Distributed and Parallel Databases, 14, 137–172, 2003

Cross-Organizational Transaction Support for E-Services in Virtual Enterprises*

JOCHEM VONK vonk@cs.utwente.nl
Center for Telematics and Information Technology (CTIT), University of Twente

PAUL GREFEN p.w.p.j.grefen@tm.tue.nl
Department of Technology Management, Eindhoven University of Technology

Recommended by: Boualem Benatallah

Abstract. In recent years, workflow management systems have become an accepted technology to support automation in process-centric environments. Lately, organizations concentrate more and more on their core business processes while outsourcing supporting processes to other organizations, thereby forming virtual enterprises. The organizations forming the virtual enterprise operate in a B2B e-commerce setting in which provider organizations perform e-services for consumer organizations. To apply workflow management technology in these virtual enterprises, current workflow management systems need to be extended to offer support for cross-organizational processes. Transaction support, already considered an important issue in intra-organizational workflow management systems, must be extended to deal with the cross-organizational aspects as well. This paper presents a high-level, compensation based transaction model and a flexible architecture to support this transaction model, as required by cross-organizational workflow processes. Characteristic of the model is the flexibility in rollback semantics by combining rollback modes and rollback scopes. This is supported by a dynamically composed architecture that is configured using the agreements that are specified in an electronic contract that has been established between the participating organizations. The transaction model supported by the dynamically composed architecture is implemented in a prototype system, based on commercial workflow management technology.

Keywords: transaction management, virtual enterprise, workflow management, B2B e-commerce, service outsourcing, cross-organizational business process, e-service

1. Introduction

Nowadays, workflow management is an accepted technology to support process-centric environments. The focus of organizations with respect to workflow management is now turning from secondary processes towards primary business processes. For this reason, it is important that workflow management systems ensure that these primary business processes are executed in a reliable and consistent manner. This can be accomplished by incorporating transaction semantics in the processes [16, 22, 28, 33].

*The work presented in this paper is supported by the European Commission in the CrossFlow Project (ESPRIT No. 28635). Partners in the CrossFlow project are IBM Zurich Research Labs in Switzerland, IBM France, SEMA Group sae. in Spain, Church and General in Ireland, GMD-IPSI and IBM Böblingen in Germany, KPN Research and University of Twente in the Netherlands. A short and earlier version of this paper has appeared in the proceedings of the CoopIS'00 Conference (Vonk et al. [44]).

On the one hand, organizations shift their focus to apply workflow management for primary business processes. On the other hand, organizations are focussing more and more on their core business thereby leaving non-core businesses to other specialized organizations. This requires that organizations cooperate to perform their end-to-end business processes. With the widespread possibilities of electronic communication and data exchange, organizations can conduct their business electronically, called e-commerce or e-business. When a cooperation between organizations is formed, the concept of dynamic virtual enterprises is introduced. Multiple organizations with their own primary processes combine forces in a virtual enterprise for a certain period of time. Afterwards, the virtual enterprise is dismantled again, hence the dynamic characteristic of virtual enterprises [36, 37].

Although a virtual enterprise can consist of any number of organizations working together in any form of cooperation, the scope of the work presented here, is limited to the commonly used consumer/provider service outsourcing paradigm. In this paradigm, an organization can act as a (service) consumer, (service) provider, or both. The (service) consumer outsources part of its business process to another organization that can perform that process, which is the (service) provider. All details of the cooperation between a consumer and provider organization in the virtual enterprise is specified in an electronic contract [31].

This paper discusses a three-level transaction model for cross-organizational workflow management, called X-transaction model, that ensures reliable execution of e-services represented by workflow processes within virtual enterprises, for which a standard transaction model is not sufficient. The X-transaction model offers various rollback modes and rollback scopes that allow flexible rollbacks of both executing or completed processes. Rolling back (part of) processes is based on executing compensating activities that semantically undo the effects of already executed activities. The architecture that supports the X-transaction model consists of a static intra-organizational infrastructure layer and a dynamically generated cross-organizational infrastructure layer. The X-transaction model and architecture are implemented in a demonstrator system on top of IBM's MQSeries Workflow [27].

In the CrossFlow project [23], support for cross-organizational workflow management (also known as inter-organizational workflow management[1]) in dynamically formed virtual enterprises has been developed. The developed model facilitates fine-grained contract-based cooperation, which supports specific cross-organizational workflow requirements. The cross-organizational transaction model, architecture and system presented here are developed in the CrossFlow project of which a more detailed description is given in the next section.

1.1. Structure of this paper

The structure of this paper is as follows. Section 2 presents the context of the work presented here, which is the CrossFlow project. It describes the cooperation support, e-services, electronic contracts and intra- and cross-organizational process models.

The X-transaction model that ensures the reliable execution of (cross-organizational) workflow processes is presented in Section 3. A real-life example of an outsourced process illustrating the application of the X-transaction model is shown in Section 4. Section 5 covers possible extensions to the X-transaction model. These extensions deal with rollback migrations, which increases the insight of the consumer organization in an outsourced process rollback, and multi-party outsourcing, which is an extension to the outsourcing paradigm introduced in Section 2. The architecture to support cross-organizational workflow process executions, including the X-transaction model, is discussed in Section 6, while the implemented demonstrator system is shown in Section 7. A discussion of related work is presented in Section 8. The paper ends with conclusions and future work.

2. The CrossFlow context

The work presented in this paper is part of the CrossFlow project [23]. The goal of the CrossFlow project was to develop the support that is required for cross-organizational workflow management in dynamically formed virtual enterprises resulting in fine-grained, contract based cooperation. Within the project, not only the support for the actual execution of cross-organizational workflow process was covered. The dynamic creation of a virtual enterprise, based on the consumer/provider service outsourcing paradigm, as well as the creation and enforcement of electronic contracts that are agreed upon between the organizations involved in the virtual enterprise, was developed. As such, the cooperation in the virtual enterprise can be seen as a specialized form of e-services, in which provider organizations offer e-services to consumer organizations. These e-services are described by means of workflow processes.

2.1. Cooperation support

To facilitate a successful and smooth cooperation between the organizations in a virtual enterprise, the business processes that are to be performed in the participating organizations must be interconnected. Because the organization boundaries of the organizations are crossed in the combined business process, merely connecting the workflow management systems of the organizations does not suffice, e.g., the terminology and technology used in the separate organizations might be different, or the business details of the organizations may not be disclosed (see also Section 2.4.2).

In the CrossFlow project, an architecture has been developed that handles the specific issues that arise as a consequence of the cooperation between different organizations. Specific cooperation support services have been developed that each deal with a different aspect of the cross-organizational workflow management requirements, e.g. transaction management ensures the reliable execution of the (cross-organizational) business process. Although many different cooperative support services can be envisaged for virtual enterprises, the CrossFlow project covers the following:

- Transaction Management which ensures the reliable execution of intra- as well as cross-organizational business processes,
- Level of Control which allows consumer organizations well specified control over the processes being executed by provider organizations,
- Quality of Service which guarantees that qualitative aspects of the e-service, such as obligations and goals that are agreed upon in the electronic contract, will be adhered to, in close cooperation with the Flexible Change Control cooperative support service, and

- Flexible Change Control which determines the paths to be taken in the process such that the goals set in the contract will be met, in cooperation with the Quality of Service cooperative support service.

The developed architecture allows other cooperative support services to be easily plugged into the system, e.g. automatic remuneration. This paper presents the transaction management cooperation support service, covering the cross-organizational transaction model, the architecture to support the model and the implemented prototype system. Details on the other cooperation support services can be found in other CrossFlow related papers, e.g. [23, 26, 30].

Not every cooperation between the organizations in a virtual enterprise will require all functionality of all possible cooperative support services. Whether or not a specific cooperation support service is required and the way it should be configured to support a specific cross-organizational workflow process execution, depends on the related clauses that can be specified in the electronic contract.

2.2. Electronic contracts

When an organization wants another organization to perform part of its process on its behalf, the organizations first need to be brought together so that they can form a virtual enterprise. Organizations find each other in an electronic marketplace where organizations put their services on offer, or search for services that are offered in the marketplace (business-to-business or B2B e-commerce). Using a matchmaking facility or trader, compatible organizations form a virtual enterprise, the cooperation in which is specified in an electronic contract [25, 26]. Figure 1 shows the contract model as it has been developed in the CrossFlow project. The contract model has a modular structure so that it can easily be extended and/or adapted to specific business domains [31].

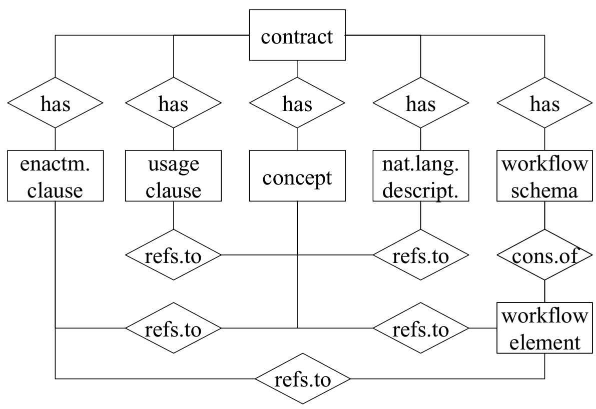

Figure 1. Structure of contract model.

The following sub-models are part of the contract model:

- *Concept model*. All concepts that are used in the contract must be defined explicitly, creating a concept space in which the other contract issues can be specified. This is not dissimilar to the terminology statements in the first section of a regular (paper-based) contract.
- *Process model*. The process model describes the internal structure of the workflow process implementing the service using a WfMC-compliant workflow model [49]. The process is composed of process elements, e.g. the individual activities and control connectors (see also Section 3).
- *Enactment model*. The enactment details must be described. This part of the contract specifies the cooperation support services that are required during the service enactment. For example, the clauses that specify the required cross-organizational transaction support are specified in this part of the contract.
- *Usage model*. The usage model defines manners in which the contract can be established. The simplest case is where one contract is made to start one instance of the service immediately. Other possibilities are contracts made to start multiple executions of the service, or contracts made to reserve the resources of the provider for a service execution at a later moment. The usage model describes the different usage possibilities of the contract and their conditions.
- *Natural language description*. The natural language description is a piece of text that is not meant for electronic interpretation, but for human reading. This piece of text can be used to describe the service in an easily understandable way and to refer to the legal context of the transaction.

The elements of the contract are not independent, see figure 1. Many of the elements can refer to other elements of the model, catering for semantic coupling between the contract elements. As stated before, the electronic contract specifies the e-service. The following subsection describes what an e-service is in the context of the work presented here. After that, the process model for both intra- and cross-organizational business processes is introduced, that form the procedural description of the e-service specified by the electronic contract.

2.3. E-services

An e-service is the electronic equivalent to a regular service offered by some organization. Using an electronic network, usually internet, goods or services are offered to consumers, who will have to perform some counter service (usually paying money) once they accept the e-service on offer. These forms of e-services are usually called web-services, in which case the consumer has no control over or insight in the way the service is performed.

