DESIGNING A DISTRIBUTED WORKFLOW SYSTEM FOR E-GOVERNMENT

D. Beer, S. Höhne, H. Petersohn, T. Pöhnitzsch, G. Rünger, M. Voigt
Computer Science Department
Chemnitz University of Technology
Straße der Nationen 62, 09111 Chemnitz, Germany
email: dbeer@informatik.tu-chemnitz.de, {s.hoehne,h.petersohn}@nhconsult.de, {thpo,ruenger}@informatik.tu-chemnitz.de, mvoigt@viosys.com

ABSTRACT
In the course of modernisation within public authorities, governmental processes are being improved and made more cost efficient. This can be done by adapting techniques already successfully used in the private sector. Legal restrictions in data access as well as the cooperation of authorities in non-hierarchical networks, however, require a different approach to this subject. In this paper, the execution of workflows arising in e-Government, is analysed and problem areas that arise out of the distributedness and heterogeneity of communicating systems are described. Furthermore a distributed architecture for workflow systems and protocols solving these problems are proposed.

KEY WORDS
distributed workflow system, hierarchical workflow, e-Government, software architecture

1 Introduction

Public authorities are forced to offer their services in a more efficient, more transparent, and more cost effective way. The result is a tendency to orientate towards the developments within the private sector. Decentralisation of organisational processes combined with the establishment of electronic networks, a distinct orientation towards the customer as well as thinking in key processes are only some examples for this development that made its way into the public sector. The change of paradigms on the path to become a more customer-oriented service provider constitutes a focal point within the actions of modernising the utilisation of information and communication systems (ICS). The transfer of developments in e-Business, which is predominantly influenced by the private sector, to governmental tasks necessitates the optimisation of public services and procedures by using ICS more efficiently, especially Internet and web-based technologies.

The project RAfEG[1] (Reference Architecture for e-Government) aims at the development of a software architecture suitable to support and realize transaction oriented governmental services by optimising processes. RAfEG emphasises the realization of real and integrated transactions. Thereby, the cross-departmental or department-internal IT-processes, respectively, consistently have to access the same sources. It has to be dealt with topicality, data consistency, integration of different databases, and distributed responsibilities.

The architecture RAfEG is a holistic approach to cover many relevant aspects like the formal description of technical connections up to the development of distributedly acting software components of governmental management processes. The research work concentrates on the following key issues:

- concepts leading to an efficient utilisation of heterogeneous systems for interactive applications in the scope of e-Government,
- a toolbox system containing models, methods and further software technological elements like interfaces and protocols for an almost completely electronic realisation of administerial processes,
- selection and development of architectural parts that consider legal directives of the administerial process flow by using formal (axiomatic and automation oriented) specifications combined with a downstream transformation,
- components of flexible process-supporting and process-control software.

The prototypical realisation of the architecture is based on plan approval procedures, taking the city of Leipzig as a reference. Plan approval procedures precede every constructional project which has influence on the environment and is subject to authorisation. Public agencies (e.g. power supply companies, post, telephone companies, environmental protection organisations) form one of the biggest group of involved participants. As a consequence processing documents of various media and data formats is one of the main characteristics. Legal restrictions are additional determining factors.

The RAfEG architecture is designed as a distributed component-based software framework. Necessary models,

[1] RAfEG is supported by the BMBF (Bundesministerium für Bildung und Forschung), the German Ministry for Education and Research.

methods, interfaces and protocols are provided to electronically implement suitable governmental processes. One of the main directives is reusability for other governmental activities. At present there is no well-founded knowledge based on detailed analyses about cost structures (i.e. cost drivers, cost elements, capabilities to save expenses, synergy effects) of institution spanning business workflows in public sector.

This article describes a conceptual design of an architecture for distributed workflow systems with special requirements of typical e-Government processes. Section 2 and 3 introduce various requirements and restrictions in governmental environments as well as specifics considered for plan approval procedures. Section 4 describes the architectural approach of workflow systems able to process distributed workflows. Section 5 presents a possible technology to realise distributed processing and Section 6 concludes.

