Towards Secure Electronic Workflows

Sebastian Fritsch, Vangelis Karatsiolis, Marcus Lippert,
Alexander Wiesmaier, and Johannes Buchmann

Technische Universität Darmstadt,
Department of Computer Science,
Hochschulstraße 10, D-64289 Darmstadt, Germany
sfritsch@cdc.informatik.tu-darmstadt.de

Abstract. Despite the introduction of information technologies in governmental administrations, most bureaucratic processes are still paper-based. In this paper we present a framework to transfer conventional, paper-based processes to electronic workflows. Thereby, the transformation to e-Government applications has two challenges. First, to find an equivalent description for the single activities and their interaction for defining the entire process. Second, to ensure the security of the process. We identified four types of activities that can be used as basic components for the workflows considered in our work. The security aspects of the electronic representation are ensured by further framework components, for example authentication or authorization. Finally, we present how this framework can be used for other scenarios and discuss some details of our prototype implementation.

Keywords: Workflow Security, Digitize Workflows, Workflow Engine, XPDL, XACML.

1 Introduction

Even though IT systems were introduced in most administrations, bureaucratic processes are still mainly paper-based. Many papers are moved from one desktop to another. Even if an electronic form is used, it will be printed to send it to other workflow participants. Another problem is security issues that appear if sensitive data is affected. There is a need for e-Government applications which are able to handle complete workflows from the initiation to the last workflow step without any media discontinuity.

1.1 Motivation

In our university the appointment of a new professorship is a traditional paper-based workflow. The purpose of this workflow is to initiate an invitation to tender, discuss the possible candidates, and finally negotiate on the contract conditions of the new professor. In this workflow many papers are moved among a lot of people. The creation, distribution, and management of those papers is a time and resource consuming task. With every new appointment the same

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 154–168, 2006.

steps must be performed. Therefore we choose to digitize this workflow. Security considerations exist in this case since personal information is involved. Security must be preserved and the goals to achieve are confidentiality, authentication, integrity, and non-repudiation.

In the federal state of Lower Saxony in Germany about 130 million of paper pages are used for purposes of state administration every year.[1] There is the need to digitize the administration processes in order to make them easier and reduce the amount of paper. They employ a PKI for achieving this. PKI is also used in the JobCard context.[2] This project deals with enabling the employees and employers to administrate their certification documents. All these workflows are in the digitization process. Therefore we need to address this fact as well as the security challenges that occur.

1.2 Contribution

This paper shows how to transfer the traditional university workflow to an electronic form. This workflow consists of a sequence of steps. We point out the security aspects since these are of great importance for the complete workflow. We develop a generic framework for e-Government applications, which supports the reuse of parts of the implementation.

The paper is organized as follows: Section 2 introduces the term workflow and discusses concrete aspects of how to transfer workflows to an electronic representation. Section 3 gives an overview of the basic components we isolated and their relevance in the context of security. Section 4 shows the implementation details of the framework and the workflow components. We explain in Section 5 how our components can be used for transferring other workflows. Section 6 draws a conclusion and describes the future work.

2 Transferring Workflows

This section gives an introduction to the terminology of the workflow context, to workflow engines and to the standard of internal representation used in our system.

The Workflow Management Coalition (WfMC) is an organization that introduced a standard for workflow descriptions. It defines workflows as follows:

> The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to procedural rules. [1, Page 8]

Business processes are defined as linked procedures or activities. Each workflow consists of one or more processes. Processes consist of activities or workflow

[1] `http://www.izn.niedersachsen.de/master/C5252172_N5505837_L20_D0_I3654280.html` (date of access 06.04.2006).

[2] `http://www.itsg.de/download/BroschuereJobcard.pdf` (date of access 06.04.2006).

steps that represent a piece of work. An activity requires human or machine interaction for execution. A workflow process has been completed if all its activities have been executed.

Digital workflows have several advantages over the paper-based ones. The first advantage is better process control. Second, auditing can be used. Third, the status of the workflows can be observed better. Fourth, enormous masses of paper can be avoided. These are general arguments that motivate electronic workflow management.