The e-services that are dealt with in this paper are of a different nature. By including the process specification that describes, on a certain abstraction level, the way the process will be performed, the consumer is allowed more insight into the service. When Level of Control is part of the service (see Section 2.1) the consumer is even allowed to perform some control on the service. Adding more cooperative support services, e.g. Quality of Service or Flexible Change Control, creates an even more specialized e-service. Clearly, the complexity of these kinds of e-service is much higher then relatively simple web-services. It is therefore more likely to be used in B2B e-commerce settings, in which organizations cooperate and form virtual enterprises, than in B2C e-commerce settings in which organizations sell services to individual persons.

A more complex form of web-services is presented in [35], in which web-services are seen as cross-organizational processes and are specified in the WSFL language. That approach is similar to our approach in which our process specification in the contract relates to the WSFL and we would therefore rather speak of e-services instead of web-services. The approaches taken in [38] and [17] also use black-box services, in which the service is seen as a monolithic closed process, as opposed to our work, which deals with white-box e-services, in which the service is opened up and e.g. offers monitoring and control to the consumer.

As discussed before, the process specification is included in the contract that forms the e-service, which is required because it provides insight in the e-service to the consumer and the transaction model as discussed in this paper relies on the control flow that is part of the process specification. The next subsection describes the process model used and explains the differences between an intra-organizational process and a cross-organizational process.

2.4. Process models

The business processes of organizations are modelled in workflow process models so that they can be executed by a workflow management system (WfMS). This subsection first describes how intra-organizational business processes are modeled. Then, a cross-organizational process model in which the cross-organizational processes are specified is presented. Both process models are illustrated in Section 4, including transactional aspects introduced in Section 3, using an example scenario.

2.4.1. Intra-organizational process model. To apply workflow management, the business processes of an organization must be modeled in workflow process models. A workflow process model consists of the *activities* that must be performed and the *order* in which those activities must be performed. The order between the activities is specified using different kinds of *control connectors* and the entire ordering of activities is called the *control flow* of the workflow process model. With the control connectors, it is possible to specify that the execution of activities must be done in sequence, or in parallel, or that a choice has to be made between activities (alternative activities), or that certain activities need to be executed more than once (iterative activities) [49].

Business processes usually are complex in nature, consisting of numerous activities and a complicated control flow. To reduce the modeling complexity of such business process, it is possible to model the business processes in a hierarchical manner, in which activities can be refined into smaller, more detailed, activities or grouped into coarser grained, less detailed, activities. The process of refinement or hierarchical decomposition results in a nested workflow process structure, consisting of *basic activities*, which are the activities that are actually executed, and *subprocesses*, which are activities that are not actually executed, but consist of other basic activities or subprocesses.

When organizations form virtual enterprises, the business processes of the separate involved organizations need to be interconnected. The intra-organizational process model, as described here, does not suffice to model the combined business process in a workflow process model, because the specific aspects of cross-organizational workflow processes, e.g. autonomy of organizations and process encapsulation, cannot be specified. However, those specific cross-organizational aspects can be modelled in a cross-organizational process model as presented in the next section.

2.4.2. Cross-organizational process model. In a cross-organizational setting based on the consumer/provider service outsourcing paradigm, two[2] organizations are involved in the execution of the cross-organizational workflow process.

The service provider organization executes a workflow process on behalf of a service consumer organization. However the service provider organization does not want to disclose all details of the workflow process it executes. To encapsulate the details of the process, the provider only discloses those aspects of the process it is willing to make publicly known and that are of interest to possible consumer organizations.

On the other hand, most consumer organizations would want to integrate a white-box view of the outsourced process into their own process, which gives the consumer organization more information on the structure and progress of the process that is being executed by another organization. However, the consumer organization does not want to know the specific details of that workflow process, but would rather have an abstract view of it.

By creating a common view of the outsourced process, the wishes of both organizations can be satisfied. The common view encapsulates the details of the process for the provider and presents the consumer with an abstracted view of the process, including the (abstracted) structure of the process, so that it can be integrated in its own process.

As this common view process specification has to be agreed upon by the consumer and provider organization, it is included in the electronic contract that has been established between the organizations in the virtual enterprise, and is therefore called '*contract level*' process. The encapsulated process, i.e. the detailed process as it will actually be executed by the provider organization is called the '*internal level*' process and is thus not visible to the consumer organization. The process that is executed by the consumer organization is called the '*outsourcing level*' process. Figure 2 shows an example cross-organizational process and serves to illustrate the three levels mentioned above. Details on the process itself will be explained in Section 4.

The activities or subprocesses that are part of the contract level are specified by the provider organization, as those activities need to be performed in the provider organization. It is thus the provider organization that determines how much detail of his process will be shown in the contract. In case no detail is presented, the entire provider process is represented by one activity in the contract. If the provider discloses all details of the process, the process specified in the contract matches one-to-one on the internal provider process.

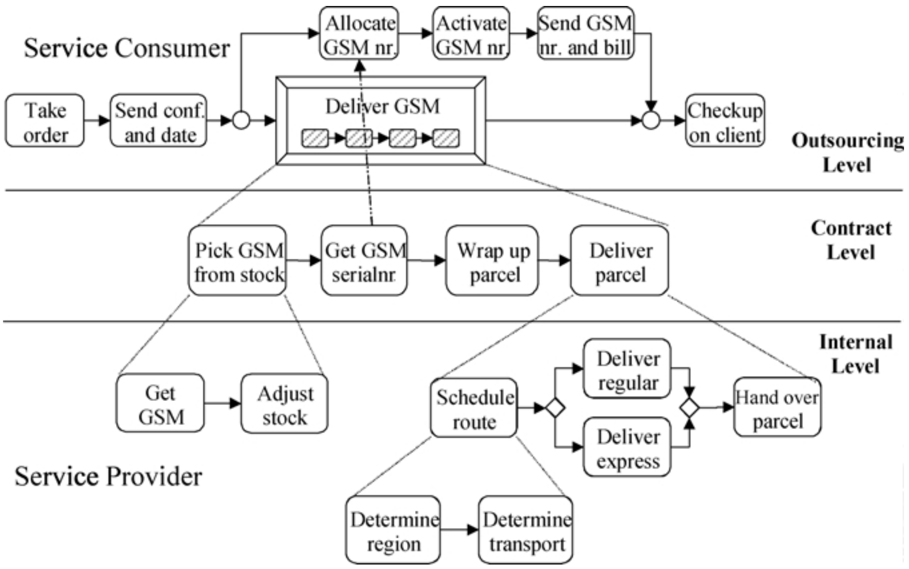

Figure 2. Cross-organizational process example.

All variations in detail disclosure between the afore mentioned extremes are possible. Note that the activities or subprocesses specified in the contract can exist on any refinement level of the internal provider process, which partitions the entire provider process in half. One half of the process is disclosed to a consumer and the other half is encapsulated and thus hidden to the consumer, i.e. an activity or subprocess is either specified in the contract or encapsulated, but not both.

Integrating the outsourced process within the consumer workflow process is possible through the placeholder concept. The *placeholder* is a special kind of activity within the consumer workflow process, representing a subprocess that is executed under the control of another organization and on another WfMS, i.e. the outsourced process, or e-service as performed by the provider organization. The placeholder can act as a black-box or as a white-box. In the former case, the one placeholder activity represents the entire outsourced process. In the latter case, the outsourced process is ‘shadowed’ within the consumer process, which means that the execution progress of the outsourced process by the provider organization is shown (or shadowed) within the consumer workflow process itself, but the actual execution of the activities is performed within the provider workflow process. The example process in figure 2 and in Section 4 illustrates this.

3. The X-transaction model

Integrating transaction management support into workflow management systems provides for reliable and consistent process executions. The traditional flat transaction model originating in the database community that ensures the ACID transaction properties, is however too strict for the inherently long-running workflow processes. Cross-organizational workflow processes involving autonomous organizations require transaction support to deal with cross-organizational aspects as well. The transaction model presented in this section deals with both issues.

This section first introduces a compensation-based transaction model that provides for transaction support in intra-organizational workflow processes. Instead of inventing yet a new transaction model from scratch, we have taken the existing global transaction model [22, 24] as a basis. Then, the specific requirements for cross-organizational transaction support are presented after which the cross-organizational transaction model, called X-transaction model, is described.

3.1. Intra-organizational transaction model

Various advanced transaction models [16, 28, 32] have been proposed to overcome the problems related to the long-livety of intra-organizational workflow processes by relaxing the atomicity and isolation constraints. The global transaction model taken as a basis for the work presented here, is in turn based on the saga transaction model as proposed in [18] which relaxes the atomicity and isolation constraints. The global transaction model extends the saga transaction model with support for cycles in process specifications and the safepoint concept that allows flexible process rollbacks [22, 24].

Long-running workflow processes consist of smaller, relatively short running, process steps that commit the results after the step completes. These steps are part of an intra-organizational workflow process, and are called I-steps. *An I-step* is thus an atomic piece of work that adheres to the ACID transaction properties.

Each I-step has a compensating counterpart specified for it that semantically undoes the effect of the original I-step. In case of failures, a compensating process is dynamically created which rolls back the failing process execution by executing the compensating I-steps in the reverse sequence in which the original I-steps have been executed. If an I-step does not have a compensating activity specified, the compensation of that activity is skipped, as it was apparently impossible or not necessary for the process designer to specify a compensation for the activity.

Marking an I-step as a safepoint indicates that a rollback could be stopped at that step, because a consistent state, from a business point of view, in the process has been reached. This means that every committed I-step executed after the safepoint is compensated, but not the safepoint itself. Whether the rollback actually stops at those safepoints is determined by the compensation algorithm and depends mainly on the complexity of the control flow between the executed activities. A formal description of the compensation algorithm is presented in [21, 24].

Whenever a rollback is required during process execution, e.g. because an activity fails, an abort request is issued to the transaction management system. In the abort request the *rollback mode* is specified, which is either partial rollback mode or complete rollback mode. In the first case, the rollback will compensate or undo the process until a suitable (set of) safepoints is encountered. In the latter case, the entire process execution will be compensated. The rollback mode offers the users of the workflow management system a flexible way to rollback processes.

3.2. Cross-organizational transaction requirements

Transaction support for cross-organizational workflow management must satisfy additional requirements imposed by the autonomy of the involved organizations. When autonomous organizations participate in a tight cooperation within a virtual enterprise, they want to preserve their autonomy as much as possible. This rules out the use of one central transaction system that governs the transactional behavior over the involved organizations using for example a two-phase commit protocol as is common in multi-database environments. In such a protocol, the organization that wants to commit its results must wait until the other organization is ready to commit its results as well and the global transaction support system signals that the commit can be executed. Obviously, such a protocol seriously reduces the autonomy of the involved organizations, which gets even more severe if more parties get involved in the consumer/provider service outsourcing paradigm. Thus, to preserve the autonomy of the involved organizations, cross-organizational workflow processes require decentralized transaction support that offers relaxed (or loose) transaction properties.

The intra-organizational transaction model described in the previous subsection can be applied to the contract level workflow process, because the contract level workflow process is also a long-running process and also requires loose transaction properties. Similar to dividing the intra-organizational processes in smaller steps (I-steps), the contract level process is divided into smaller steps that each commit their results when the step finishes and are compensated in case they need to be undone. Because these smaller steps relate to cross-organizational workflow processes these steps are called X-steps and because the contract level activities encapsulate the internal level activities, an X-step corresponds to one or more I-steps. So, committing the result of an X-step is in fact done through the underlying I-steps, and *an X-step* thus has relaxed atomicity characteristics. An X-step is not executed in isolation because the intermediary results produced by the underlying I-steps are committed even though the X-step itself is not yet completed.

