The preparation and the forwarding of the request defined in T4 and T5 are assigned to Alice. Alice is overloaded (lack of resources) and need to delegate T4 to one of his assistant. Delegation criteria is based on the role hierarchy (RH) of Eurojust, where the assistant Bob is a subordinate to the prosecutor Alice based on the global policy definition. The delegation relation *RD = (T4,Alice,Bob,Assistant,RH)*.

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

3.3 Task Delegation Model (TDM)

In recent work we defined a task delegation model (TDM) based on task life cycle specifications in the workflow management coalition [22, 8]. Figure 2 depicts a UML state diagram that illustrates the life cycle of our TDM in the form of a state transition diagram from the time that a task is created through to final completion, cancellation or failure. It can be seen that there are a series of potential states that comprise this process.

TDM defines the different states and transitions a task can take in a workflow. A task, once created, is generally assigned to a user. The assigned user can choose to start it immediately or to delegate it. Delegation depends on the assignment transition, where the assigned user has the authority to delegate the task to a delegatee in order to act on his behalf. Moreover, we enriched the model with intermediate states supporting delegation features such as:

- **Delegation mode**: It defines how delegation request is issued. Pull mode assumes that a delegator has at his disposal a pool of delegatees to be selected to work on his behalf. Pull mode assumes that a delegator is waiting for an acceptance from a potential delegatee. Derived transitions from push mode are *accept*, *cancel* and *revoke*.
- **Delegation kinds**: It may be classified into grant or transfer [5]. A grant delegation model allows an instantiated task to be available for both delegator and delegatee. As such, the delegator is still having the control to *validate* or *revoke* the task, and the delegatee to execute it. However, in transfer delegation models, the assigned task is transferred to the delegatee worklist. The is no validation required and the task is terminated by the delegatee. Transfer delegation is used to support administrative delegation.

Intermediate states define controlled delegation within a workflow. For instance, the delegator might want to cancel. Our TDM would then go back to the state in which it was before (*Assigned* state). The delegation control flow behaviour remains internal according to the task model, where *Completed*, *Cancelled* and *Failed* are the final states.
Note that each edge within the TDM is prefixed with either an S or an U indicating that the transition is initiated by the workflow system or the human resource respectively. We define u_1 and u_2 belonging to the set of users U, where u_1 is the delegator and u_2 the delegatee.

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 [6].

4.1 Task Execution Model

We define a task execution model based on the TDM state diagram. We define a UML activity diagram including 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 start/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 (see Fig.2).
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 task instance.

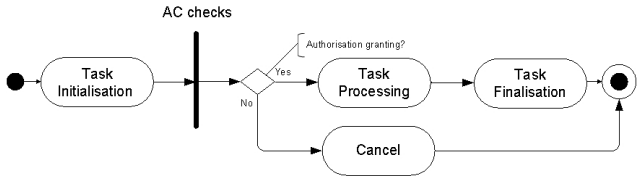

Figure 3: Task execution model

4.2 Authorisation Constraints

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. For instance, the task T4 "Prepare Request Document" requires *read()* and *write()* functions on the MLA request object as material resources. Therefore, the assigned user "Prosecutor A" need to be authorised to access T4 task resources.

Definition 2. A workflow authorisation schema is a pair (T,U), where $A \subseteq T \times U$; u is authorised to perform (or execute) t iff $(t, u) \in A$.