Applications and representation standards have been developed for handling and describing workflows. The applications are called workflow engines. They manage the sequence of workflow steps. Workflow engines can be described as run-time environments for processes. The workflow engine can be called from external applications to get or update the status of a workflow instance. Another way to work on active workflow instances are tool agents. A tool agent is an application that is called by the workflow engine directly. Mostly a tool agent has to solve one special problem, for example to send an email to all participants of a workflow. A tool agent can work on the given workflow data. Finally a tool agent can update the workflow state. Usually the associated workflow activity is completed when the tool agent has completed its task. For describing the processes, an XML based standard has been developed called XPDL [2].

Each XPDL description defines a package of workflow process definitions. Additionally, participants are defined, which are roles, persons, systems, resources or organization-units. A process defines the activities, for example to fill out a form. The activities are connected to each other by transitions. Further, route activities are used to realize decisions, branches, and merges in the process flow. Routing can be performed sequential or in parallel.

In each package, process or activity variables and attributes can be defined for characterizations or information storage. The workflow definition collected in these XPDL descriptions can finally be loaded into a workflow engine. In this engine the processes are instantiated.

Transferring a whole workflow is more than only transferring each step. It is not sufficient to transfer workflow steps into a web-application. Workflows can contain a lot of sensitive data. Therefore, the security properties are very important. The data's authenticity, non-repudiation, confidentiality, and integrity has to be provided for the whole workflow.

2.1 Related Work

The Electronic Circulation Folder (ECF) [6] has been proposed for realizing various e-Government applications. It is based on examining the way that typical processes in a bureau take place. This approach is based on the adoption of folders circulating among bureaus. These folders consist of two parts: the description and the content. The description part is used for describing a process as well as its status (for example at which office a document is found at a point in time). The content part contains all necessary data, like the documents needed in a process.

There are three important aspects that we can observe from the ECF concept. ECFs can contain all kind of data and documents. The flow or migration of an ECF defines the involved office workers and the set of steps that must be performed for a complete processing. Finally, the initiator of the workflow can define the migration of one individual ECF in a flexible way.

The step migration and the possibility to integrate the user in the definition of the processes was adopted in our work. The step migration is arranged by a workflow engine and the user integration by user modelling tools. But the ECF concept does not consider the security aspects which are the focus of our work.

In [5], Kandala and Sandhu present models for secure workflow management systems. They are based on roles and the RBAC framework. In our work we concentrate on the security of a concrete workflow. We introduce components to achieve security. Our goal is to reuse these components for securing other workflows as well.

3 Workflow Scenario and Components

This section introduces the workflow components. First we present the application scenario *Appointment of a new Professorship*. After this, the relationship between the components is described. Finally, we explain the different functionalities of the components.

3.1 Scenario: Appointment of a New Professorship

The benefit of transferring the scenario to an electronic workflow is to save the paper and shorten the time an instance of the workflow needs to be finished. Also the flow and lifetime of the process can be controlled more easily. These facts lead to the introduction of e-Government in the scenario's context. An overview of the whole scenario is given in Figure 1.

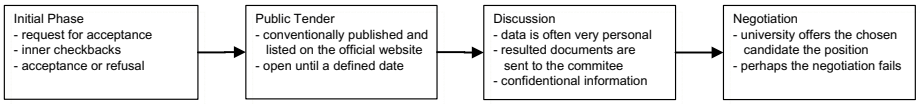

Fig. 1. Appointment of a new Professorship

The corresponding workflow can only be initiated by the request of a faculty's dean. The request includes detailed information of the intended professorship, the number of employees, the number of allocated rooms, and the period of time the advertisement will be open.

The request is answered by the university's president. The president's response depends on the answers of other administrative departments of the university. Authentication of all workflow participants and confidentiality of the workflow data is important at this stage of the workflow. After that, the dean is informed of the president's decision. In parallel the advertisement is initiated. It will be

public for a defined period of time. While it is open, candidates are allowed to view a detailed description of the appointment.

Here availability must be supported since the defined time of acceptance has to be guaranteed. The candidates can send their application for evaluation. All information sent by the candidates has to be confident and unaltered. Additionally, the authenticity of the candidates has to be ensured.

After closing the advertisement, all applications are made available to all members of the appointment commission for internal discussion. These discussions are done on personal and electronic basis. Everything discussed in the commission is undisclosed and must be kept confidential. The discussion may include personal opinions of the commission members. At this stage of the workflow, confidentially is important and no information is allowed to appear in public.