To be able to undo these X-steps in case of a rollback, each X-step must have a corresponding compensating activity specified for it. The X-steps correspond to the process that is specified in the contract, i.e. the contract level, and are executed by the provider organization. Therefore, it is the provider organization that has to specify these contract level compensating activities and to inform the consumer about how the outsourced process will be compensated by the provider, those activities will also be specified in the contract.

From the issues described for the intra-organizational transaction model, the cross-organizational transaction requirements and the cross-organizational process model described in Section 2.4, it follows that a cross-organizational workflow process consists of three levels that have transactional semantics. These three levels are:

1. *The outsourcing level*. The entire workflow process of the consumer organization on the level of I-steps. The outsourcing level is only visible to the consumer organization. Note that the placeholder, i.e. the activity in the consumer process that represents the outsourced process, is a normal I-step and must therefore have a compensating activity specified for it.

2. *The contract level*. The X-steps as they are specified in the contract. All X-steps at the contract level encapsulate the more detailed, internal activities of the provider process. The contract level is visible to both consumer and provider organizations.
3. *The internal level*. The entire workflow process steps of the provider organization on the level of I-steps. The internal level is only visible to the provider organization.

To offer transaction support for cross-organizational workflow processes, the transaction model must be able to handle the above mentioned three transactional levels.

3.3. Cross-organizational transaction model

The cross-organizational transaction model, called the *X-transaction model*, combines the three transactional levels described in the previous subsection, consisting of the I-steps and X-steps, into one transaction model, i.e. a three-level transactional workflow process model. The X-transaction model offers the required loose transaction properties for the intra- as well as cross-organizational workflow processes. An X-transaction consists of all X-steps and I-steps of the cross-organizational workflow process, together with the corresponding compensating steps.

Similar to the possibility of specifying I-steps as safepoints, it is also possible to specify X-steps as safepoints. This way, the flexibility in rollback handling offered by partial rollbacks is offered at the contract level as well.

The X-transaction model offers a flexible rollback mechanism that allows rollbacks to take place at any of the three different transactional levels:

1. *Outsourcing level.* A rollback on the outsourcing level is performed entirely by the consumer organization. If the outsourced process needs to be compensated, the compensating activity corresponding to the placeholder is executed. This compensating activity of the placeholder is specified by the consumer, because it represents the outsourced process within the consumer process itself, and does not necessarily involve the provider organization that has actually executed the outsourced process. Just like any other I-step, the placeholder can also be specified as a safepoint.
2. *Contract level*. A rollback on the contract level will involve only the X-steps of the cross-organizational process by executing the compensating activities that correspond to those X-steps. Note that the contract level X-steps and their compensating counterparts are specified in the contract and are therefore visible to the consumer and the provider organization. A rollback on this level requires that the compensating X-steps are mapped to the underlying compensating I-steps.
3. *Internal level*. A rollback will only involve I-steps that are internal to the provider process and are thus not visible to the consumer.

Similar to the rollback mode, it is possible to indicate a *rollback scope* when a rollback request is issued. The rollback scope indicates whether the rollback is intra-organizational or cross-organizational. The first means that the rollback will only be performed in the organization that issues the rollback, the latter means that the rollback will also be performed in the other organization. When a rollback is required in which the rollback scope is set

Table 1. Rollback mode and rollback scope combinations.

Rollback: Scope and mode	Rollback starts at: Consumer organization	Provider organization
Intra-organizational and complete	Entire consumer process is rolled back using the compensation of the placeholder.	Entire provider process is rolled back.
Intra-organizational and partial	Only the consumer process is rolled back until a safepoint is found. If the outsourced process must be rolled back it is done by compensation of the placeholder.	Only provider process is rolled back until a safepoint is found.
Cross-organizational and complete	The consumer process is rolled back in its entirety. The outsourced process is rolled back by the provider.	The provider rolls back its own process in its entirety. After that, the consumer must rollback in partial rollback mode starting at the placeholder (which is the failing activity for the consumer).
Cross-organizational and partial	The consumer process is partially rolled back. If the outsourced process must be must be rolled back it is rolled back by the provider.	Not possible: Either the process is rolled back partially to a safepoint in the provider process (which is thus intra-organizational) or the rollback is cross-organizational and thus the provider process is rolled back completely.

to cross-organizational, the rollback can involve a combination of the three transactional levels presented above. For example, if the consumer starts a rollback (cross-organizational rollback scope), the rollback will involve the outsourcing level and also the contract level. The combination of different rollback modes and rollback scopes in a rollback request determines the effects of the rollback execution, which are presented in the Table 1.

When a rollback takes place at the internal level of the provider organization, the consumer organization can be aware of it, e.g. using a monitoring mechanism, but the consumer is not able to trace the compensation, because the details of the internal process are not visible to the consumer. The process will be restarted after the compensation is finished, so the consumer will see that part of the outsourced process is executed a second time. Section 5 presents an extension to the X-transaction model that offers more flexibility in the execution of compensating activities by migrating rollbacks over the different transactional levels. This also offers the consumer organization more insight in rollbacks that are being performed within the provider organization.

The next section illustrates the X-transaction model and operational semantics using a real-life business scenario.

4. Transactional cross-organizational process example

This section presents a real-life business scenario[3] in which an organization offers its business process as an e-service, and forms, with another organization, a virtual enterprise. The resulting cross-organizational process, is supported by the X-transaction model.

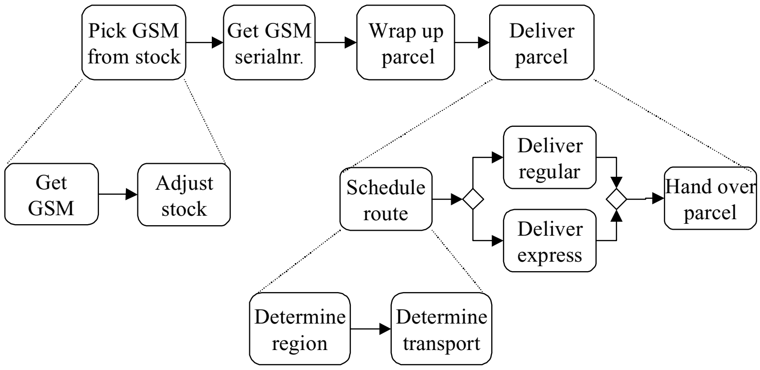

Figure 3. Logistics process.

4.1. From business process to e-service

The scenario consists of an intra-organizational business process, which is modeled as a three-layered nested workflow process, is shown in figure 3. It illustrates the business process of a logistics organization that delivers cellular phones (GSM phones) from a warehouse to a customer. The process starts by picking a GSM phone from the stock. When the stock is updated, the GSM phones serial number is retrieved and linked to a specific phone number (not shown in the figure), after which the phone is put in a parcel. The parcel is delivered to the customer by first scheduling a route, which depends on the region the customer lives in and the kind of transport required, second it is decided whether the parcel needs to be delivered by regular or express mail. At the end of the process, the parcel is actually handed over to the customer. In the remainder of this paper, the cross-organizational workflow aspects, with the emphasis on transaction support, will be illustrated using this process.

In the figure, the rounded rectangles represent activities. The arrows and diamonds together represent the control flow. The arrows link the control connectors to the activities. If no control connector is specified between activities, the arrows represent a sequential execution order between activities. The diamond is a control connector representing an OR-split, meaning that one of the succeeding activities can be executed, or an OR-join, meaning that the following activity is executed whenever one of the preceding activities has been completed. The dotted lines represent hierarchical decomposition, e.g. the ‘Deliver parcel’ subprocess is decomposed in one subprocess and three activities.

Although the control flow in this example is relatively simple, the full set of control connectors as specified in [49] and as offered by most commercial workflow management products is supported in CrossFlow intra- and cross-organizational process models.

Suppose the logistics organization discovers that their process could be interesting for other organizations (most likely telecom companies) and decides to offer their process as an e-service in an electronic marketplace. The logistics organization thus wants to act as a service provider organization, but is willing to show only the top four activities (or subprocesses) of the process, which will constitute the contract level process. The other activities are encapsulated by the contract level and form the internal level, which is thus not visible to a consumer organization.

4.2. From e-service to cross-organizational process

When some organization requires an e-service because it wants to outsource part of its process, it will search for a provider organization and, when a suitable one is found, it will form a virtual enterprise (possibly after some negotiations) and together they will perform the cross-organizational process.

The business process introduced above is further elaborated in this section. A virtual enterprise has been established by, and consists of, a telecom company acting as the service consumer organization and a logistics company acting as the service provider organization. The two organizations have agreed upon an electronic contract in which the cooperation, including the outsourced process on the contract level, is described.

The cross-organizational process of the virtual enterprise is shown in figure 4. The telecom organization takes orders from its clients to sell GSM phones. After the order is received, a confirmation is sent to the client together with an estimate of the delivery date. Then, two activity branches proceed in parallel represented by the circle control connector, which is either an AND-split (more than one outgoing arrow) or an AND-join (more than one incoming arrow). One parallel branch continues the execution of the consumer process, the other parallel branch consists of the outsourced process (shown as a double lined rectangle, i.e. the placeholder, as explained in the previous section). Note that the placeholder shadows the common view of the process inside the consumer workflow process. In the provider organization, the GSM phone is taken from stock and the serial number of the GSM phone is sent to the consumer using the monitoring mechanism (indicated by the dashed arrow). The consumer can then allocate a telephone number to the serial number, activate it and send the telephone number together with the bill to its client. Parallel to those consumer activities, in the provider process the GSM phone is wrapped up in a parcel and delivered to the client, which ends the outsourced process. As a last activity in the cross-organizational process,

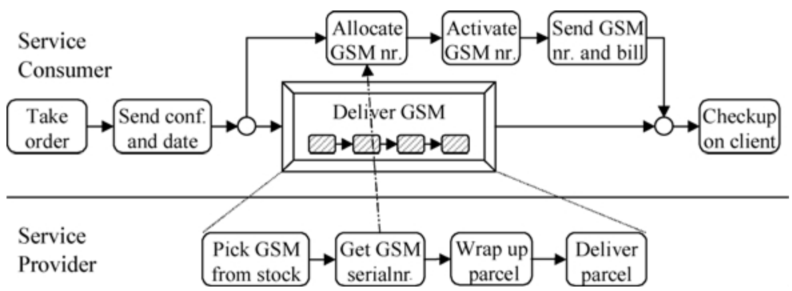

Figure 4. Cross-organizational process with outsourced logistics process.

the telecom company checks up on the client for marketing and customer satisfaction purposes.

The detailed provider process of this scenario is the process presented in Section 4.1. As can be seen from figure 3 and figure 4, the common view of the process, or contract level, consists of the top four activities of the provider process. The outsourced process in this example has a simple sequential structure, but can be arbitrarily complex in more elaborate business processes.

The next section illustrates how the X-transaction model can be used to guarantee reliability in case of failures during process execution that requires rolling back (part of) the executed process.

4.3. Rollback examples

To illustrate the effects of different rollback modes and rollback scopes in a rollback request, the cross-organizational process introduced above is used. Figure 5 shows the state of the X-transaction of the process at a certain point in time, i.e. the process execution history. The process is being executed and has progressed to the activities 'Send GSM nr. and bill' and 'Deliver GSM' at the consumer organization, and to activity 'Wrap up parcel' at the provider organization. This means that those three activities are still running and all preceding activities have finished.

