Adaptive Outsourcing in Cross-Organizational Workflows

Justus Klingemann, Jürgen Wäsch, and Karl Aberer

Integrated Publication and Information Systems Institute (GMD-IPSI)
GMD – German National Research Center for Information Technology
Dolivostr. 15, D-64293 Darmstadt, Germany
{klingem,waesch,aberer}@darmstadt.gmd.de
http://www.darmstadt.gmd.de/~{klingem,waesch,aberer}

Proceedings of the 11th International Conference on Advanced Information Systems Engineering (CAiSE'99), Heidelberg, Germany, June 1999, pp. 417-421.

Adaptive Outsourcing in Cross-Organizational Workflows*

Justus Klingemann, Jürgen Wäsch, and Karl Aberer

German National Research Center for Information Technology (GMD)
Integrated Publication and Information Systems Institute (GMD-IPSI)
Dolivostraße 15, D-64293 Darmstadt, Germany
{klingem,waesch,aberer}@darmstadt.gmd.de

Abstract. The workflow concept has been very successful in streamlining business processes by automating the coordination of activities, but has so far been limited to the use within single organizations. Any attempt to distribute workflows among different organizations has to face the problems posed by the complex relationship among autonomous organizations and their services. To address these problems we propose a service-oriented model for cross-organizational workflows. Modeling the workflow execution as a cooperation of services allows different organizations to interact via well-defined interfaces. We further show how the execution can be optimized by selecting services depending on their contribution to quality criteria of the workflow.

1 Introduction

Competitive markets force companies to minimize their costs while at the same time offering solutions which are tailored to the needs of their customers. This urges organizations to form virtual enterprises by outsourcing activities to external service providers. Hence, business links with other organizations have to be set up and managed. This has to be achieved in a fast and flexible way to guarantee a short time to market while allowing a dynamic reaction to new customer demands and changing offers of service providers in electronic commerce environments.

Information technology has provided different technologies to address these requirements. The workflow concept [1–5] has been very successful in coordinating and streamlining business processes but is so far limited to a single organization. On the other hand, the tremendous growth of global networks like the internet provides the possibility to efficiently exchange data and communicate with a large number of possible service providers. Thus, workflow management systems (WfMS) can limit their scope no longer to a single organization but have to exploit the network infrastructure to cross organizational boundaries.

However, the extension of workflows beyond the borders of a single enterprise raises new challenges. One important challenge is the necessity to choose

* This work has been partially supported by the ESPRIT project CrossFlow (www.crossflow.org). However, it represents the view of the authors.

among different services that potentially satisfy the customers requirements. In particular, it has to be decided which activities or group of activities should be outsourced to which business partners. Relevant criteria with regard to that decision are the required time, cost, or the adherence to domain-specific quality of service parameters.

The autonomy of the participating organizations implies that the initiator of the workflow has only limited control over the outsourced activities. This requires that both sides agree on an interface which allows the service requester to monitor and probably control the outsourced activities to a certain extent.

To address these problems, we propose a service-oriented model for cross-organizational workflows. Modeling the workflow execution as a cooperation of services allows different organizations to interact via well-defined interfaces. We further show how the execution can be optimized by selecting services depending on their contribution to quality criteria of the workflow.

2 A Service-Oriented Model for Cross-Organizational Workflows

In this section, we describe our conceptual model for cross-organizational workflows. However, we confine ourselves to the essential concepts which are necessary to understand the approach presented in this paper. For further details see [6]. We consider a workflow as a collection of activities which are related by certain dependencies. A workflow is modeled as a graph with activities as nodes and edges which represent the control and data flow. In addition, *quality of service (QoS)* parameters are assigned to the workflow, e.g., the maximal duration and the maximal cost allowed for a workflow. Despite these more or less application-independent QoS parameters, domain-specific QoS parameter can exist.