The commission agrees on an ordered list of three candidates. This list is still closed, so it is only allowed to be read by the commission members and the university's president. Next, the president has to approve a candidate from the list, usually the first one. In the next step, the president has to negotiate with the chosen candidate on his conditions. This negotiation includes a lot of personal data. This forces to ensure confidentiality and integrity.

If the negotiation succeeds the workflow is finished. In case of failure, the president has to choose another candidate. To reduce the scenarios complexity, we assume that the entire workflow has to start over again. Thus, there is no need to observe all special cases in the workflow description.

3.2 Components

We use a top-down approach to design and implement the workflow. This also suggests a modular design which allows the reuse of basic components. We developed two different types of components. First, the activity components which are mapped to the activities that are used to build the workflow processes. Most e-Government applications are form based. Therefore, activity components are used for processing form data, inform affected workflow participants and handle branches in the logical process flow. Second, the technical components to realize security properties and to manage the entire workflow process. An overview of the relationship of the components is given in Figure 2.

3.3 Activity Components

As described above the activity components are used to represent most of the workflow activities. Four components have been developed.

Forms Component. In the scenario description some requests were considered, which are traditional paper form based. We represent them with the Forms component. Forms are composed of classical form elements like text fields or choice fields. This component can be suspended, its input can be frozen at any time. Later it can be resumed from storage. Offline usage is also possible. A workflow participant can export a form from the system, work on the data and send it back to the system.

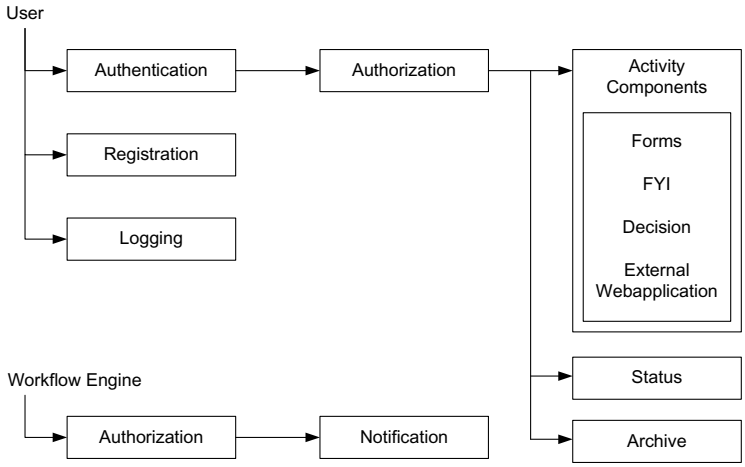

Fig. 2. Relationship of Components

FYI Component. FYI is the abbreviation of "for your information". For example the dean's request at the beginning of the workflow has to be shown to the president. This component presents some afore inserted information to a participant. This participant has to be informed only at this point of the process. After reading the information, the participant must commit having read it.

Because of using different routing strategies, the FYI component can be used as blocking component. That means the whole process has to wait until reading was committed, or it can be routed in parallel to the rest of the process.

Decision Component. In the scenario the president has to decide whether the dean's request is accepted or not. Later on, a candidate with whom the negotiations will start, must be chosen. These tasks are realized by introducing the Decision component. This component controls the process flow. Decisions from this component can be read in the following workflow steps and they can be used for defining which branch the process has to follow. To be concrete, the users are shown some kind of question that they have to answer and a decision therefore is met. The users' decision will be stored in a workflow variable.

External Webapplication Component. The advertisement and the electronic discussion of the appointment commission in the scenario are complex applications, which need special attention. They are linked to a process to provide important data, for example the applications of the candidates. We decided to introduce the External Webapplication Component. One external application can be linked to one process activity in a workflow.

The component mostly supports the input of data from external participants to a running workflow process. The data inserted to the web-application is available in the other workflow activities.

3.4 Technical Components

The technical components are used by the workflow system to support security properties and manage the workflow process.

Authentication Component. In the scenario's description a lot of authentication aspects have been introduced. Precisely, each request to the workflow system is based on an authentication.

The authentication is based on the user's knowledge or presentation of some information. Possible authentication mechanisms are, for example, username-password, PKI-based, or biometric authentication. Additionally, the physical presence of the user can also be supported as a traditional authentication mechanism.