The thick-lined rectangles in the figure represent the safepoints that are specified in the process. Compensating activities are not shown in the figure, as they are not part of the normal control flow of the process, however, every activity in this scenario has a compensating activity specified for it.

For a first rollback example, suppose that the running activity at the provider ('Wrap up parcel') fails, as indicated by the crossed-out rectangle in the figure, because it is discovered during packaging that the GSM phone is not the correct model. In this case, the provider will start a rollback to bring its process into a consistent state. To reach a consistent state, the entire cross-organizational process needs to be rolled back, thus the rollback mode is complete. The consumer and provider processes are closely linked, because the GSM number is linked to the GSM serial number. Therefore, it is stated in the contract that the provider can only issue a rollback with complete rollback mode if the rollback scope is

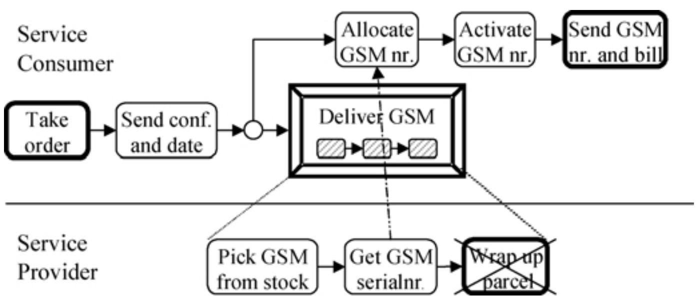

Figure 5. Rollback example.

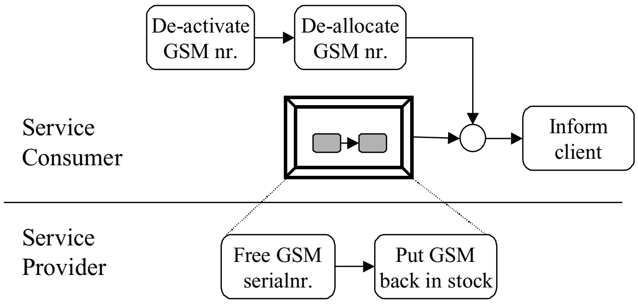

Figure 6. Example compensation process.

cross-organizational, which implies that the consumer process will also be rolled back (in partial mode, see Table 1).

The resulting compensation process, consisting of the compensating counterpart activities, is shown in figure 6.

Parallel to the rollback at the provider, which frees the GSM serial number and puts the GSM phone back in stock, the consumer de-activates and de-allocates the GSM number. After that, the consumer informs its client that there is a delay in the delivery of the GSM phone (the compensating activity of ‘Send conf. and date’, which is ‘Inform client’, and the original process is restarted again with activity ‘Send conf. and date’.

As a second rollback example, suppose that the last activity ‘Checkup on client’ in the cross-organizational workflow process of figure 4 fails, which requires the consumer to rollback the process. In this situation, the consumer has the choice to involve the provider in the compensation (if the contract has not expired yet and it is stated in the contract that provider side rollbacks are allowed) or to compensate the process by itself, using the compensating activity of the placeholder. In the first choice, the rollback scope is cross organizational and the provider will pick up the GSM phone, unwrap the parcel, frees the GSM serial number and puts the GSM back in stock. In the second choice, the rollback scope is intra-organizational and the consumer executes, in parallel to its own compensating activities (de-activating and de-allocating the GSM number), also its own compensating activity of the delivery of the GSM, i.e. execute the compensating activity of the placeholder, which requires the client to send back the GSM phone. Depending on the reason the activity has failed, the consumer can decide for the first option or the latter.

5. Extended X-transaction model

The X-transaction model of Section 3 provides the basic transactional functionality to support cross-organizational business processes. This section presents two extensions to the basic X-transaction model. The first extension increases flexibility and usability through the use of rollback migrations. The second extension increases applicability, by supporting multi-party outsourcing instead of the more limited consumer/provider service outsourcing paradigm.

5.1. Rollback migrations

Supporting rollback migration allows compensating activities to be specified for all workflow activities or subprocesses and not only for the basic activities that are actually executed, see Section 2.4. A subprocess is only a hierarchical grouping of its comprising activities and has therefore the same business semantics as the combined activities it groups together. Similar to that, a compensating subprocess has the same business semantics as all compensating activities that belong to it. It is thus possible to execute the compensating counterpart of a subprocess if all activities that belong to the subprocess need to be compensated and the rollback of a group of activities is then migrated to a higher level (the compensating subprocess). This contrasts to the nested saga model [20], in which sagas can be nested and the rollback is propagated through parent and child sagas but the actual compensation is still performed by executing the compensating activities corresponding to the basic activities (the leaf activities in the nested saga tree).

As an example, suppose the activities ‘Print letter’, ‘Print invoice’ and ‘Wrap parcel’ of the process shown in figure 7 must be compensated. These activities together form the subprocess ‘Wrap up parcel’. The compensating counterpart of ‘Wrap up parcel’ has the same business semantics as the combined compensating counterparts of the three separate activities. It is therefore possible to execute the compensating counterpart of ‘Wrap up parcel’ to reach the same effect as executing the three compensating counterparts of the activities. The rollback has then been migrated to the higher level subprocess.

In the extended X-transaction model, the service consumer organization is offered more insight in the provider compensation process as a result of the rollback migration. In this case, the internal level compensation can be migrated to a contract level compensation. This means that, if all internal level I-Steps belonging to one contract level X-step need to be compensated, the compensation of that contract level X-step has the same effect and can therefore be used instead of the compensating internal level I-steps. Because the compensating activity of the X-step is also specified in the contract and acts as a black-box activity encapsulating the internal level activities and their compensating counterparts, the consumer can trace the compensation process while the provider is executing it, which would be impossible if the compensation would only take place on the provider internal level.

In the same example as above, the consumer does not know about the three internal level activities (‘Print letter’, ‘Print invoice’ and ‘Wrap parcel’), but the activity ‘Wrap up parcel’

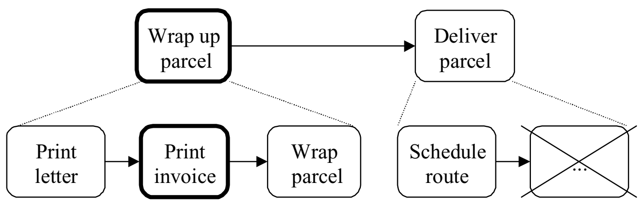

Figure 7. Safepoint constraint violation.

is visible to the consumer. Instead of compensating the three internal level activities, the same effect is reached when the contract level activity is compensated and the consumer is able to view the compensation.

Although rollback migration increases the flexibility in rollback executions, it reduces the freedom in marking activities as safepoints. A consistency constraint exists between a subproces safepoint and its comprising lower level activity safepoints (or contract level safepoints and internal level safepoints in a cross-organizational setting, respectively). It is only possible to mark a subprocess as a safepoint if the last activity of that subprocess is a safepoint as well. If that is not the case, the business semantics of the subprocess and the combined activities that belong to it are different, and thus the rollback migration is not allowed.

For example, the thick-lined activities ('Wrap up parcel' and 'Print invoice') in figure 6 are safepoints and the activity after 'Schedule route' fails. Without rollback migration, a partial rollback will have to undo activities 'Schedule route' and 'Wrap parcel'. A migrated rollback will only undo 'Schedule route', because 'Wrap up parcel' is a safepoint. Both rollback possibilities are different and will have a different effect on the business process if executed. Marking the subprocess 'Wrap parcel' as a safepoint should thus not be allowed in this case.

5.2. Multi-party outsourcing

The paper has so far dealt with service outsourcing between two parties, the service consumer and service provider. This section presents the extensions to the transaction model to accommodate for more than two parties participating in virtual enterprises. It requires an extension to the rollback scope, so that a cross-organizational rollback can also identify the involved organization(s) that needs to roll back.

We base multi-party virtual enterprises on a combination of bilateral service outsourcing relations. Multi-lateral contracts increase the complexity of the process model and transaction model severely in the presence of complex and multi-lateral rights and obligations that can be specified in such contracts and is outside the scope of this paper.

We have identified two different types of multi-party outsourcing: chained multi-party outsourcing and independent multi-party outsourcing. Both are presented in the next two subsections.

5.2.1. Chained multi-party outsourcing. In the *chained multi-party outsourcing* setting, it is possible for an organization to take the role as service consumer as well as the role of service provider. This means that if a provider organization performs a process on behalf of a consumer organization, it can again outsource part of that process to another organization and thus become a consumer organization as well. As the entire cross-organizational process resembles a chain of outsourced process, it is called chained multi-party outsourcing.

For every link in the outsourcing chain a separate contract is created between the consumer and provider organizations. So, an organization will only be aware of its direct consumer organization and direct provider organization. It is the responsibility of an organization to

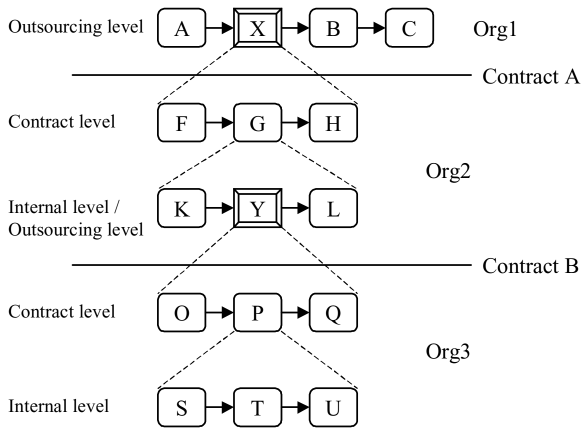

Figure 8. Chained outsourcing.

make sure that it complies with all the rights and obligations that are specified in the one or two contracts that the organization is involved in.

An example of a chained multi-party outsourcing process is presented in figure 8. In this example three organizations are involved in the virtual enterprise. Org1 outsources part of its process to Org2, which in turn outsources part of its process to Org3. The cooperation between the organizations is specified in electronic contracts, i.e. contract A and contract B in the example.

Two issues arise when multiple organizations are participating in a multi-party outsourcing setting. The first is the identification of organizations within the cooperation and the second is the possibility of concurrent rollback requests.

In the case that the rollback scope is set to cross-organizational in a rollback request, it means that the *other* organization is involved in the rollback as well, as explained in previous sections. In a multi-party setting, the other organization is only uniquely identifiable when the issuing organization has either the role as consumer or role as provider, but not both. If an organization has both roles within the same cross-organizational process, it is the responsibility of that organization to unambiguously specify the rollback scope. This is accomplished by extending the rollback scope with an extra parameter that is used to indicate whether the rollback should proceed up, or down or both up and down the outsourcing chain.

In the example of figure 8, suppose that activity ‘L’ of the running process in Org2 fails. It is then possible to start a rollback that is cross-organizational and effects Org1 by compensating activity ‘A’ or Org3 by compensating activities ‘U’, ‘T’, and ‘S’, or both organizations.

When more than two organizations are participating in a virtual enterprise and are executing parts of the cross-organizational workflow process at the same time, the chance that concurrent rollback requests occur increases, in which case more than one, possibly contradicting, rollback requests must be handled by the transaction management support system. This concurrent rollback request problem has already been dealt with in the WIDE project, and the cross-organizational transaction model and architecture presented in this paper can be easily extended with the ideas and algorithms presented in [43].