2 Workflows in Governmental Environments

What makes governmental workflows different from those used in e-Business are its cross-institutional links. Several authorities taking part in one common workflow may have to share some resources, e.g. documents, while at the same time having tight limitations on remote data access and electronic interaction with one another. Workflows in e-Government are bound to laws and have to comply with legal formalities. Every authority has a well-defined set of data which it is allowed to share with other authorities, and strong regulations on the confidentiality of other data. These aspects require the decomposition of the common workflow into sub-jobs with a defined interface containing all the information that are to be shared. Each authority may use its own document management system and authentication methods for the sub-job without granting remote access to its resources. Those jointly working workflow sub-jobs guide the executives in a distributed environment through complex processes helping to keep all necessary rules and time limits. As a consequence the workflow combined with the electronic management of documents reduces the workload of the employees and enables faster processing of administerial tasks.

Due to the legal background of administerial tasks, defining workflow in e-Government is easier as it is for non-formalised processes. However the separate definition of workflow sub-jobs adds dynamics and complexity to the distributed workflow and requires well defined interfaces between them. In order to cope with the heterogeneity and to allow for interoperability between the workflow systems of different authorities, the resulting framework should mainly be based on open standards. The KBSt[2], the counselling and coordination body of the German government in the realm of information technologies, has put together a portfolio of standards and architectures for e-Government applications (SAGA)[1] that fulfil the demands of openness and usability in distributed environments. During the process of selecting components to be used in RAfEG, the suggestions made in SAGA were compared with alternatives considered to be suitable. Among other standards, the use of XML in combination with XSLT[3] was regarded as a core demand.

Furthermore a standardised way of connecting workflow and other services to the central authentication system had to be found. The activities in governmental workflows are not assigned to individuals but to roles. This role-based model, which helps to comply to legal regulations on the confidentiality and access limitations of governmental data, is used in other components of the RAfEG-architecture as well, like the document management system (DMS). Using roles allows for a more flexible, centrally controlled way to adapt to organisational and personnel changes like setting up a proxy for an executive being off sick. Additionally this can be used for a controlled way of gaining access to the DMS by the workflow engines for up- and downloading documents related to a running workflow.

One of the main aspects in governmental business processes are time limits that have to be kept. Some processes like tendering procedures, have minimum terms whereas others have a due date. It is part of the workflow engine to guide the governmental staff to adhere to these terms. Whenever tasks are assigned to employees for manual processing, information on associated terms has to be given. In case an activity of a workflow has been assigned to a person who unexpectedly becomes unavailable due to sickness or for other reasons the work item can not be finished and may cause starvation of the workflow. The workflow engine thus has to detect tasks that require some action and run the risk of exceeding a term. It has to escalate those actions in their importance as well as to bring such issues to the attention of the administrator so that such problems can be solved in time by manual intervention.

For a workflow processing an administerial procedure, several key aspects can be determined.

- Firstly, the parallel execution of activities throughout the public authority, which can consist of spatially distributed municipal offices.
- Secondly, the parallel execution of activities within a workflow, which requires means of locking as well as an exchange of workflow properties.
- And finally, the long time frame, sometimes several years, between the start and end of a workflow.

These aspects necessitate a workflow software that supports the distributed, fault tolerant, and parallel execution of a workflow.

[2] Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung

[3] Extensible Stylesheet Language Transformations

3 Geographic Information Systems

Dealing with geospatial information is a fundamental part of plan approval procedures. Many tasks done by hand so far can be partly or fully automated. Workflow systems must therefore cooperate with various installed geographic information systems (GIS). GIS form an essential tool for collecting, modelling, analysing and visualising of spatial data. So-called GIS software implements those functionalities which may differ from task to task. The main focuses of plan approval procedures are the exchange of relevant spatial data between involved authorities without the necessity of media change as well as the creation of official plans in form of maps available via Internet.

Documents submitted by the agency responsible for the building project contain planning information in form of files consisting of CAD drafts and geographic maps. They are only available in proprietary formats by the software used. The Road Construction Office in Leipzig uses "CARD/1"[4], which has been adapted to the special requirements of the assignment. The Regional Commission in contrast utilises "ArcInfo"[5].

