Model, Architecture and System for Cross-Organizational Transaction Support in Virtual Enterprises♣

Jochem Vonk‡, Wijnand Derks*, Paul Grefen‡, Marjanca Koetsier‡

‡Center for Telematics and Information Technology (CTIT),
University of Twente, {vonk, grefen, marjanca}@cs.utwente.nl
*KPN Research, w.l.a.derks@kpn.com

♣ The work presented in this report is supported by the European Commission in the CrossFlow Project (ESPRIT No. 28635). Partners in CrossFlow 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 shorter version of this report has appeared in [Vonk00].

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. To apply workflow management technology in these virtual enterprises, support for cross-organizational processes is necessary. 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 report presents a high-level transaction model and architecture for cross-organizational workflow processes. Characteristic of the model is the flexibility in rollback semantics by combining rollback modes and rollback scopes, supported by a dynamically generated architecture that is configured conforming to an electronic contract that has been established between the different organizations. The transaction model and architecture are independent of the underlying workflow management system platform, however, in the CrossFlow project the presented technology has been implemented on top of IBM's MQ Series Workflow.

keywords: Transaction Management, Virtual Enterprises, Workflow Management, B2B E-Commerce

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 the workflow management system ensures that these business processes are executed in a reliable and consistent manner. This can be ensured by incorporating transaction semantics in the processes [Elma92, Leym95, Jajo97, Gref99b].

Besides the focus-shift of organizations to apply workflow management for primary business processes, organizations are focussing more and more on their core business while leaving non-core businesses to other specialized organizations. This introduces the concept of dynamic virtual enterprises. Multiple organizations with their own primary processes combine forces in a virtual enterprise for a period of time. Afterwards, the virtual enterprise is dismantled again, hence the dynamic characteristic of virtual enterprises [Ludw99, Ludw00]. The CrossFlow project [Gref00] aims to develop support for cross-organizational workflow management in these dynamically formed virtual enterprises, resulting in fine-grained contract-based cooperation. Although a virtual enterprise can consist of more than two organizations, the scope within the CrossFlow project is limited to the commonly used consumer/provider paradigm. In the consumer/provider paradigm, one organization acts as a (service) consumer that outsources part of its business process to another organization, called the (service) provider. The thereby formed cooperation between the organizations is specified in an electronic contract. The transaction model and architecture presented here, is developed within the CrossFlow project of which a more detailed description is given in Section 6.

This report discusses a three-level transaction model for cross-organizational workflow management that ensures reliable execution of the workflow processes within virtual enterprises, for which a standard transaction model is not sufficient. The developed transaction model offers different rollback modes and rollback scopes that allow for flexible rollbacks of executed processes. Rolling back processes is based on executing compensating activities that semantically undo the effects of already

executed activities. The presented architecture that supports the transaction model consists of a static intra-organizational infrastructure layer and a dynamically generated cross-organizational infrastructure layer.

The structure of this report is as follows. Section 2 discusses related work. In Section 3, the intra- and cross-organizational business process models are explained and an example scenario is introduced. The architecture to support cross-organizational workflow process executions is presented in Section 4. The X-transaction model that ensures the reliable, consistent execution of (cross-organizational) workflow processes is presented in Section 5. The architecture and implementation issues to support the X-transaction model are discussed in Section 6, together with the CrossFlow project, which is the context of the work described in this report. The report ends with conclusions and a discussion of future work.

2. Related Work

Numerous advanced transaction models have been proposed in the past, see e.g. [Elma92, Jajo97, Kuma98], that offer specific transaction properties required in advanced application areas, like workflow management. The cross-organizational transaction model presented in this report is not created from scratch, but combines aspects of existing advanced transaction models, extended to deal with cross-organizational issues. The WIDE advanced transaction model [Gref99b] is taken as a basis, and specific cross-organizational transaction aspects have been added. In the WIDE model, compensating activities are used to undo already executed and committed activities. The safe-point concept offers the possibility to rollback only parts of a process instead of the entire process. Using compensations to rollback long-running processes, like workflow processes, is first described in [Garc87] and is called the saga transaction model. Also based on compensations is the transaction model developed in the Exotica project [Alon96], however, it relies on statically computed compensation patterns, while our model dynamically computes a compensation process only when a rollback is necessary. The transaction model described in [Leym95, Leym00] 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. All transaction models mentioned above do not deal with cross-organizational aspect but are limited to intra-organizational workflow support. In [Geor96] a model is presented to represent customized transaction semantics to determine whether the model can be executed by the underlying systems. In addition, an architecture is presented to complement missing transaction management facilities.