In any multi-party contract, but especially in chained multi-party outsourcing, where one organization must comply with multiple contracts, it must be ensured that the rights and obligations in the contracts do not contradict each other, i.e. the different contracts of one organization must be consistent. For example, suppose that in one contract it is specified that a rollback of the cross-organizational process is allowed, while in another contract it is specified that the cross-organizational process may not be suspended. These two agreements in the contract contradict each other, because the process must be suspended before a rollback can be handled, see [24]. As this subject is not specific to transaction, it is outside the scope of this paper. Inconsistencies in contracts are less likely to happen in the independent multi-party outsourcing scenario, because an organization in that case will always act in one role, i.e. an organization is either consumer or provider, but not both.

5.2.2. Independent multi-party outsourcing. In the independent multi-party outsourcing setting, the organization acting as service consumer can outsource different parts of its process to multiple provider organizations. A different electronic contract is created between the service consumer and every provider organization. The outsourced processes have no dependencies between them and the different provider organizations are not aware of each other, which is why this scenario is called independent multi-party outsourcing. As every electronic contract involves the service consumer, inconsistencies between the different contracts are less likely to happen and are more easily avoided because the consumer organization has knowledge of all other contracts that are created within this virtual enterprise.

An example of an independent multi-party outsourcing scenario is shown in figure 9, in which three organizations are involved. Parts of the business process running in Organization Org1 are being outsourced to organizations Org2 and Org3. Separate contracts are created between Org1 and Org2, i.e. Contract A and between Org1 and Org3, i.e. Contract B.

The same two issues as described in chained multi-party outsourcing apply to independent multi-party outsourcing, i.e. identification of the participating organizations and concurrent rollback requests.

In the multi-party independent outsourcing virtual enterprise configuration, the responsibility of the rollback scope lies with the service consumer organization. Any service provider

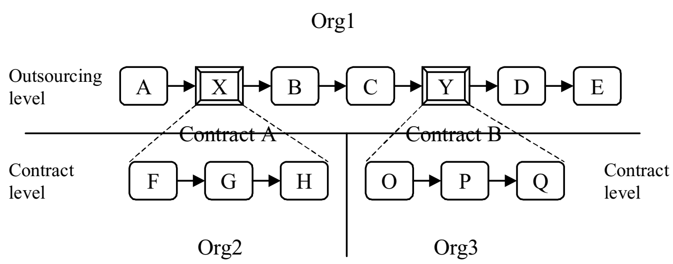

Figure 9. Independent outsourcing.

organization only has one direct service consumer, so for the provider this configuration is not different from the two-organization consumer/provider outsourcing setting. The service consumer organization, however, has to be able to uniquely identify its service provider organizations (and the contracts that correspond to it). Extending the rollback scope with the organization identifier parameter, a vector based parameter that can hold the organization identifications that are involved in the rollback request) is sufficient to provide this functionality, similar to the chained multi-party outsourcing solution.

For example, suppose activity ‘E’ in figure 9 fails and a cross-organizational rollback needs to be performed to reach a consistent process state again. The consumer organization (Org1) now has the choice to:

- rollback the process executed at Org3 and undo the process of Org2 by executing the compensating activity that corresponds to placeholder ‘X’, or
- Undo the process of Org3 by executing the compensating activity that corresponds to placeholder ‘Y’ and rollback the process executed at Org2, or
- rollback both processes, or
- rollback none of the processes by executing the compensating activities that correspond to both placeholders ‘Y’ and ‘X’.

As described for the chained multi-party outsourcing setting, concurrent rollback requests can also occur in independent multi-party outsourcing when multiple provider organizations contribute to the consumer process in parallel. The same solutions can be applied here.

The next section presents the architecture to support the cross-organizational transaction model.

6. Architecture

Support for the cross-organizational processes within a virtual enterprise, as discussed in this paper, requires a flexible and dynamic architecture to provide for the specific functionalities related to the cross-organizational setting. First, the basic architecture is explained that supports process outsourcing, i.e. the process model as described in Section 2.4. This basic architecture is then extended by adding the specific components that implement the X-transaction model.

6.1. Basic architecture

Support for cross-organizational workflow management requires more than just connecting the workflow management systems of the involved organizations, for which a standard has been specified by the workflow management coalition, called interface 4 [48]. In any cross-organizational setting, organizations are involved that each have their own business rules and business culture and want to preserve their autonomy as much as possible. How much autonomy can be preserved for an organization is determined by the outcome of negotiations that take place before an electronic contract is signed and the cooperation is started, see Section 2. Specific clauses in the electronic contract can strengthen or weaken the autonomy of an organization.

As explained before, the involved organizations have a common view of the outsourced workflow process on the contract level. The electronic contract contains all specific information necessary to execute the cross-organizational process including the rights and obligations of both organizations, which must be enforced by the supporting cross-organizational workflow management architecture. As the specific cooperation requirements, the rights and the obligations of the involved organizations differ per contract, a flexible architecture is required that can be configured such that the agreements specified in the electronic contract can be enforced.

Considering the topics mentioned above combined with the different process levels mentioned in Section 2.4, it follows that the basic architecture has three layers. The bottom layer deals with the internal workflow processes of the consumer and provider organizations and consists of the individual WfMSs of those organizations. The top layer deals with the aspects and requirements related to the cross-organizational execution of the workflow processes, i.e. the contract level workflow process. This architectural layer is contract specific and only required for the time that the contract is valid. Therefore, it will be dynamically created when necessary. The middle layer is an isolation layer that encapsulates the specific details of the bottom layer from the cross-organizational aspects of the top layer and provides for the mapping between those other two layers, using the *internal enactment specification* (IES). The IES describes how the internal process and resources of the organizations relate to the workflow process and enactment clauses (see also Section 2.2) as specified in the electronic contract. The isolation layer makes it therefore possible to change the underlying workflow management system with little effort; only the parts of the isolation layer that relate to a specific workflow management system need to be modified.

The basic architecture is shown in figure 10. The upper half of the architecture is the dynamic cross-organizational infrastructure. It is configured by the *Contract & Configuration*

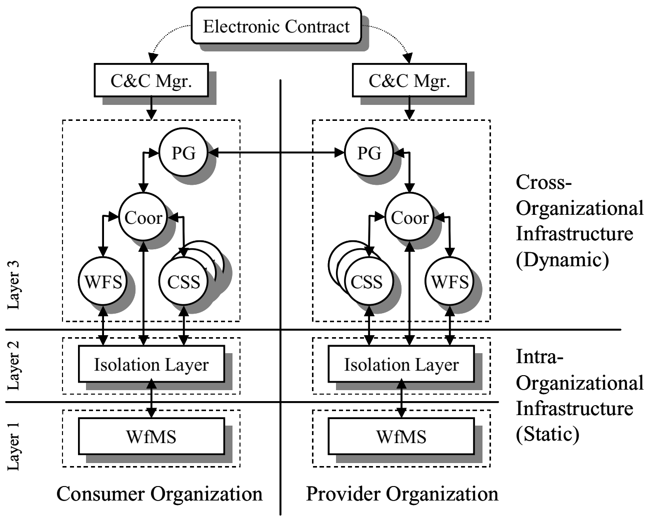

Figure 10. Basic cross-organizational WfM architecture.

Manager (C&C Mgr.) using the agreements specified in the electronic contract, as indicated by the dotted arrows, and is dismantled when the contract expires. Because the rights and obligations of the consumer and provider organization are usually different, the cross-organizational infrastructure will be configured differently for both organizations.

The cross-organizational infrastructure consists (per organization) of a coordinator (Coor), a proxy gateway (PG), a workflow state module (WFS) and one or more *cooperative support services* (CSSs) modules. The CSSs provide the dedicated support for different aspects of the cross-organizational workflow process execution. The proxy-gateways provide a communication mechanism to handle all communication between the involved organizations and provide, at the same time, the security mechanisms to protect the organizations. All communication related to the cross-organizational process flows through the coordinator. The WFS is a module that registers activity and process state changes at the internal level and maps these to the contract level, so that the contract level and internal level reflect the same state.

CSS modules register themselves to the coordinator and indicate what information or messages they expect. The Coordinator then acts similar to a switchboard and passes the incoming messages to the correct CSS module. This provides for a flexible architecture in which new CSS modules can be simple 'plugged in' and then used in the system, e.g. a dedicated renumeration CSS could be implemented and plugged in, so that renumeration in the virtual enterprise could be supported. Communication between the participating organizations takes place through the proxy gateways. They contain the mechanism to handle the aspects related to crossing organizational boundaries, e.g., security aspects.

The bottom half in the figure is the intra-organizational infrastructure. It consists of the local workflow management system (WfMS), which executes the processes of the organization, and the isolation layer that shields the cross-organizational infrastructure from WfMS specifics. The *isolation layer* maps the WfMS independent entities that are specified in the electronic contract to the WfMS specific entities and vice versa using the IES. For example, the contract level process is mapped to the internal level process and the parameters specified in the contract are mapped to the parameters used by the specific WfMS. This ensures that the cross-organizational infrastructure can be applied in a heterogeneous environment, because it is independent of the underlying WfMS platform, which can and usually will be different for each organization.

6.2. Cross-organizational transactional architecture

The architecture that supports the X-transaction model is shown in figure 11. The cross-organizational infrastructure of figure 10 has been expanded with the specific modules that facilitate cross-organizational transaction support. The generic CSS modules shown in figure 10 have been replaced by the specific cross-organizational transaction support CSS modules. They are created and configured by the Contract & Configuration Manager (C&C Mgr.) according to the agreements that are specified in the electronic contract. In addition, the static intra-organizational infrastructure layer has been extended with a module that provides for the intra-organizational transaction support, i.e., the ITM.

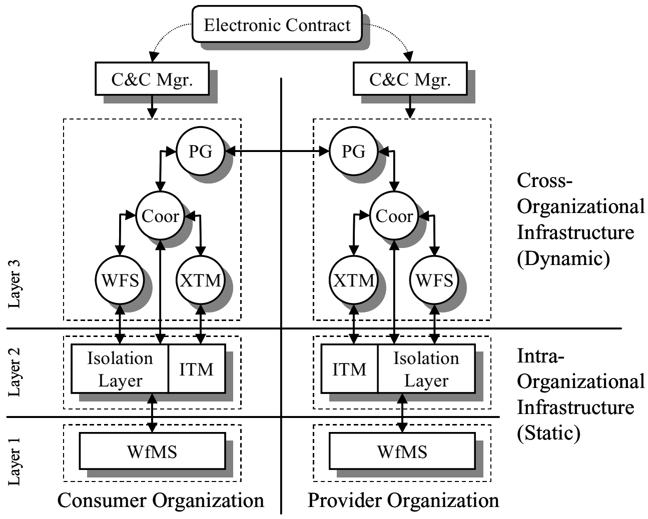

Figure 11. Transaction architecture.

The cross-organizational infrastructure related to transaction support consists of two cross-organizational transaction managers (XTM) and two workflow state modules (WFS), one of each for the consumer organization and one of each for the provider organization. Both the XTM and WFS are configured according to the contract that has been agreed upon by the two organizations. As the rights and obligations are usually different for the consumer than for the provider organization, the consumer side XTM and WFS will thus be configured differently from the provider side XTM and WFS by the C&C Manager.