Besides the highway board department, there are a lot of public agencies (e.g. townships, water, power and environmental authorities, post and rail). They receive relevant parts of the documents from concerned authorities. Almost all public agencies are equipped with GIS software. Due to the fact that numerous different GIS solutions exist, more than a dozen systems of various versions are in use (e.g. MapInfo, Poligis, Tiffany, Microstation, Gaja Kommunal, SiCAD, Archikart, Gaja Gis, LiCAD open, SmallWorld, Intergraph, Gestra, AutoCad).

The advantage of software specific data formats is the optimal data storage. However the resulting incompatibility between different softwares limits an efficient data exchange between all institutions involved in the process.

One possible solution is to make use of a software component which converts spatial data from arbitrary into some common formats without loosing any essential information. Another possibility is to determine standardised interchange formats, for example DXF[6] which is suitable for most GIS and CAD[7] systems.

The second main focus concerns the dispatch of geographic maps and plans via Internet. Thus interested citizens have the possibility to inform themselves about intended building projects. For these purposes a server application can be used which implements the map server specification [2] developed by OpenGIS Consortium[8]. The map server has to create the required section of the map as a raster image dynamically and has to perform simple map content requests.

[4] http://www.card-1.com

[5] http://www.esri.com/software/arcgis/arcview/index.html

[6] Drawing Exchange Format

[7] Computer Aided Design

[8] http://www.opengeospatial.org/

The workflow system is generally responsible for running process instances in involved public agencies and for coordination of data exchange if technically feasible. On the basis of lists containing any information about available public agencies and given criteria, identification of affected ones can be done (partially) automatic. The workflow system uses predefined mappings to automatically decide which communication channels are appropriate for particular agencies (e.g. direct network connection, electronic mail, letter post, etc.) and adapts execution of steps containing communication. Because public agencies receive only relevant parts of documents and spatial data, automatic filtering and map extraction including automatic transformation into appropriate formats are necessary.

4 Design of Distributed Workflow Systems

4.1 Basic Architecture

Figure 1 depicts a prototypical setup of a governmental workflow management system, which is based on the Workflow Reference Model defined by the Workflow Management Coalition (WfMC[9]) [3]. Although only two public authorities are shown in the example, the same applies to any number of cooperating authorities.

In each authority the workflow enactment service acts as the central control system while running several cooperating workflow engines. There are two interfaces for the processing of activities, namely, one for human interaction and one for automatic application invocation. The attached document management system (DMS) serves for the maintenance of electronic process files.

4.2 Workflow Description Languages

The design and requirements of the workflow system do have a direct impact on the definition language used to model and describe the workflow. As the whole workflow system is used in a heterogeneous distributed environment, the workflow design language has to support these aspects. A common method to face heterogeneity in documents is to use the standardised markup language XML. In case some special elements or attributes have to be used in the workflow description they can easily be added by using XSLT that transforms a given XML-description into the structure required for the workflow process. Defining a workflow, however, is not a one-time task, but due to refinements, changes in business procedures and reevaluations, forms an evolutionary process. Different modelling software products are required for those refinement steps, each of which may use different internal data formats. To assure interoperability between all tools the export into an XML format, as for instance the XML Process Definition Language (XPDL) [4], has become a common feature.