The authentication ends with retrieving a list of groups in which the user is a member. The combination of both, that is the name and the group membership of the user, is called the user's identity. This identity is used in the next component, the Authorization component.

Authorization Component. This component restricts access to resources in the scenario, e.g. reading requests or access to the electronic discussion.

The authorization's decision takes place in a separated part of the application. The decisions must be enforced at the policy enforcement point. At this point the policy requests are generated and sent to the policy decision point.

If a user requests to perform a command on a resource, a policy request is generated. After sending a request the response returns a *deny* or *permit*. This is performed by the decision point.

Notification Component. The Notification component is used to inform participants for changes or news in the workflow proceedings. The component can be used for binding persons closer to the system when they use the system only sporadically, like the commission members do.

Security aspects have to be addressed in this component because information is sent out of the workflow system, and data can no longer be controlled. We decided to perform policy checks in this component, which are run by the authorization component.

Registration Component. The candidates are not members of the workflow system, but they have to send their applications to the system. Up to now our implementation is based on a smart card based public key infrastructure. Since it is not practical to integrate the candidates by registering them and providing them with a smart card, we have to find another solution.

Our solution is to provide temporarily valid certificates. The certificates and private keys are delivered in software. These certificates have to be mapped to the newly introduced external participants. These participants are handled as normal participants from the workflow system's point of view.

If a workflow process instance is finished, the corresponding certificates must be disabled. We chose not to revoke the certificates but to disable the authentication

possibilities associated with them. Thus the access control is delegated to the Authentication component.

Status Component. If a participant has to make a decision based on some information added in previous workflow steps, this component supports the user to get an overview of the whole workflow and its attached documents. In this component policy checks have to be performed.

Archive Component. When introducing e-Government applications a central question is, how data, which is finally available only electronically, is archived. This is even more important in workflow systems since workflows are finished at some point of time. So there is a need to archive and later reconstruct the workflow's processing and the related data. These functions must be provided by an archive component.

The Archive component was designed to allow access to already finished workflow processes. The documents of these workflows are still available for all participants who have the authorization to view these files. An important aspect is to decide whether changes to the authorization settings have an impact on any archived data. If this shall be avoided the authorization settings must also be archived.

Logging Component. This component logs authentications, resource or policy requests, and changes to the workflow states. These auditing mechanisms allow to detect problematic authentications or policy decisions.

4 Implementing Components

We have seen the components that we need for realising the workflow. In this section we see their implementation.

4.1 The PKI Installation

In our university a campus-wide smart card is used. Every student possesses a smart card that contains a key pair and a corresponding certificate. This certificate is used for digital signature purposes. In the second phase of the project a second key pair can be written on the card. This will be used for encryption. The employees of the university will receive a smart card, too, that can be used for encryption and digital signature. The encryption keys will be backed up. The employees are the ones that are using the framework. The technical entities (like web servers) are also being certified for supporting services like SSL.

The PKI is used for securing our electronic workflow. It offers the authentication, confidentiality, and non-repudiation services. The authorization part of the workflow is organized by the use of XACML [8]. The authentication and authorization information are combined together in order to realize the access control mechanisms needed in the framework. Therefore we benefit from the existence of the current PKI installation.

4.2 Framework Concept and Implementation

Our implementation is based on a four-tier architecture. The whole framework concept is shown in Figure 3.