Because cross-organizational transaction support is only required when the cross-organizational process is actually being executed, the XTM and WFS are only required when the outsourced process starts and are no longer required when the contract expires.

The *XTM* provides the cross-organizational transaction support. It has the necessary algorithms to calculate compensating workflow processes to undo executed X-steps, as discussed in Section 3. To determine which X-steps have been executed, thereby forming the workflow execution history on the contract level, it uses the information provided by the workflow state module (WFS).

The intra-organizational transaction manager (*ITM*) provides transaction support for intra-organizational workflow processes. Currently, it is a separate module on top of the WfMS because advanced transactional functionality required by workflow management, see Section 3, is not yet offered by any commercial WfMS. In the future, the ITM and WfMS might be integrated into one system. The workflow execution history necessary for the ITM to calculate the compensating workflow process, is retrieved from the WfMS. When the compensating workflow process is created by the ITM, it is returned to the WfMS (after making it persistent) so that it can be executed. Formal specifications of the algorithms used in the ITM can be found in [24].

Note that, when a rollback occurs that involves a combination of the three transactional levels as described in Section 3, the compensation process will be computed by a combination of XTMs and ITMs. The entire compensation process will consist of multiple separate compensation processes, one for each involved transactional level, which must be synchronized during execution (reversed order of original execution) and are therefore tightly related to each other.

7. Implementation

This section describes the implementation of the cross-organizational transaction support system based on the architecture described in the previous section, and illustrates this, using screenshots of a real-life business process being executed on the implemented system.

7.1. The implemented system

In the CrossFlow project, a prototype has been built to test and demonstrate the X-transaction model. This section presents the implementation platform on which the system has been build. Then the implementation of the specific transaction support modules is discussed, including a dedicated module that extends the implementation platform functionality required for transaction support. Finally, the graphical user interface is shown that has been implemented to enable issuing rollback request during (cross-organizational) workflow process execution.

7.1.1. The implementation platform. In the CrossFlow prototype system, the underlying workflow management system (WfMS) is IBM's MQSeries Workflow v3.2 [27]. It is a full-fledged, commercially available, workflow management system that offers all (almost all, see Section 7.1.3 below) the functionality required for the cross-organizational extensions developed in the CrossFlow project. MQSeries Workflow uses IBM's DB2 database management system to store process specifications.

The entire prototype system has been implemented in Java using RMI as the communication mechanism [13, 41]. Because of the platform independence of Java, the prototype system can run on any platform for which a java virtual machine has been implemented. The backend systems, i.e. MQSeries Workflow and DB2 are available on a number of platforms, e.g., Unix, Windows NT/2000, and Linux. In testing and demonstrating the prototype system in the CrossFlow project, a combination of linux and Windows NT machines have been used. The former running the backend components and the latter running the workflow clients and the components developed within the project.

The proxy-gateways have all security mechanisms implemented that are related to the crossing of organizational boundaries, because they are the only means for the organizations that participate in the virtual enterprise to communicate with each other (using RMI).

7.1.2. ITM and XTM. The intra-organizational transaction support (ITM) is based on the transaction manager built in the WIDE project [22]. The XTM module is a dynamic event-based software module, like the other CSS modules, that contains the algorithms to compute compensation processes on the contract level. It passes contract level compensation processes to the ITM, which has the functionality to make those processes persistent, so that they can be imported into the workflow management system after which they can be executed.

Although the prototype is built on top of a specific commercial WfMS, the implementation of the isolation layer ensures that it requires only a small effort to exchange the workflow management system used in CrossFlow with another workflow management system.

7.1.3. RRP module. As advanced transaction support is currently not offered by any commercially available workflow management system, including MQ Series Workflow, executing processes will never be (partially) undone. For this reason, additional functionality, e.g. resetting process states and activity states, is required of the workflow management system. In CrossFlow, an additional dedicated module has been implemented by IBM, that resets process states and activity states so that the workflow processes that are compensated are brought into the correct state. This module is called the Reset Restart Process (RRP) module and is only used by the transaction support modules. They provide the RRP module with the identifications of the activities and/or processes that need to have their state reset.

7.1.4. Graphical user interface. Issuing rollback request is done through the graphical user interface (GUI) shown in figure 12. For a selected process that is being executed (shown at the top in the window), it is possible to specify the required rollback mode and rollback scope. In case of a partial rollback, the activity that fails, i.e. the activity from which the rollback will be started, must be specified, so that the correct safepoint(s) can be determined at which the rollback can be stopped and restarted [24]. By pressing the 'Execute' button,

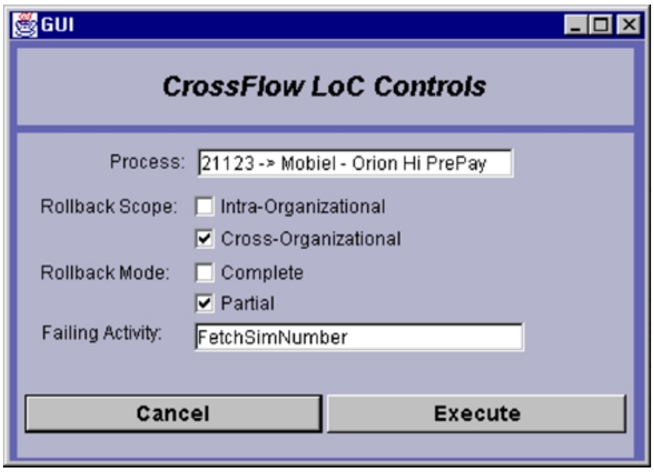

Figure 12. Graphical user interface.

the rollback request is passed to the system after which the transaction modules will stop the process execution, determine which activities need to be undone, dynamically create the compensation process that actually undoes those activities and then start the created compensation process.

7.2. *Example process*

The process of the scenario introduced in Section 4 is a simplification of the real-life business process between a telecom company and a logistics company used in this section. The effects of the developed transaction support on the actual execution of a cross-organizational workflow process running on the implemented CrossFlow system is clearly shown using actual screen shots of the real-life process running on the developed system.

The configuration of the system is as follows:

- one machine running MQWF server (incl. DB2),
- one machine running MQWF client representing the consumer, organization,
- one machine running MQWF client representing the provider. organization.

Although in reality, both organizations would run their own private MQWF server and database system, we have used only one MQWF server and database for reasons of clarity, so that the screen shots show all that is going on in the system. For the same reason, all processes are visible to the user of the system. However, we have demonstrated the developed system using two separate MQWF installations, a video of which can be downloaded from the CrossFlow website [42].

Figure 13 shows the standard user interface of MQSeries Workflow (MQWF) Client that informs the user about the state of workflow processes existing within MQWF. In this example, the processes with category KPN-Consumer corresponds to the consumer process (the telecom company) and the process with category KPN-Provider corresponds to the provider process (the logistics company). The other processes are either subprocesses (category KPN-internal) or different business processes all together. As can be seen in the figure, both the consumer and provider processes are running.

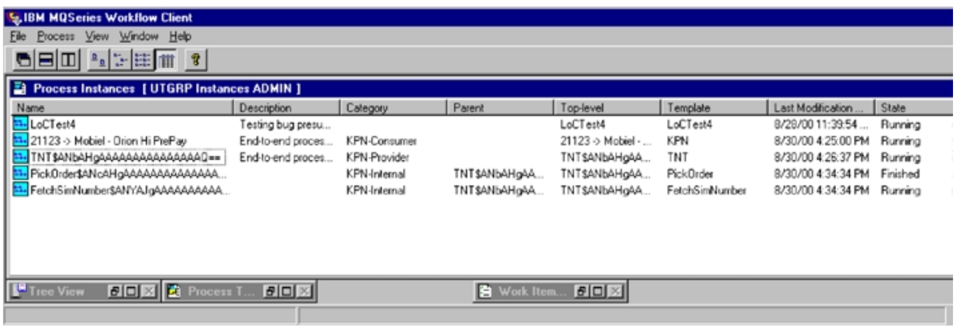

Figure 13. Showing all process states.

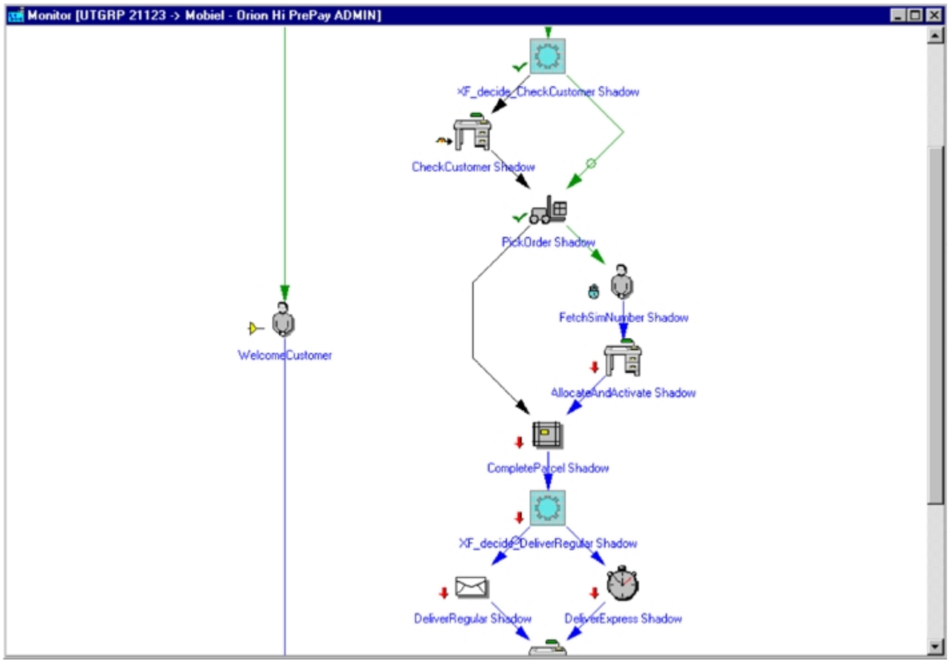

Figure 14. Process monitor before rollback request.

Figure 14 presents the MQWF process monitor applied to the process. It shows the progress of the process. The process on the left-hand side of the figure represents the consumer process, while the right-hand side of the figure represents the provider process. The monitor shown in figure 14 belongs to the consumer organization in which the provider process is shadowed, i.e., the placeholder acts as a white-box process, hence the 'shadow' prefix of the activity names of the provider process, meaning that the activities are not the actual activities being executed.

The symbol next to the activity 'WelcomeCustomer' means that the activity is running, the symbol next to 'FetchSimNumber Shadow' represents a running outsourced activity. All preceding activities are finished ('checked-symbol') and all succeeding activities are not yet started ('down-arrow-symbol'). The symbol next to 'CheckCustomer Shadow' means that the activity was skipped.

In this example, a rollback is issued by the provider organization with a complete rollback mode and cross-organizational rollback scope, using the graphical user interface as shown in figure 12 (but with different parameters). Dealing with the rollback request, the following steps will be performed by the transaction support system:

1. the running processes are suspended.
2. create the compensation process
3. execute the compensation process
4. restart the original processes.