[9] http://www.wfmc.org

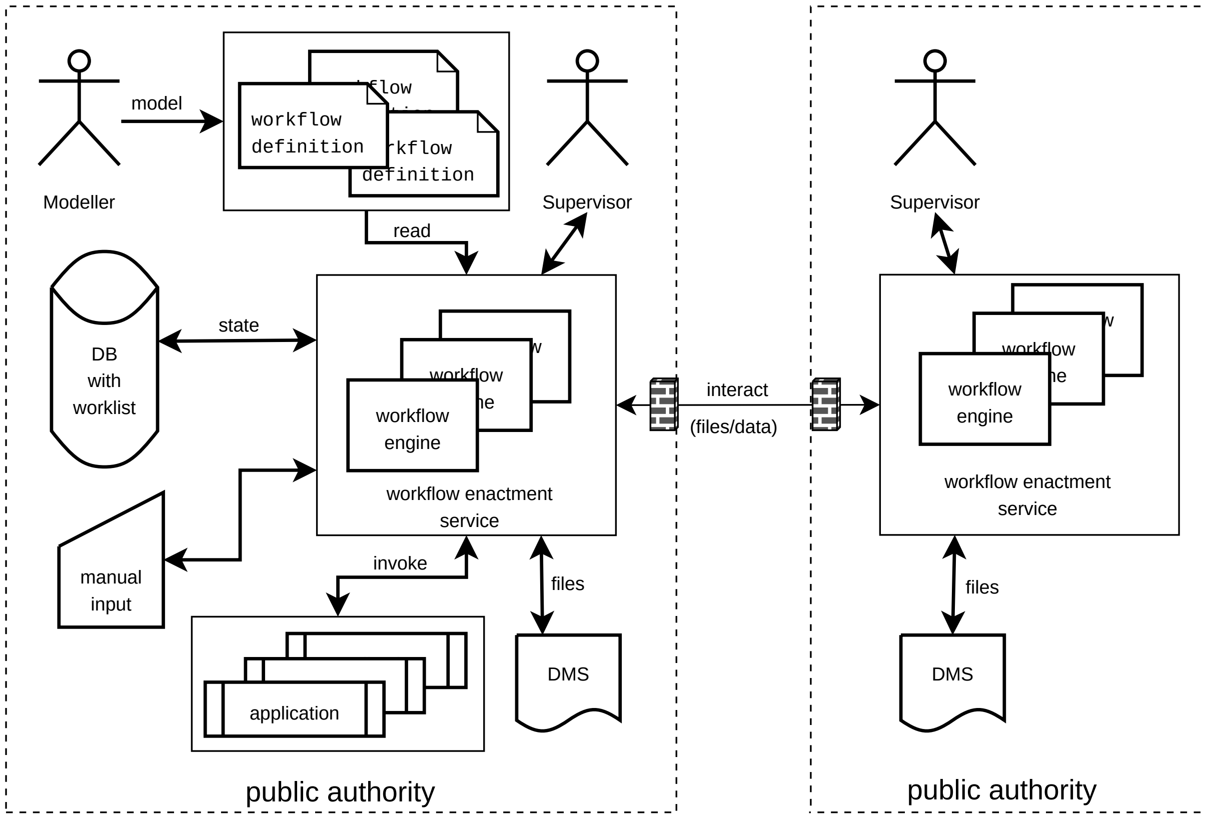

Figure 1. Scheme of the workflow system used in RAfEG, based on the WfMC Workflow Reference Model. It shows the workflow enactment service as the central component reading different workflow definitions and controlling several workflow engines, as well as its connections to a central database (DB) and document management system (DMS) as data repositories. Furthermore the activities between components and the point of interaction with persons and applications are shown. The second authority, identical in its structure is used to show the interaction between any two authorities.

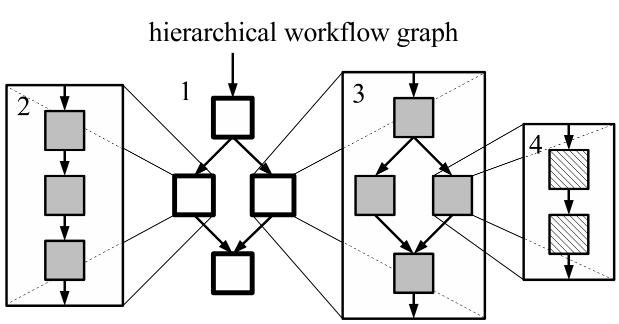

Figure 2. This figure shows an example of a hierarchical workflow. The exploded boxes represent activities containing sub-workflows. The numbers are used to identify those sub-workflows with tree nodes in Figure 3.

All processes in an inter-organisational workflow that reside outside the administrative scope of an authority constitute a black box [5]. Many process description languages – including XPDL – provide an opportunity to model single activities as sub-workflows, which in turn consist of other sub-workflows and so on. Consequently applied this leads to a hierarchical workflow. In Figure 2 a simplified illustration of an example workflow is shown in form of a directed graph. The nodes either represent single activities or boxes containing other workflows.