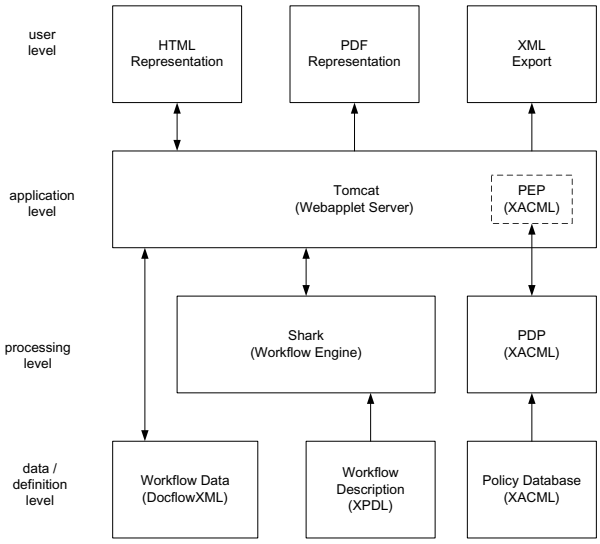

Fig. 3. Framework Overview

The main elements are the workflow engine and the web applet server. Our prototype is completely written in Java. Since Java offers platform independency, a flexible integration of our solution to existent systems is possible. We have chosen to use an Apache Tomcat web applet container for running the web application. Since we use XPDL as workflow description language we have chosen to use Enhydra Shark as the corresponding workflow engine. This workflow engine is queried by the central web application which is used by the workflow participants. The implementation of the components is done inside a servlet. The rest of this section will explain the implementation of the activity and the technical components.

4.3 Activity Components

The underlying software system, the workflow engine, enables to define attributes for each activity. The attributes are used as parameters to define the type and appearance of the different activity components described next.

All activity components are generally based on an HTML representation. In the XPDL definition can optionally be defined, if the inserted form data should be digitally signed. If this is enabled for an activity, the creation of the signature completes this activity.

Forms Component. The implementation of the Forms component is realized with HTML forms. The form structure is loaded from an external defined file similar to XForms.[3]

This component needs some attributes. First the XForms file which contains the structure of the form, and second the document's name to map the form data to the information storage place. The storage called Docflow[4] is document-centric organized.

If the workflow is suspended in this step, the data is saved to the Docflow file of the running instance. Later, the form elements input will be restored from that file. When completing the insertion, the data can optionally be digitally signed.

Another implementation that will be done as future work is a PDF based implementation. Therefore, the XForms information is read and a PDF document containing form elements is created.

FYI Component. This component shows some information to the assigned recipient. This information can be static or stored in the running workflow. The second case needs a reference to an internal workflow document. The participant has to commit that the information has been read. The blocking or non-blocking aspect is not affected by the implementation because this is determined by the process definition.

Decision Component. This component was introduced to control the process flow. Some type of question and possible answers is presented to the user. Finally, a workflow variable is set to the given answer.

This component needs the decision's question and answers. The name of the workflow variable must also be given. The value of this variable can be accessed in the next workflow processes.

External Webapplication Component. This flexible component is an extension of the workflow system implemented as a web-application. Complex workflow steps can be performed in such an external application.

To build such components, knowledge of the underlying workflow engine is needed. We have chosen to provide a lot of functionality in a library. This library was developed while building some example components. The most advanced example in our scenario is used for receiving applications to advertisements.

Integrating this component provided a lot of flexibility to our framework. Because of the web based structure there is no break in the representation to the user.

4.4 Technical Components

Authentication Component. Authentication has to be performed before a user is allowed to send a request to the system. Our implementation is mostly

[3] XForms is a standard to define form elements used in web-applications, see `http://www.w3.org/MarkUp/Forms/` (date of access 07.04.2006).

[4] As part of this work, Docflow was defined as a simple document management system which stores the workflow related data. We decided to implement this abstract definition with an XML document structure.

based on a smart card based public key infrastructure. Thus, digital certificates can be used for authentication. First, the webserver checks if a valid certificate is presented. If it succeeds, an HTTPS connection to the server is established.

Next it is checked if the certificate's distinguished name is allowed to log into the system. This information is stored in an XML file containing all valid users. If a user is allowed to log on, the user's group memberships are retrieved from an XML file as described in Section 3. In this file groups and subgroups are defined. It is possible to describe a complete hierarchy. After that, the identity of the user is known to the system.

If the authentication was not successful, a failure message is shown to the user. Further processing will not be done.

Authorization Component. We implement authorization by using an architecture based on XACML [8]. An XACML system is divided into three parts: the policy database, the Policy Enforcement Point (PEP), and the Policy Decision Point (PDP).

The policy database is an XACML file which contains the policies that the system has to enforce. The PEP is located on the application side and generates the policy requests which are sent to the PDP.

The PDP asks the policy database if a requested policy, a *(subject, object, command)* triplet, results in a *deny* or *permit* response. The PEP has to enforce the responded decisions. In the background an audit message is created which includes details about the requested resource and the response. Auditing is supported by this component. It can be enhanced by implementing the Logging component.

