On Delegation and Workflow Execution Models

Jason Crampton  Hemanth Khambhammettu
Information Security Group
Royal Holloway, University of London

SAC 2008
Delegation is of increasing interest in workflow management systems (WfMSs)

- There are a number of execution models for computerized workflows
- There are a number of different types of delegation
Delegation is of increasing interest in workflow management systems (WfMSs)

- There are a number of execution models for computerized workflows
- There are a number of different types of delegation

How do these various factors affect delegation in WfMSs?
A workflow specification is a partially ordered set of (abstract) tasks.

- A WfMS instantiates the workflow specification to create a workflow instance.
A **workflow specification** is a partially ordered set of (abstract) tasks

- A WfMS instantiates the workflow specification to create a **workflow instance**

Authorization information determines which users can perform which tasks

- Typically specified using access control lists or role-based access control data structures
- Users are authorized for abstract tasks
- A user may execute a **concrete task** in a given workflow instance if she is authorized for the corresponding abstract task
Workflow Execution Models

WfMS-driven execution model (WDEM)

- WfMS assigns tasks to users on basis of authorization information
- User performs the task that has been assigned

User-driven execution model (UDEM)

- Users initiate (access) requests to perform tasks
- A reference monitor decides whether the user is authorized to perform the task
Delegation of Access Rights

- A user may delegate an access right $r$ for which he is authorized
- Delegation of access rights may take two forms: grant and transfer

<table>
<thead>
<tr>
<th>Delegation operation</th>
<th>Delegatee is authorized for $r$</th>
<th>Delegator is authorized for $r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grant</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Transfer</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>
Delegation in Workflow Systems

Abstract task delegation authorizes the delegatee to perform the delegated task in any workflow instance.
Delegation in Workflow Systems

- **Abstract task** delegation authorizes the delegatee to perform the delegated task in any workflow instance.
- **Concrete task** delegation authorizes the delegatee to perform the delegated task only in the specified workflow instance.
Problem Statement and Motivation

- What does delegation mean in a workflow system?
- What is the effect of a successful delegation operation?
Problem Statement and Motivation

- What does delegation mean in a workflow system?
- What is the effect of a successful delegation operation?
- The answer may depend on
  - the workflow execution model (WDEM or UDEM)
  - the type (abstract or concrete) of the delegated task
  - the type (grant or transfer) of the delegation operation
Problem Statement and Motivation

- What does delegation mean in a workflow system?
- What is the effect of a successful delegation operation?
- The answer may depend on
  - the workflow execution model (WDEM or UDEM)
  - the type (abstract or concrete) of the delegated task
  - the type (grant or transfer) of the delegation operation
- Existing work does not distinguish between execution models and only considers grant operations
Tasklists

When a workflow specification is instantiated a tasklist is generated for the workflow instance

- A tasklist is a list of user-task pairs
- A user is obliged to execute tasks to which she has been assigned

There are two different ways in which tasklists may be managed

- The tasklist may be completely determined when the workflow is instantiated (static tasklists)
- Entries for pending tasks are appended at runtime (dynamic tasklists)

Tasklist management has no effect on delegation (unless constraints exist on task execution . . . )
Delegation of Concrete Tasks

In WDEM each concrete task is assigned to a single user who is obliged to perform that task

- Grant delegations can not occur

A user may transfer a concrete task to which she is assigned

- The delegatee is required to perform the task
- The effect of a transfer of a concrete task is to update the tasklist
Delegation of Abstract Tasks: Authorization

Authorization for abstract tasks may be granted or transferred

- If the (abstract) task is granted then both delegatee and delegator may be assigned a concrete instance of that task in a tasklist

- If the task is transferred then only the delegatee may be assigned that task in a tasklist

Successful abstract task delegation must update the authorization information

- The delegatee is now authorized for the task
- The delegator may not be authorized (depending on whether it was a grant or a transfer)
Instances of the delegated abstract task may have been assigned to the delegator

- What should happen to such task assignments if the delegation is a transfer?
- A cascading transfer delegation transfers all existing assignments to the delegatee
- A non-cascading transfer delegation causes the delegator to retain all existing assignments
### Summary

<table>
<thead>
<tr>
<th>Task type</th>
<th>Operation</th>
<th>Update Tasklist</th>
<th>Update Authorization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concrete</td>
<td>Transfer</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Abstract</td>
<td>Grant</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Abstract</td>
<td>Transfer</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Abstract</td>
<td>TransferCasc</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

1. Concrete tasks can only be transferred and authorization information does not change
2. Abstract tasks may be granted or transferred and authorization information always changes
3. Cascading transfer changes tasklists
Introduction

Tasks are not assigned to users by the WfMS

► The WfMS maintains a list of pending tasks
► Users select tasks to perform
► The WfMS includes a reference monitor that decides whether the user is authorized to perform the task
Delegation

There are no tasklists

- Delegation of concrete tasks is meaningless
- Cascading transfer of abstract tasks is meaningless

A user may delegate an abstract task

- Grant and transfer are supported
- Grant and transfer both update authorization information
Introduction

WDEM

UDEM

Concluding remarks

Questions
# Summary of Delegation Effects

<table>
<thead>
<tr>
<th>Delegation operation</th>
<th>WDEM</th>
<th>UDEM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Abstract</td>
<td>Concrete</td>
</tr>
<tr>
<td>Grant</td>
<td>$u$ retains $t$</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>$v$ gains $t$</td>
<td>n/a</td>
</tr>
<tr>
<td>Cascading transfer</td>
<td>$u$ loses $t$; transfers all $t$</td>
<td>transfers $t$</td>
</tr>
<tr>
<td></td>
<td>$v$ gains $t$; receives all $t$</td>
<td>receives $t$</td>
</tr>
<tr>
<td>Non-cascading transfer</td>
<td>$u$ loses $t$; retains all $t$</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>$v$ gains $t$; does not receive $t$</td>
<td>n/a</td>
</tr>
</tbody>
</table>

$t = \text{abstract task}; \ t = \text{concrete task}; \ u = \text{delegator}; \ v = \text{delegatee}$
Contributions and Observations

This work represents the first attempt to consider the effect of different types of delegation on authorization information and tasklists in WfMS

- Not conceptually deep, mainly preparatory work
- Does highlight the differences between different types of delegation and different execution models
- Part of larger programme of work to consider the effect of delegation on workflow satisfiability

The paper also includes delegation of roles in WfMSs that employ role-based access control
Ongoing and Future Work

Consider authorization constraints, delegation and workflow satisfiability

- Execution of pairs of tasks is constrained in some way (separation of duty, binding of duty, etc.)
- Does permitting a concrete task delegation prevent completion of the workflow instance?
- Does permitting an abstract task delegation render workflow specification unsatisfiable?

Revocation and workflow satisfiability

- Does permitting a revocation request affect workflow satisfiability?