As related in the literature review, the workflow is made of tasks, where a task defines a unit of work that at each invocation performs the binding between different resources needed to complete a specific part of the workflow [16]. The resources that may be involved are different. We distinguish material and human resources for business objects and workflow actors (users), respectively.
We aim by this AC specifications to motivate to two sides of task requirements, namely material and human resources. Once the resources are identified an access control has to be defined to check the authorisation of the initiated user that we call it *subject*. An authorisation makes the explicit binding between a subject, a task resource (object) and his rights over it (action). An access control policy specifies a level of defining access to task resources. Its pseudo formal expression is:

```
<Authorisation>
<Subject>[role]
<Resource>[object]
<Action>[operation]
<Task>[task type]
< /Authorisation>
```

4.3 General Control Process

We illustrate the message flow between access controls components involved during task execution. We present a UML sequence diagram that illustrates whenever a subject claims a task instance to perform it. Basically, a task assignment can be defined in the worklist by the workflow system [22]. In addition, a subject can claim a task access request without being initially assigned, thereby involving dynamic checks of his authorisation credentials.
Whenever a subject issues a claim request to perform a task that is that is protected by the control components. All requests are intercepted by an authorisation enforcement point. The authorisation point is not capable of making an access decision on its own. Therefore, the authorisation enforcement point will request a decision from the authorisation decision point. To make a decision, the decision point queries the policy manager for all policies that are affected to this request. Thus, all policies that apply to the identity of the subject, his role, and policies related to the requested task instance in the corresponding workflow are prompted from the policy manager. The policy manager will browse through its policy repository and returns the set of affected policies to the decision point.
Some of these policies may contain dynamic constraints depending on the current task state and the process history. To get this information the decision point will ask the Task Service Manager (TSM) for the current process state. The TSM retrieves all relevant state information and returns it to the authorisation decision point. Now the decision point is able to evaluate the static and dynamic policies. Depending on the used rule base (e.g. Deny, Overrides, Permit) the binary authorisation decision is returned, i.e. access denied or access granted. In the case the access request is denied, the enforcement point will mediate this decision to the subject, for instance as some error message. In the case access was granted. The intercepted request is forwarded to the TSM hosting all task instances. The TSM will initialise the task and passes the original request to it. In case that the claimed task is not in the subject worklist, the TSM will update the worklist and the task will be performed and will send back potential results to the subject.

We made some assumptions and simplifications in the task assignment process. For instance, we assume the subject's identity and his role attribute are already known in the control architecture. Otherwise the subject has to identify himself against some trusted authority or authentication mechanisms. The first one could issue some kind of identity cer-

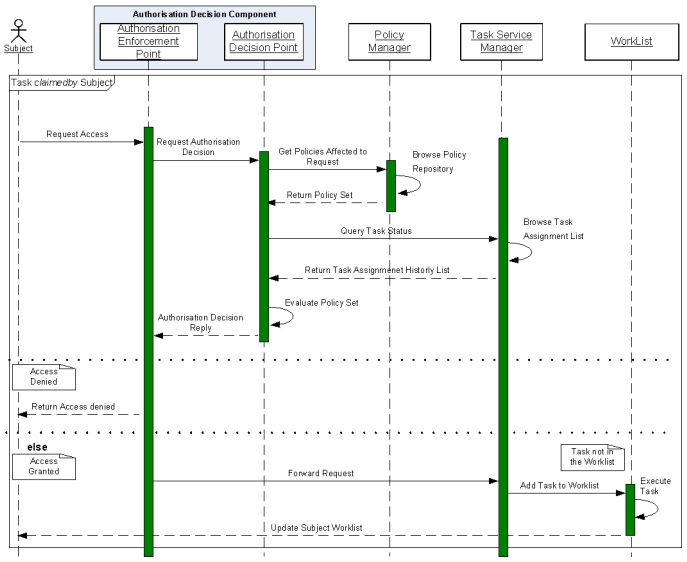

Figure 4: Task assignment sequence diagram

tificate that can be verified by the control architecture, the later one would request the actor's password before the access request will be accepted by the authorisation enforcement component.
As a simplification we omitted the logging of audit data performed by the TSM while a task is requested, performed, and finished. Nevertheless, this information is mandatory with respect to the enforcement of dynamic separation of duty control, such as operational or history-based separation of duty. This point will be discussed in the next section.