In order to allow the execution of activities at runtime, we need to define a mechanism that assigns activities to particular "agents" that are responsible for the execution of the activity instances. Usually, in intra-organizational workflows, agents are considered to be human beings or computer programs. If a workflow is allowed to span different organizations, there is a third kind of agents that can be involved in the execution of a cross-organizational workflow instance, namely *external service providers*. Therefore, we introduce a *service-oriented model*. We believe that this model is better suited for cross-organizational workflows while at the same time being applicable in ordinary enterprise-wide workflows. The basic entity in our model is a *service*. Informally, a service is an abstract specification of the amount of work that a resource promises to carry out with a specific quality of service. A service specifies which part of a workflow it covers. In general, a service is not restricted to execute a single activity of a workflow, it can span multiple activities. With $A(S)$ we denote the set of activities covered by service S. With each service a *service provider* is associated. This can be either an internal resource or an *external organization*. A service offered by an external organization is called an *external service*. Otherwise, it is called *internal service*.

In contrast to internal services, external services are not executed under the control of the service requester, i.e., the organization that runs and controls the cross-organizational workflow. The service requester has only a limited possibility to get information on the state of an external service execution and to influence this execution. Moreover, the internal work process of an external service might not be known to the service requester due to the autonomy of the service providers. Thus, while they are executed, services have to be treated as "black boxes" from the viewpoint of a service requester. Only the interfaces to and from the services are known by the service requester and which activities are subsumed by the service. This includes a specification how an external service can be invoked, which parameters have to be supplied, etc. Besides the interfaces, a service description contains the quality of service offered by the service provider to the service requester.

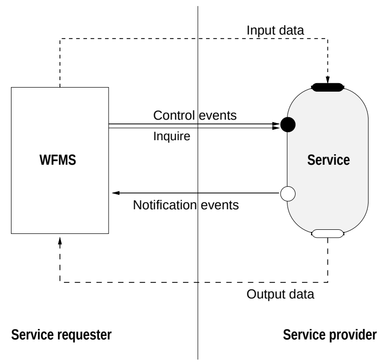

Fig. 1. Interfaces of a service and interaction with the WfMS

Figure 1 illustrates our view on a service. A service S has four interfaces. Two of them (*Input* and *Output*) are concerned with the data flow and two of them (*Control* and *Notification*) are concerned with control flow in form of control and notification events [6]. Control events (specified in the service description) can be sent by the service requester to the service provider in order to influence the processing of the service. Notification events are used to inform the service requester about the state of the processing of the external service. They can be sent as a result of an internal event of the service provider or as a reaction to an inquiry of the service requester.

3 Selection of Services in Cross-Organizational Workflows

In this section, we discuss how services will be selected to execute the activities of the workflow. We assume that all relevant information about services is made available in a *service repository*. Therefore, we do not consider issues like finding out which services are available.

The workflow and its activities have certain QoS parameters. Some of them are *mandatory*, others should be fulfilled as well as possible and are optimized with the following algorithms. For the sake of simplicity, we assume that the latter parameters can be aggregated into a single value which can be used to compare different service offers, i.e., each service S is assigned a value q which we call the *price* of the service. We further assume that this price can be calculated for all services offered. In [7] we have described a technique how a service requester can derive and validate QoS parameters of an offered service based on the observation of previous executions of this service.

Since the control flow part of a workflow specification establishes enabling conditions for activities, it is usually not clear at the start of a workflow which activities actually have to be executed. For example, a workflow can contain branches with alternative sequences of activities for certain parts of the workflow. Conditions can be evaluated when all necessary parameters are available due to the execution of previous activities. Service selection mechanisms can be differentiated depending on when activities are assigned to services. In this paper, we will consider the case where we assign only those activities to services of which, in the respective state of the workflow, we know that they have to be executed. In [6] we have extended this mechanism to more complex subworkflows which contain decisions yet to be made on which parts of the subworkflow have to be executed. We denote the part of workflow W which can be outsourced at time t as $Outsourcable(W,t)$.