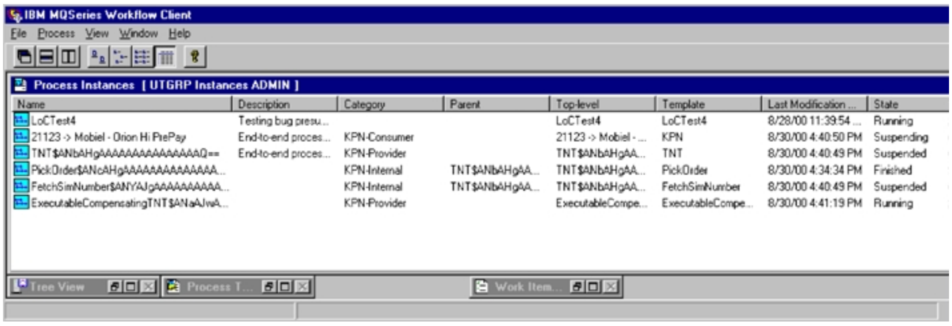

Figure 15. Process states during rollback execution.

Figure 15 shows the process states during the handling of the rollback request. Both the consumer and provider process (and also the provider internal process) are suspended and a new compensation process has been created, started and is running (called 'ExecutableCompensatingTNT$. . . '). The prefix to the process names is created by MQSeries Workflow and used internally in the system to uniquely identify process instances.

The process category of the compensation process shows that this compensating process belongs to the provider organization and it will therefore undo all activities that have been executed within the provider process.

After the compensation within the provider organization has finished, a rollback request is sent to the consumer organization (because of the cross-organizational rollback scope). According to Table 1 in Section 3.3, the consumer process must rollback in partial rollback mode. In this process, the activity that preceded the outsourcing (called 'CheckOrder') is marked as a safepoint and thus the compensation process for the consumer process is empty, i.e. there are no activities that need to be compensated, so the rollback request has been completed.

After the rollback request has been handled, the original process is resumed again, see figure 16. Because the compensation process has finished, it doesn't appear in the

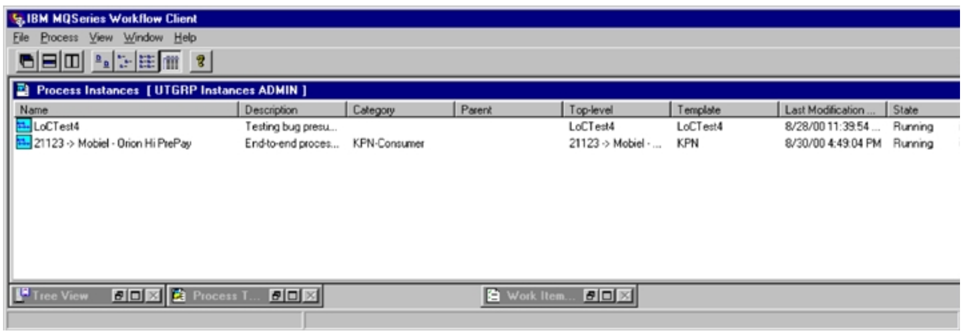

Figure 16. Process states after rollback has been performed.

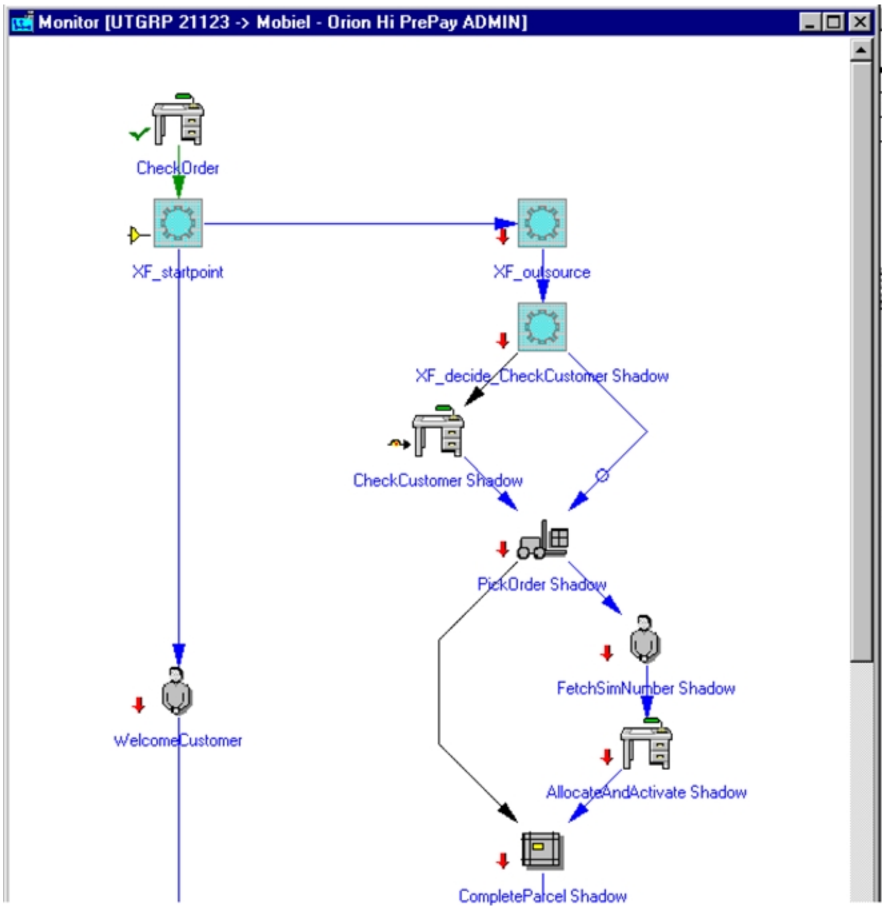

Figure 17. Monitor on process after rollback has been performed.

process state list anymore. The provider process has been completely undone, so that process doesn't appear in the process state list anymore either. The process monitor of the cross-organizational process after the rollback has been completed is shown in figure 17. From this figure, it can be seen that only one activity is running ('XF-startpoint', which is a preliminary activity before outsourcing starts) and that all provider activities (right-hand side of the figure) have not been started yet.

8. Related work

In this section, we discuss work related to the approach described in this paper. As the two main aspects of our work are transactionality and cross-organizational distribution of workflow management, we have organized the related work into these two topics:

- advanced transaction models in workflow management,
- distributed and cross-organizational workflow management.

Both topics are discussed in a separate subsection below. In case related work can be placed into both topics, it is discussed in the topic most appropriate in the context of this paper.

8.1. Advanced transaction models in workflow management

Numerous advanced transaction models have been proposed in the past that offer specific transaction properties required in advanced application areas like workflow management—see e.g. [16, 28, 32] for overviews. Typical and well-known advanced transaction models for process-centric environments are nested transactions [11, 12], flexible transactions [5], multi-level transactions [46], and sagas [18]. The work in [8] describes how ACTA is used as a tool for the synthesis of extended transaction models. Although the framework could be used to show the correctness of a particular implementation, it does not discuss the implementation itself nor does it give an architecture for it. As our approach is based on compensation, we focus on compensation-based approaches below.

Using compensations to roll back long-running processes, like workflow processes, is first described in [18] as the saga transaction model. This model does not include the notion of safepoints and therefore does not explicitly support partial compensation, nor does it include support for cycles in the process. The basic model is later extended to nested sagas [19, 20] to allow for hierarchically organized processes.

The upper level of WIDE advanced transaction model (global transaction model) [22] is taken as a basis for the approach described in this paper. As the WIDE model caters for intra-transactional workflows only, specific cross-organizational transaction aspects have been added. In the WIDE model, compensating activities are used to undo already executed and committed activities of long-running processes. The safepoint concept offers the possibility and flexibility to roll back only parts of a process instead of the entire process [24].

Also based on compensations of workflow processes is the approach developed in the Exotica project [3] as an extension to the Flowmark workflow management system. The Exotica approach relies on statically computed compensation workflow patterns that are used as extensions to the basic workflow specification. The same approach is taken in the FlowBack system [29]. Our approach, however, uses dynamically computed compensation processes. This means that dedicated compensation processes are constructed from process execution history upon rollback request, thereby allowing more complex processes to be compensated. Cross-organizational workflows are not explicitly discussed in this work.

The transaction model described in [33, 34] presents atomicity spheres and isolation spheres. As in the previously mentioned models, both the atomicity and isolation properties of the standard ACID transaction model are relaxed using compensations. Cross-organizational aspects are not included in this work. A sphere-based approach to transactionality in workflows is also presented in [14]. Our approach is not based on identifying specific parts of a workflow as having specific transactional characteristics (the spheres), but rather on treating an entire workflow as a homogeneous, long-running transaction.

8.2. Distributed and cross-organizational workflow management

Distributed execution of workflow processes has received a lot of attention in recent years. The workflow management coalition has created a standard [48] to facilitate the interoperability between different, heterogeneous workflow management systems, indicated as Interface 4. This Interface 4 definition does not include definition of transactional properties or support for transaction management. The specific aspects related to the cooperation between different organization are not mentioned either.

Approached to cross-organizational workflow management as a specialization of distributed workflow management are described in [36], which present key problems related to the cross-organizational workflow management subject. van der Aalst [1] focuses on the modeling and analysis aspects of cross-organizational workflow management. As transaction issues are not covered, the transaction model presented in this paper can be seen as complementary to it.

Transaction support in distributed workflow management is dealt with in [6, 43, 47]. However, these approaches are only applicable to intra-organizational workflow processes, as cross-organizational aspects are not considered. Barbará et al. [6] describe the concept of INCAs (INformation CArriers), which contain all necessary information to execute a workflow process over multiple autonomous systems. The transaction support offered by an INCA depends on the transaction support offered by the autonomous system that executes the INCA. In the Mentor project [47], a transaction processing monitor is used to ensure reliable distributed workflow executions. Transactions are, however, restrictive as they comply with the strict ACID transaction properties. Vonk et al. [43] describes transaction support for distributed workflow management based on compensations, but only covers intra-organizational processes, and serves as the basis for the X-transaction model as described in this paper.

The WISE project [4] covers cross-organizational management and presents an infrastructure for virtual enterprise business processes. Execution guarantees for processes are given based on spheres of atomicity and isolation, the model of which is not elaborated upon. Long-running conversations are proposed in [10]. In this conversational model of interactions, each organization explicitly specifies permissible operations. The state of the conversation is tracked by each organization's system and recovery is performed using the created log. Each organization is therefore responsible for its own internal operations, whether they are transactional or not. Our approach uses the electronic contract to configure the transaction support modules. It is therefore possible to determine beforehand what is allowed or not and the use of logs is not required for that purpose.

RosettaNet [39] is an approach to standardize the electronic exchange of business documents between partners. It consists of black-box service descriptions and offers no control-flow specifications. Our approach is aimed at a fine-grained cooperation between the participating organizations in a virtual enterprise, which requires a more open description of the processes involved and a explicit control flow specification to offer the transaction support as presented in this paper.

The work presented in [2] deals with interorganizational workflow processes, but covers only the modelling aspects. The starting approach is fundamentally different in the sense that we take already existing processes as starting point for which a common process is specified in the electronic contract. In [2] the starting point is to specify a common process and then specify processes in the participating organizations that can satisfy the common process. Also, transaction support is not covered in.