5. SECURING TASK DELEGATION

In this section, we introduce delegation protocols to support both *push* and *pull* mode. Delegation protocols define two different models that depict the dialogue between a delegator and a delegatee during a secure task delegation. We model the different protocols using UML sequence diagrams. Delegation protocols will ensure delegation of authority in access control systems.

5.1 Pull Delegation (TDM1)

The pull delegation model (TDM1) is based on a direct allocation of the task through a delegation without any notion of role. This model associates implicitly an authorisation to a subject. When a subject holding a task initiates a delegation process, then the following procedure manages it:

1. First the delegator is sending a request for delegation to the Delegation Component (DC) for a specific task and a specific subject (the delegatee).
2. The DC checks with the help of the Authorisation Component (AC) if the delegator can actually delegate and the delegatee can receive.
   - a) The AC first retrieves the attributes affecting the policy and conducts an initial evaluation regarding the delegator's right to delegate. This is due to the fact that certain task assignments are exclusive and are not allowed to be delegated. In the context of an access control policy, it is defined as an obligation to a rule effect.
   - b) The AC checks then the task status with the Task Service Manager (TSM) component which browse the current task assignment list to check the availability of the task.
   - c) The AC receives the history list from TSM. Finally, the AC sends a response to the DC based on the intermediate results received.
3. The Delegation Component then keeps track of the current delegation within internal history records.
4. DC updates the appropriate policy in the policy repository.
5. DC updates the appropriate worklists (delegator and delegatee's) if the delegation is related to a task instance.
6. Then the delegation request is forwarded to the designated delegatee.

In case of transfer delegation, the given authorisation from the delegator's set are removed from the policy repository.

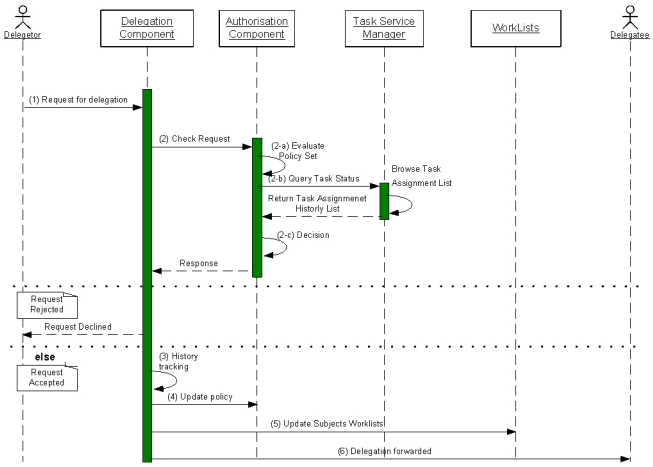

Figure 5: Task delegation pull model

5.2 Push Delegation (TDM2)

The model TDM2 is based on an allocation of the task through a delegation to a role and not directly to a subject. When a subject holding a task initiates a delegation process, then in the TDM2 the following procedure manages it:

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 subjects 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's) if the delegation is related to a task instance.
9. Then the delegation is forwarded to the designated delegatee.