4.3 Communication Layers and Protocols

In order to achieve interoperability between individual workflow systems of the same or of different vendors, it is necessary to establish a standardised connection for communication and data transfer. It has to be considered that typical sub-workflow invocations – especially in governmental workflows – can take a long time, ranging from a few minutes to several months. Thus, simple remote procedure calls sufficient for short-living tasks are inappropriate for our design. Hence, remote requests have to occur asynchronously.

In distributed environments, there are two major techniques for calling remote processes. In the first case the calling workflow can continue its work immediately after the remote process has been started. Thus the task is completely handed over to the remote engine. This is called a chained workflow. In the other case the calling engine has to wait until the task is completed and the result is available, hence it behaves just like procedure calls. This is called a nested workflow. Of course, both methods can be combined inside one single workflow definition. Despite of the different behaviour both techniques are modelled in the same way, as described in Section 4.2.

As a precondition for interconnection, potential workflow systems must provide their service to all other workflow systems. This is realised via naming services or Uniform Resource Identifiers (URI) [6], which depends on the chosen implementation.

Requests are passed via an interface whose operations are initially divided into two groups:

1. The first group consists of operations for instantiation of remote processes, for passing initial parameters and for starting those processes.

2. Operations for monitoring purposes belong to the second group, for instance to retrieve the process state (running, suspended, completed, terminated, etc.), the work progress or the final results, which are of importance only for nested workflows.

In order to find out when the remote task is completed the originating engine can frequently poll the state of the process instance. This consumes resources of the operating system and the network on both sides. To avoid this behaviour the called engine can notify the calling engine as soon as the process has been completed. For that purpose the interface has to be expanded by a third operation category, which allows

3. the active workflow engine to wake up the waiting engine and to return results or to inform about abortion or termination.

When using distributed workflows on heterogeneous systems, several inconsistencies of object types, data formats of data relevant for the workflow, and naming conventions can occur. In order to cope with vendor specific object and data views it is recommended to interpose a gateway application between communicating workflow systems to perform transformations. The gateway application can also support different protocol environments by applying mappings to encode the API calls and associated parameters.

4.4 Adaptation of Workflow Instances to Specification Changes

As mentioned before, governmental processes usually span a period of several years. Thus changes in process definitions are likely to occur and have to be included not only in the workflow description but also in running instances. Modifications can be necessary if the execution of a process is affected by changes of laws or regulations. As a result it might happen that additional activities must be included or some activities become redundant. In both cases the structure of the workflow must be altered.

For better illustration the hierarchy can be seen as a tree with the basic workflow as its root. The leaves represent atomic activities which are not divided any further (see Figure 3). Activities that do not lie on any path from the root to running instances are currently not instantiated.

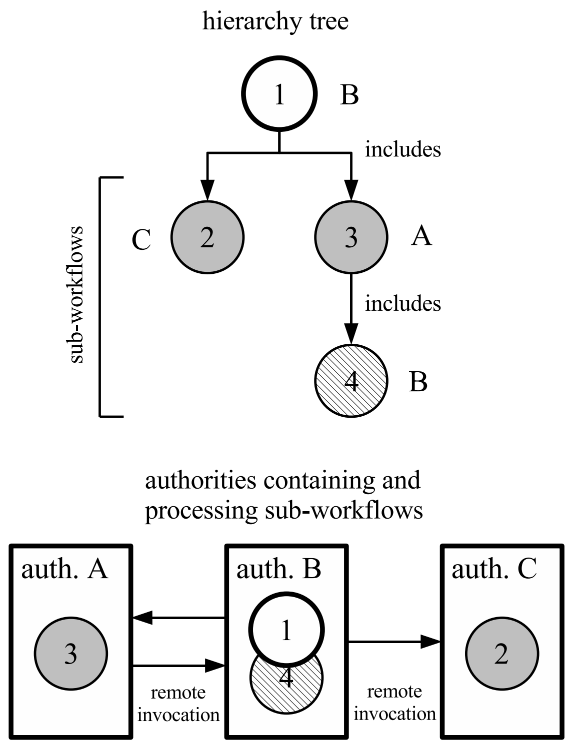

Figure 3. This figure shows the hierarchy tree that corresponds to the workflow in Figure 2, and the assignment to several authorities A, B, or C that remotely invoke the sub-workflows. The mapping of sub-workflows to tree nodes is designated by filling pattern and numbering.