9. Conclusions and future work

This paper describes an advanced transaction model and architecture to support cross-organizational (or inter-organizational) workflow process executions in virtual enterprises. Although the cross-organizational process model consists of an arbitrary number of nesting levels, only three levels have transactional semantics. The advanced transaction model is called the X-transaction model and relies on compensations to undo already executed workflow activities. The combination of rollback scope and rollback mode in the X-transaction model offers a highly flexible rollback mechanism for cross-organizational workflow management.

The cross-organizational transactional architecture facilitates the implementation of the X-transaction model and consists of three layers to provide modularity. The dynamically created cross-organizational infrastructure layer handles the transactional aspects related to the outsourcing of workflow processes, which are described in an electronic contract. The static intra-organizational infrastructure consists of a layer that incorporates the local WfMSs and a layer that includes an isolation layer and a transaction manager that provides intra-organizational transaction support. The architecture is highly flexible in the sense that, besides cross-organizational transaction management, other cooperative support services, e.g. Quality of Service, can be plugged into it, i.e. the architecture consists of a software bus to which cooperative support services can be connected.

The prototype built in the CrossFlow project is tested using two real-world scenarios. One scenario is a real-life logistics scenario which is presented in Section 7.2. The other scenario is a motor damage claim handling process of an insurance company, in which the administrative and financial subprocesses are outsourced to an other organization that is specialized in these kind of processes [7].

At the University of Twente, the work described in this paper is currently used in two spin-off research directions. The first direction is the use of a compensation approach as an element of a flexible support of cross-organizational transactional characteristics in federations of autonomous systems. The second direction is the specification of cross-organizational workflows in electronic contracts, where the specification of execution characteristics includes transactional elements.

Further interesting research topics include the extension of the bilateral model described in this paper to a multi-lateral model. Whereas the work in this paper can support multi-party workflows through the composition of multiple bilateral workflows, a multi-lateral model would allow the direct specification of multi-party workflows. This model, however, will have complex transactional semantics—certainly in the presence of concurrent rollback requests from multiple participating organizations, see Section 5. A formalization of cross-organizational aspects can be elaborated to provide more formal semantics of our approach, e.g. following the lines of Grefen et al. [24].

Acknowledgments

All members of the CrossFlow project are acknowledged for their role in the realization of the X-transaction model, architecture and implementation described in this paper.

Notes

1. Even though inter-organizational workflow management might now be the accepted and most used term to indicate workflow management dealing with processes that span multiple organizations, we call it cross-organizational workflow management for historical reasons: the name of the project is derived from that term and we want to be consistent in relation to other publications related to the CrossFlow project.
2. For reasons of clarity, we will focus on two organization service outsourcing in the following sections. Multi-party service outsourcing requires extensions, which will be explained in Section 5.2.
3. This real-life scenario is based on one of the scenarios [9] used within the CrossFlow project, but is simplified in this paper for reasons of clarity and brevity.

References

1. W.M.P. van der Aalst, "Interorganizational workflows—An approach based on message sequence charts and petri nets," Systems Analysis–Modelling–Simulation, vol. 34, no. 3, 1999.
2. W.M.P. van der Aalst and M. Weske, "The P2P approach to interorganizational workflow," in Proc. of the Conference on Advanced Information Systems Engineering (CAiSE), Switzerland, 2001.
3. G. Alonso, D. Agrawal et al., "Advanced transaction models in workflow contexts," in Procs. Int. Conference on Data Engineering (ICDE), USA, 1996.
4. G. Alonso, U. Fiedler, C. Hagen, A. Lazcano, H. Schuldt, and N. Weiler, "WISE: business to business e-commerce," in Procs. Int. Workshop on Research Issues in Data Engineering, Australia, 1999.
5. M. Ansari, L. Ness, M. Rusinkiewicz, and A. Sheth, "Using flexible transactions to support multi-system telecommunication applications," in Procs. Very Large Data Bases (VLDB), Vancouver, Canada, 1992.
6. D. Barbará, S. Mehrotra, and M. Rusinkiewicz, "INCAs: Managing dynamic workflows in distributed environments," Journal of Database Management—Special Issue on Multidatabases, vol. 7, no. 1, 1996.
7. S. Browne, "Insurance prototype deployment report," Public CrossFlow Deliverable D12, 2000 (also available via: http://www.crossflow.org).
8. P.K. Chrysanthis and K. Ramamritham, "Synthesis of extended transaction models using ACTA," ACM Transactions on Database Systems, vol. 19, no. 3, 1994.
9. Z. Damen, W. Nijenhuis, and M. Verwijmeren, "Transport scenario description," Public CrossFlow Deliverable D2a, 2000 (also available via: http://www.crossflow.org).
10. A. Dan et al., "Business to business integration with TpaML and a B2B protocol framework (BPF)," IBM Research Report RC21863, IBM research Division Thomas J. Watson Research Center, USA, 2000.
11. U. Dayal, M. Hsu, and R. Ladin, "Organizing long-running activities with triggers and transactions," in Procs. 1990 ACM SIGMOD Int. Conf. on Management of Data, Atlantic City, USA, 1990.
12. U. Dayal, M. Hsu, and R. Ladin, "A transactional model for long-running activities," in Procs. 17th Int. Conf. on Very Large Databases, 1991.
13. P. Deitel and H. Deitel, Java: How to Program v1.2, Prentice Hall, 1999.
14. W. Derks, J. Dehnert, P. Grefen, and W. Jonker, "Customized atomicity specification for transactional workflows," in Proc. 3rd International Symposium on Cooperative Database Systems for Advanced Applications, Beijing, China, 2001.
15. M. Duitshof, "Logistics prototype deployment report," Public CrossFlow Deliverable D13, 2000 (also available via: http://www.crossflow.org).
16. A.K. Elmagarmid (Ed.), "Database Transaction Models for Advanced Applications, Morgan Kaufmann: USA, 1992.

17. A. Eyal and T. Milo, "Integrating and customizing heterogeneous e-commerce applications," The VLDB Journal, vol. 10, no. 1, 2001.
18. H. Garcia-Molina and K. Salem, "Sagas," in Procs. 1987 ACM SIGMOD Int. Conf. on Management of Data, USA, 1987.
19. H. Garcia-Molina et al., "Modeling long-running activities as nested sagas," in IEEE Data Engineering Bulletin, vol. 14, no. 1, 1991.
20. H. Garcia-Molina et al., "Coordinating multitransaction activities with nested sagas," in Recovery Mechanisms in Database Systems, 1998.
21. P. Grefen, J. Vonk, E. Boertjes, and P. Apers, "Semantics and architecture of global transaction support in workflow environments," in Procs. Int. Conf. on Cooperative Information Systems (CooPIS), United Kingdom, 1999.
22. P. Grefen, B. Pernici, and G. Sánchez (Eds.), Database Support for Workflow Management—The WIDE Project, Kluwer Academic Publishers, 1999.
23. P. Grefen, K. Aberer, Y. Hoffner, and H. Ludwig, "CrossFlow: Cross-organizational workflow management in dynamic virtual enterprises," Int. Journal of Computer Systems Science & Engineering, vol. 15, no. 5, 2000.
24. P. Grefen, J. Vonk, and P. Apers, "Global transaction support for workflow management systems: From formal specification to practical implementation," The VLDB Journal, vol. 10, no. 4, Springer, 2001.
25. Y. Hoffner, "Supporting contract match-making," in Procs. IEEE 9th Int. Workshop on Research Issues in Data Engineering, Australia, 1999.
26. Y. Hoffner, S. Field, P. Grefen, and H. Ludwig, "Contract-driven creation and operation of virtual enterprises," Computer Networks, vol. 37 (2001), pp. 111–136, Elsevier, 2001.
27. IBM Corporation, MQ Series Workflow web site, http://www.software.ibm.com/ts/mqseries/workflow, 2002.
28. S. Jajodia and L. Kerschberg (Eds.), Advanced Transaction Models and Architectures, Kluwer Academic Publishers, 1997.
29. B. Kiepuszewski, R. Muhlberger, and M. Orlowska, "FlowBack: Providing backward recovery for workflow management systems," in Procs. 1998 ACM SIGMOD, Int. Conf. on Management of Data, USA, 1998.
30. J. Klingemann, "Controlled flexibility in workflow management," in Procs. 12th Conf. on Advanced Information Systems Engineering, Sweden, 2000.
31. M. Koetsier, P. Grefen, and J. Vonk, "Contracts for cross-organizational workflow management," in Proc. 1st Int. Conf. on Electronic Commerce and Web Technologies (EC-Web), United Kingdom, 2000.
32. V. Kumar and M. Hsu, Recovery Mechanisms in Database Systems, Prentice Hall, 1998.
33. F. Leymann, "Supporting business transactions via partial backward recovery in workflow management systems," in Procs. BTW '95.
34. F. Leymann and D. Roller, Production Workflow—Concepts and Techniques, Prentice Hall, 2000.
35. F. Leymann, "Web services flow language (WSFL 1.0)," IBM Corporation, 2001. http://www-4.ibm.com/software/solutions/webservices/
36. H. Ludwig, C. Bussler, M. Shan, and P. Grefen, "Cross-organizational workflow management and coordination—WACC '99 workshop report," ACM SIGGROUP Bulletin, vol. 20, no. 1, 1999.
37. H. Ludwig and P. Grefen, "Report on ISDO'00: The CAiSE'00 workshop on infrastructures for dynamic business-to-business service outsourcing," ACM SIGMOD Record, vol. 29, no. 3, 2000.
38. M. Mecella and B. Pernici, "Designing wrapper components for e-services in integrating heterogeneous systems," The VLDB Journal, vol. 10, no. 1, 2001.
39. RosettaNet Implementation Framework: Core Specification (RNIF 02), RosettaNet, 2001. http://www.rosettanet.org/
40. C. Stricker, S. Riboni, M. Kradolfer, and J. Taylor, "Market-based workflow management for supply chains of services," in Procs. 33rd Hawaii Int. Conf. on System Sciences, Hawaii, 2000.
41. Sun Corporation, The Java web site, http://java.sun.com, 2002.
42. The CrossFlow website, http://www.crossflow.org, 2002.
43. J. Vonk, P. Grefen, E. Boertjes, and P. Apers, "Distributed transaction support for workflow management applications," in Procs. 10th Int. Conf. on Database and Expert System Applications (DEXA), Florence, Italy, 1999.
44. J. Vonk, W. Derks, P. Grefen, and M. Koetsier, "Cross-organizational transaction support for virtual enterprises," in Procs. 7th Int. Conf. on Cooperative Information Systems (CoopIS), Eilat, Israel, 2000.

45. J. Vonk, W. Derks, P. Grefen, and M. Koetsier, “Model, architecture and system for cross-organizational transaction support in virtual enterprises,” CTIT Technical Report 00-20, University of Twente, The Netherlands, 2000.
46. G. Weikum, “Principles and realization strategies of multilevel transaction management,” ACM Transactions on Database Systems, vol. 16, no. 1, 1991.
47. D. Wodtke, J. Weissenfels, G. Weikum, and A. Dittrich, “The MENTOR project: Steps towards enterprise-wide workflow management,” in Procs. Int. Conf. on Data Engineering, USA, 1996.
48. Workflow Management Coalition, Interface 4: Interoperability Abstract Specification v1.0, 1996.
49. Workflow Management Coalition, Terminology & Glossary v3.0, 1999.