The workflow system has to determine the policy triplets. After collecting this information, the request is generated. If the response is *deny*, the system may fall back to request another command, for example if write access was denied, the system may ask in a second step to retrieve only read access.

We could have used XACML to accomplish the task of user authentication. The idea was introduced in [7]. Role Assignment Policies are used to determine a user's group memberships. The advantage of performing user authentication with XACML is the small number of different file formats. We did not use XACML for authentication purposes, because for determining the whole set of a user's group membership one request per available group is needed. This is very inefficient for a large number of groups.

Notification Component. This component has a wide range of implementation variants. We decided to use notifications based on email messages.

The content of these notifications can be static or dynamically filled with some document data inserted in the workflow process. We provide the possibility to enforce signing and encrypting of these notifications. This is no problem, if the infrastructure is certificate based, as our implementation which uses smart cards and X.509 certificates. Each user has an X.509 [3], [4] certificate, that enables the use of S/MIME [9]. This also includes the external workflow participants who are provided with a temporary valid certificate.

Policy checks have to be enforced, because information is sent out of the workflow system. Thus, the data leaves the security controlled system. The recipient of the notification is the subject that has to be policy checked.

Registration Component. An unknown user has to fill a web-based form in order to register to the system. Next, a new digital identity is created and a certificate is matched to his identity. The certificate and the corresponding private key is delivered to the user in a PKCS#12 [10] file by e-mail.

In the next step, the certificate must be installed in the user's browser. The PKCS#12 file can also be installed in an email client for securing the e-mail communication with a registered user. The created identity will be added to the user repository.

The further components, Status, Archive, and Logging have been designed but not implemented yet.

4.5 Conclusion

A prototype of the e-Government framework has been developed. We defined the configuration for the scenario *Appointment of a new Professorship*. This application was completely transferred to an e-Government application inside our framework.

However, since we developed a framework for transferring former paper-based workflows to e-Government applications, we have to show if and how other scenarios can be transferred. We decided to introduce two more scenarios and describe how the transformation process can be performed. This is discussed in detail in the next section.

To introduce new applications, three major steps have to be performed. The scenario or application has to be described in an XPDL workflow description. This can be done by using graphical tools. The paper forms used in the conventional workflow must be converted to an electronic representation, wherefore we use XForms. Finally, the security of the application must be defined. Documents introduced in the XPDL workflow definition are referenced in the policies. Additionally the participants and the corresponding actions have to be defined. After performing these three steps, new e-Government applications can be provided by our framework.

5 Applying the Framework to Other Scenarios

We have chosen two further applications from the university context to show how they can also be electronically transferred. The first scenario is called the *Request for Scholarship*. A student can request to receive a scholarship. The request is sent to the student office and is processed. The second scenario is the *Travel Expense Accounting* scenario. Employees request to refund the cost of their business journey. Both scenarios are explained in detail in the next two sections.

Scenario: Request for Scholarship

Two parties are involved in the Request for Scholarship scenario, a student and the student office. A student fills out a request for a scholarship and sends it to the university's student office. The affected office worker has to decide if further information is needed. If more information is needed, a request can be sent to the student who has to answer it.

Finally, the office worker (or a commission) has to decide if the student will receive the scholarship. The student will be informed about the decision. In this scenario the security targets are the authenticity of the student and the confidentiality of the workflow's data.

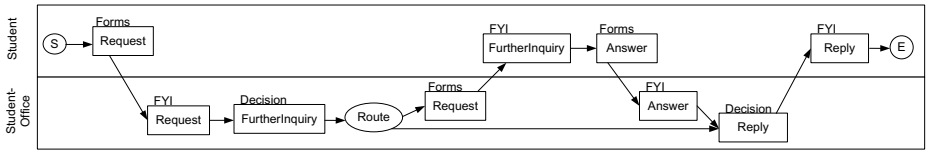

Fig. 4. Scenario Request for Scholarship

Scenario: Travel Expense Accounting

After a business journey, for example after visiting a conference, the employee is asked to fill out a travel expense accounting. This is given to the administrative office. The accounting is checked for completeness. If some information is missing the request is returned to the employee, who has to complete it. If the information is complete, the travel expense is accepted by the office, it is signed by the head of department and the workflow finishes.