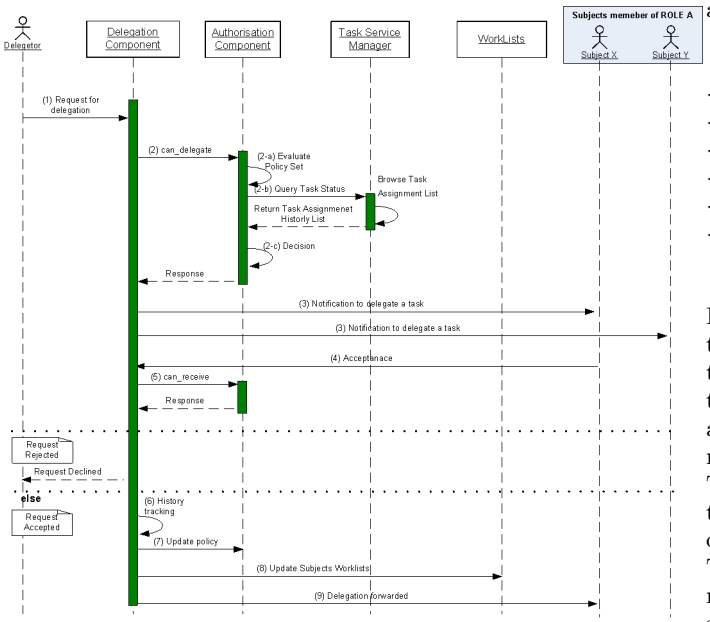

Figure 6: Task delegation push model

5.3 AC enforcement

The access control model handles normally a delegated task and does not need to be modified. The expression of the associated authorisation for delegation is updated in the existing policy. This way provides several advantages:

- In our TDM1 model, the delegatee is not granted with a new role. The delegation process does not need further control of the permission; the access control model handles normally a delegated task and does not need to be modified. The delegatee inherits his own permissions.
- In our TDM2 model, the link of the authorisation with the role is kept. It allows us to reuse our established access control model based-role.

Based on our models and the case study, we exemplify here our work. The TDM1 model delegation answers the requirements defined in the first delegation scenario (DS1). User Bob exists in the delegatees list of user Alice and is directly assigned to the task T4. The formal expression of TDM1 is:

```
<Delegation>
<Issuer>[Subject:Alice]
<Receiver>[Subject:Bob]
<Task>[T4]x[TaskId]
<Authorisation>[Authorisation Instance]
< /Delegation>
```

The TDM2 model delegation answers the requirements defined in the first delegation scenario (DS2). User Alice need to delegate based on role equivalence. User Claude with role Prosecutor is one of the potential delegatees that may accept this delegation request. User Claude is the first to accept the request. The formal expression of TDM2 is:

```
<Delegation>
<Issuer>[Subject:Alice]
<Receiver>[Subject:Claude]x[Role:Prosecutor]
<Task>[T4]x[TaskId]
<Authorisation>[Authorisation Instance]
< /Delegation>
```

If the request is granted by the delegation component then this delegation message will be forwarded to the delegatee that will include the given authorisation to its own set. From this moment on, the delegatee can claim achieving the task as he is provided with the same access rights functions (permissions) as the previous owner of the task (the delegator). The authorisation instance is explicitly related to the role of the first owner of the authorisation to execute an instance of the task (e.g. task T4).
The structure of the delegation and the granularity of our model allow different delegation. A new dimension to classify the delegation can be now characterised by the degree of delegated power. This parameter evaluates how important is the quantity of responsibilities delegated for a task. Additional specifications regarding the authorisation attributes may be added to support different types of delegation [18].

5.4 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 [9]. For simplification, our model of revocation is closely related to the task delegation user-to-user model. Actually, the decision of revocation is issued from the delegator in order to revoke the delegated task, or to cancel the current work. Revoking a task defines an access control enforcement on the current authorisation policy. The delegator has to revoke privileges (permissions) granted to the delegatee in order to complete his task. Revocation can ne triggered automatically based on time constraint, where the delegated privileges are limited in time. Delegator has also the right to

request revocation manually. For instance, delegator want to retrieve his task while the revocation time constraint is not consumed yet. To this end, a proactive access control system is required to support delegation based on dynamic states changes. This is a part of our future work.

5.5 Deploying Task Delegation

The deployment of delegation scenarios in a workflow system requires the development of several components which support TDM specifications. On one hand, we need a workflow system capable of defining and synchronising the delegation process within the workflow process in itself. On the other hand, we need to check the delegatee credentials to authorise task delegation.

Intalio Tempo is a set of runtime components that support human workflow within a service-oriented architecture (SOA). The main goal is to provide a complete and extensible workflow solution with a bias towards interoperable technologies (BPEL, BPEL4People, XForms, REST, and web services) as a default implementation [11].