Modifications of sub-workflow definitions have no effect until the next instantiation of the affected step takes place. Typical workflows for plan approval procedures contain loops. Hence most of the activities are passed through multiple times and are thereby updated. Because the initial workflow (which contains the start activity) corresponds to the root node of the hierarchy tree, the process instance is only created once and will be destroyed not until the whole workflow processing is finished. Hence, modifications of the root process definition have no influence on already started processes. Whenever an immediate change of this process definition is required, manual intervention is inevitable. But it is more likely that details in sub-workflows are changing instead of fundamental procedures.

Since there are data and control dependencies between preceding and subsequent activities, modifications are subject to various constraints. It has to be defined exactly which data should be transfered on remote invocation. Both, order of data and the data types must be equal on both caller and calling side. If there is a change necessary on one side the same changes have to be made on the other side, respectively. Furthermore it has to be assured that a correct identification of sub-workflows on remote systems is possible, i.e. the names or id's must correspond and must be unique.

4.5 Central Administration and Monitoring

Due to the fact that governmental processes are subject to legal restrictions, the administration of workflows is limited to individual authorities. Thus every authority has its own workflow supervisor.

The principal duties of a supervisor is to observe all running processes and to eventually reassign human and automatic resources. This can be necessary if a person in charge is temporarily not available, due to illness for example. In order to avoid exceeding the deadline, tasks that are already assigned to that person must be released and assigned to a proxy with appropriate skills.

Because of the complexity of governmental workflows the occurrence of failures can not fully be excluded. In such cases manual interferences are required. Even changes in workflow definitions could be necessary as described in Section 4.4.

Finally the monitoring tool is capable to execute different analyses and statistical functions, e.g. work throughput or utilisation ratio.

5 Choice of Technology

Because of installed firewalls (see Figure 1), the communication channel between authorities can put restrictions on the traffic allowed to pass. This has influence on the choice of the utilised communication protocols.

One popular technology to consider is web services. Existing web service protocols work best when the service can provide an answer quickly, within a couple of minutes at the longest. Such short-living services could simply be synchronously started, e.g. using SOAP, and the caller has to wait for completion. As already mentioned, typical sub-workflow invocations can last several months. Thus, there is a need for an asynchronous web service protocol (AWSP) which can start, control, and monitor an instance of a web service asynchronously. The Wf-XML protocol developed by the WfMC extends this approach by running this service on workflow engines [7]. The service factory maps to a process definition, the service instance maps to a process instance. Wf-XML provides a message-oriented middleware. One workflow engine sends a Wf-XML coded message to another workflow engine including all necessary parameters which corresponds to a remote procedure call. To ensure that this technique works both engines have to provide an appropriate API which is capable to parse Wf-XML messages and is able to properly react.

The generally used transport mechanism is HTTP. This enables the Wf-XML protocol to operate between systems even protected by firewalls, whose application can be expected due to the high security regulations existing in governmental environments.

6 Conclusion

We have presented an architecture for distributed workflow systems for e-Government. Distributed execution of workflows represents an important step towards efficient realisation of cross-institutional decision making processes. Many standards still have to be developed until general purpose workflow vendors completely support interoperability in heterogeneous environments.

References

[1] Bundesministerium des Inneren (Hrsg.), SAGA Standards und Architekturen für E-Government-Anwendungen, 2003.

[2] Open Geospatial Consortium Inc., Geographic information - web map server interface, 2004.

[3] David Hollingsworth, Workflow Management Coalition – The Workflow Reference Model, 1995.

[4] Workflow Management Coalition. Workflow Process Definition Interface – XML Process Definition Language, 2002. Document Number WFMC-TC-1025, Document Status 1.0 Final Draft.

[5] Michael zur Muehlen, *Workflow-based Process Controlling. Foundation, Design, and Implementation of Workflow-driven Process Information Systems.* (Berlin: Logos, 2004).

[6] T. Berners-Lee, R. Fielding, and L. Masinter, RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax, 1998.

[7] K. D. Swenson, M. D. Gilger, and S. Predhan, Wf-XML 2.0, XML Based Protocol for Run-Time Integration of Process Engines, 2003.