In the following, we describe an approach how to select services for the activities in $Outsourcable(W,t)$. Let Σ be the set of services which offer to cover an appropriate part of the workflow in line with the specification of the workflow. In particular, these services fulfill all mandatory QoS parameters and execute the activities in an execution order that is in line with the workflow specification[1]. Additional restrictions can be included, e.g., the service requester might want to exclude the services of a certain service provider. The problem can then be formulated as finding a set of services from Σ which covers all activities in $Outsourcable(W,t)$ exactly once, while optimizing the non-mandatory QoS parameters. Mathematically, this problem is a minimal weighted exact set cover, i.e., can be formalized as follows: Find a subset O of Σ, that forms an exact cover, i.e., $\bigcup_{S \in O} A(S) = Outsourcable(W,t)$ and $\forall S_i, S_j \in O : A(S_i) \cap A(S_j) = \emptyset$ and such that the *cost* of the cover $\sum_{S_i \in O} q_i$ is minimized. This problem is NP-complete [8–10]. However, since the set activities to be outsourced is usually small, we expect that our approach is applicable to many real-world workflows. In [6] we have described an algorithm how to select an optimal choice of services

[1] Note, that this does not mean that the actual execution order has to be the same as in the workflow specification. Due to the autonomy of the service provider and the resulting limited monotoring capabilities we only require that the execution order cannot be distinguished at the interface of the service.

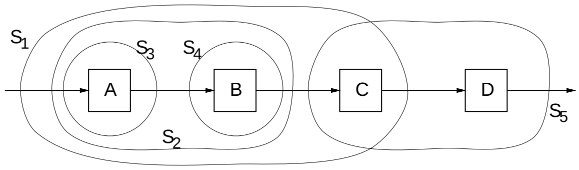

Fig. 2. Services offered for a sub-workflow

Service	S_1	S_2	S_3	S_4	S_5
Price	5	4	2	3	2

Fig. 3. Services and their prices

by solving the minimal weighted exact set cover. In the following we will give an example of this approach.

Figure 2 shows a simple subworkflow consisting of four activities. We assume that $Outsourcable(W, t) = \{A, B, C, D\}$. There are five services $S_1, \ldots, S_5$ which offer to perform parts of this subworkflow. Figure 3 shows the prices of these services. A correct selection of services would be, for example, $O_1 = \{S_3, S_4, S_5\}$ as each activity is covered by exactly one service. The cost of this selection is 7. Note that the cover $\{S_1, S_5\}$ is not allowed as the activity C is included in both services. However, besides O_1 there exists another exact cover namely $O_2 = \{S_2, S_5\}$ with cost 6. Thus, our algorithm returns O_2 as the optimum. Extensions of this technique for situations in which one activity is allowed to be covered by more than one service are discussed in [6].

References

1. D. Georgakopoulos, M. Hornick, and A. Sheth. An overview on workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3(2), April 1995.
2. G. Alonso, D. Agrawal, A. El Abbadi, and C. Mohan. Functionality and limitations of current workflow management systems. *IEEE Expert Journal*, 12(5), 1996.
3. S. Jablonski and C. Bussler. *Workflow Management — Modeling Concepts, Architecture and Implementation.* International Thomson Computer Press, 1996.
4. A. Dogac, T. Öszu, A. Biliris, and T. Sellis, editors. *Workflow Management Systems and Interoperability.* NATO-ASI Series. Springer Verlag, 1998.
5. A. Cichocki, S. Helal, M. Rusinkiewicz, and D. Woelk. *Workflow and Process Automation - Concepts and Technology.* Kluwer Academic Publishers, 1998.
6. J. Klingemann, J. Wäsch, and K. Aberer. Adaptive outsourcing in cross-organizational workflows. GMD Report 30, GMD — German National Research Center for Information Technology, 1998.
7. J. Klingemann, J. Wäsch, and K. Aberer. Deriving service models in cross-organizational workflows. In *Proceedings of RIDE-Information Technology for Virtual Enterprises*, Sydney, Australia, March 1999.
8. M. R. Garey and D. S. Johnson. *Computers and Intractability — A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, 1979.
9. R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.
10. D. S. Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems.* PWS Publishing, Boston, 1997.