In this context, we developed a delegation framework to deploy our task delegation model. We implemented the TDM by extending the task object model defined in Intalio Tempo in order to support delegation states. Subsequently, additional transitions such as *delegate* and *revoke* are extended to the defined transitions in the task object model. We also leveraged the task list web-based user interface for users to monitor the task delegation life cycle. The idea is to monitor the delegation control flow based on the triggered transitions and to be able to cope with the unexpected events such as the cancellation of a delegation. Intalio Tempo seems to offer a suitable solution to support our TDM requirements, however, the proposed security framework do not fulfill all the authorisation requirements for delegation and will need further investigations. Actually, the proposed approach based on RBAC model (Role-based access control) remains basic and does not cover the delegation of authority aspect [15].

6. RELATED WORK

WebSphere MQ Workflow provides a richer model that allows users to be described in a broader organisational context. It also supports roles and there can be a many correspondence between users and roles [10]. Staffware has a relatively simple model that denotes users (i.e. individual resources), groups and roles, and allows work to be assigned on the basis of these groupings [20]. IBMs WebSphere MQ Workflow and Staffware both allow a task to be delegated to any user, regardless of his access rights. However, this does not necessarily permit the delegatee to perform the task. For instance, suppose a task T2, in our use case, that involves fetching some data from the CMS system. Suppose Bob does not have the access rights for fetching this data. Delegating T2 to Bob will not allow him to perform it since Bob credentials does not fulfill the authorisation requirements related to task T2.

Role-based access control (RBAC) is recognised as an efficient access control model for large organisations. In [2], authors extend the RBAC96 model by defining some delegation rules. Barka and Sandhu proposed a role-based delegation model. They deal with user-to-user delegation. However, proposed approaches do not support task delegation requirements described in the MLA case study delegation scenarios.

The eXtensible Access Control Markup Language (XACML) 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 [13]. Unlike other application-specific, proprietary access-control mechanisms, this standard can be specified once and deployed beyond the boundaries of organisations and countries. In [15], 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 do not support task delegation constraints.

Chadwick et al. [3] proposed a solution based on the XACML conceptual and data flow models supporting dynamic delegation of authority. The proposed architecture offers a flexible and dynamic way to manage user credentials and assert them in the remote credential providers, however this is not enough to support dynamic delegation of authority. We do believe that delegating a task requires more effort and involves additional specifications related to task delegation states. Task delegation model based-states provides the means to determine faithfully delegation polices proactively, thereby ensuring reactive policy decisions when states such as revoked and cancelled are triggered during task delegation process.

7. CONCLUSION AND FUTURE DIRECTIONS

The execution of cross-domain eGovernment processes need to support more sophisticated interactions of monitoring. We define this as requirements to support human-centric collaboration for inter-organisational workflows. Collaboration must be done in a controlled and transparent way. In this paper, we propose a delegation based mechanism to support and control such kind of interactions between governmental organisations. Our primary concern was to define an approach to support organisational flexibility and to ensure delegation of authority in access control systems. To satisfy this need, we first proposed an extended task model supporting delegation. Our model defines the delegation control flow within a process. We then analysed task delegation requirements with regards to workflow invariants such as task, user and resources. In this context, we proposed an extension to an open source workflow framework to support task delegation. Finally, we detailed delegation protocols to ensure delegation of authority in access control systems based on workflow authorisation constraints.

We consider this paper as a primer for future related work in the areas of collaboration and security. The next stage is the integration of an access control systems within our framework to fulfil the delegation of authority requirements. Moreover, we plan to further investigate the area of compliancy accordingly to workflows global policies specifications.

8. REFERENCES

[1] Eurojust / Europol collaboration, 2006. SIXTH FRAMEWORK PROGRAMME, Information Society Technologies, R4eGov.

[2] 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.