In this scenario we have three involved roles: the employee, the office worker, and the head of department. The task to check the completeness and the task of acceptance is bound to the administrative office. All data concerning this workflow may only be read by these three roles. In addition only the employee is allowed to fill out such a form and only the department head is allowed to finally sign it.

We succeed in transferring both scenarios by using our framework. We show in Figure 4 what components have been (re)used to implement the Scholarship

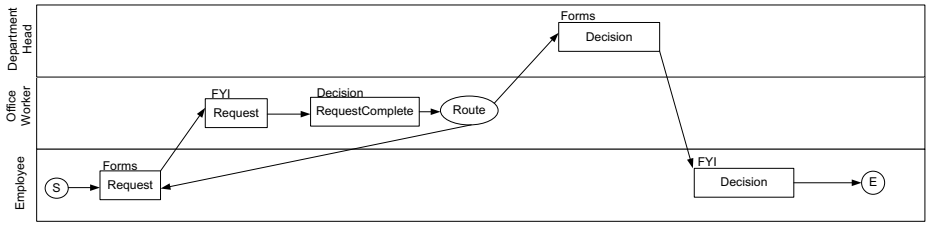

Fig. 5. Scenario Travel Expense Accounting

scenario. The implementation contains one routing branch. Which branch will be activated is decided in the *FurtherInquiry* element, which is an instance of the Decision component. A simple yes-no question is asked. In addition or alternatively the Notification component could be used to inform the student about the decision of the student office. The realization of the travel expense scenario is shown in Figure 5.

6 Conclusion and Future Work

We introduced a framework for transferring traditional workflows to electronic ones. First, we gave an introduction to the terminology. Then we showed the scenario *Appointment of a new Professorship* and how our framework fulfills its requirements by introducing the different components. Their concept and implementation have been developed in a top-down approach. Finally, we briefly showed how to transfer other workflows by using the same components.

Our studied workflows can be extended by a retrace functionality. This is important, for example, if a participant has not completely filled out a form and this step should be reassigned again. We call this a retrograde step migration. We must test whether the used workflow engine supports this functionality.

We can also enhance the scope of our framework to be used in other contexts. Arbitrary binary data can not be efficiently integrated in our current implementation, since forms are the basis of our data representation. Such data is contained for example, in a workflow that processes construction plans. An efficient representation of this data is required. Moreover, this data has special security properties that we must also consider.

References

1. Workflow Mangement Coalition. WfMC Terminology & Glossary, Document Number WFMC-TC-1011. Avalaible at `http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf` (06 Apr. 2006), February 1999.
2. Workflow Mangement Coalition. XML Process Definition Language, Document Number WFMC-TC-1025. Avalaible at `http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf` (06 Apr. 2006), October 2002.
3. R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure - Certificate and Certificate Revocation List (CRL) Profile. *IETF Request For Comments*, 3280, April 2002.
4. Recommendation X.509 ITU-T. Information Technology – Open Systems Interconnection – The Directory: Authentication Framework, August 1997.
5. S. Kandala and R. Sandhu. Secure Role-Based Workflow Models. In *Database and Application Security XV, IFIP TC11/WG11.3 Fifteenth Annual Working Conference on Database and Application Security*, volume 215 of *IFIP Conference Proceedings*, pages 45–58. Kluwer, 2001.
6. B. Karbe, N. Ramsperger, and P. Weiss. Support of Cooperative Work by Electronic Circulation Folders. In *Proceedings of the ACM SIGOIS and IEEE CS TC-OA conference on Office information systems*, pages 109–117, April 1990.

7. G. López, O. Cánovas, and A. F. Gómez-Skarmeta. Use of XACML Policies for a Network Access Control Service. In *The 4th International Workshop for Applied PKI, IWAP 2005*, pages 111–122, September 2005.
8. Organization for the Advancement of Structured Information Standards (OASIS). XACML 2.0 - OASIS Standard Specification Set. Avalaible at `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml` (06 Apr. 2006).
9. B. Ramsdell. Secure / Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification. *IETF Request For Comments*, 3851, July 2004.
10. Laboratories RSA. PKCS#12 v1.0: Personal Information Exchange Syntax. Avalaible at `http://www.rsasecurity.com/rsalabs/node.asp?id=2124` (06 Apr. 2006), June 1999.