[3] D. W. Chadwick, S. Otenko, and T.-A. Nguyen. Adding support to xacml for dynamic delegation of authority in multiple domains. In *Communications and Multimedia Security, 10th IFIP TC-6 TC-11 International Conference, CMS 2006, Heraklion, Crete, Greece, October 19-21, 2006, Proceedings*, pages 67–86, 2006.

[4] M. Contenti, M. Mecella, A. Termini, and R. Baldoni. A Distributed Architecture for Supporting e-Government Cooperative Processes. In *TCGOV*, pages 181–192, 2005.

[5] J. Crampton and H. Khambhammettu. Delegation in role-based access control. In *Proceedings of the Computer Security - ESORICS 2006, 11th European Symposium on Research in Computer Security, Hamburg, Germany, September 18-20, 2006*, Lecture Notes in Computer Science, pages 174–191. Springer, 2006.

[6] J. Crampton and H. 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.

[7] K. Gaaloul, F. Charoy, A. Schaad, and H. Lee. Collaboration for human-centric egovernment workflows. In *Web Information Systems Engineering, Proceedings of the WISE 2007 International Workshops, Nancy, France*, Lecture Notes in Computer Science, pages 201–212. Springer, 2007.

[8] K. Gaaloul, A. Schaad, U. Flegel, and F. 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.

[9] A. Hagstrom, S. Jajodia, F. Parisi-Presicce, and D. 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.

[10] IBM. IBM Websphere MQ Workflow Getting Started with Buildtime. Version 3.4. IBM Corp., 2003.

[11] Intalio Tempo. Intalio Tempo Architecture . `http://www.tempo.intalio.org/`.

[12] C. Jensen and W. Scacchi. Collaboration, Leadership, Control, and Conflict Negotiation in the NetBeans.org Community. In *26th International Software Engineering Conference*, 2004.

[13] E. T. Moses. eXtensible Access Control Markup Language (XACML) Version 2.0, OASIS. Last viewed on Mar. 28, 2007.

[14] R4eGov Technical Annex 1. Towards e-Administration in the large. SIXTH FRAMEWORK PROGRAMME, Information Society Technologies, March 2006, note = `http://www.r4egov.info`,.

[15] E. Rissanen and B. S. Firozabadi. Administrative Delegation in XACML. Swedish Institute of Computer Science, Kista-Sweden.

[16] N. Russell, W. M. P. van der Aalst, A. H. M. ter Hofstede, and D. Edmond. Workflow resource patterns: Identification, representation and tool support. In *Proceedings of the Advanced Information Systems Engineering, 17th International Conference, CAiSE 2005, Porto, Portugal*, pages 216–232, 2005.

[17] A. 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.

[18] A. Schaad. A Framework for Organisational Control Principles. PhD thesis, The University of York, York, England, 2003.

[19] K. A. Schulz and M. E. Orlowska. Facilitating cross-organisational workflows with a workflow view approach. *Data Knowl. Eng.*, 51(1):109–147, 2004.

[20] Staffware. Staffware Process Suite Defining Staffware Procedures Issue 2. Staffware plc, Maidenhead, 2002.

[21] The Workflow Management Coalition. Process Definition Interface XML Process Definition Language (2005). `http://www.wfmc.org`.

[22] The Workflow Management Coalition. Workflow Management Coalition Terminology and Glossary. Document Number WFMC-TC-1011, February 1999.

[23] R. Traunmüller and M. Wimmer, editors. *e-Government at a Decisive Moment: Sketching a Roadmap to Excellence*, volume 3183 of *Lecture Notes in Computer Science*. Springer, 2004.

[24] C. Wolter, H. Plate, and C. Herbert. Collaborative Workflow Management for eGovernment, September 2007. Accepted in the 1st international workshop on Enterprise Information Systems Engineering (WEISE).

[25] L. Zhang, G.-J. Ahn, and B.-T. Chu. A rule-based framework for role-based delegation and revocation. *ACM Transactions on Information and System Security*, 6(3):404–441, 2003.