Distributed execution of workflow process has also received a lot of attention in recent years. The workflow management coalition has created a standard [WfMC96] to facilitate the interoperability between different, heterogeneous workflow management systems, albeit without transactional properties. Aspects related to the specific cooperation between different organization are not mentioned either. Cross-organizational workflow management, a special kind of distributed workflow management is discussed in [Ludw99], which presents key problems related to the

cross-organizational workflow management subject. [Aals99] focuses on the modeling and analysis side related to cross-organizational workflow management. As transaction issues are not covered, the transaction model presented in this report can be seen as complementary to it.

Transaction support in distributed workflow management is dealt with in [Barb96, Wodt96, Vonk99]. However, only intra-organizational workflow processes are considered. [Barb96] describes 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 [Wodt96], a transaction processing monitor is used to ensure reliable distributed workflow executions. Transactions are, however, restrictive as they comply to the strict ACID transaction properties. [Vonk99] describes transaction support for distributed workflow management based on compensations, but only covers intra-organizational processes.

The WISE project [Alon99] 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 [Dan00]. 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.

3. Process Model and Example Scenario

The business processes of organizations are specified in workflow process models so that they can be executed by a workflow management system (WfMS). This section first describes how intra-organizational business processes are modeled, illustrated with an example. Then, a cross-organizational process model in which the cross-organizational processes are specified is presented. Finally, an example scenario is introduced which shows a cross-organizational process that is used throughout the rest of this report to illustrate the cross-organizational transactional aspects.

3.1 Intra-Organizational Process Model

To apply workflow management, the business processes of an organization must be modeled in a workflow process model. A workflow process model consists of the activities that must be performed and the order in which they must be performed, called the control flow. Within the control flow it is possible to specify parallelism of activities, choices between activities and loops over activities [WfMC99] using different control connectors. Most business processes are complex in nature, consisting of numerous activities and a complex control flow. Therefore, 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 activities. This 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 consist of other basic activities or subprocesses.

3.2 Example Intra-Organizational Process Scenario

Figure 1 shows an example of a business process that is modeled as a nested workflow process. It illustrates the business process of a logistics organization that delivers GSM phones from a warehouse to a customer. In the remainder of this report, this process will be further elaborated. The rounded rectangles represent activities and the arrows represent the control flow. The diamond is a control connector representing an OR-split, meaning that one of the following activities can be executed, and 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 four activities. For reasons of brevity, the control flow in this example is relatively simple. However, the full set of control connectors as specified in [WfMC99] is offered in the intra- and cross-organizational process models described in this section.

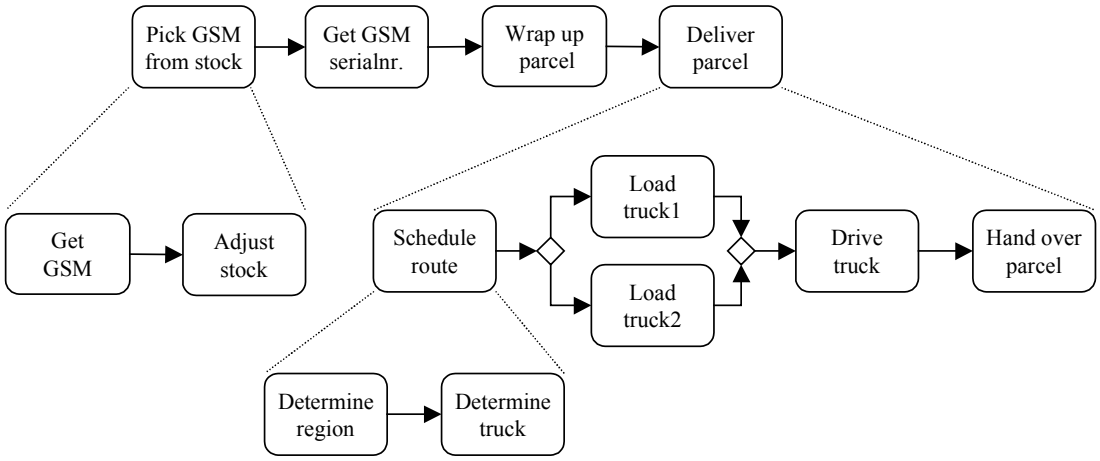

Figure 1: Logistics process

When organizations form virtual enterprises and cross-organizational workflow processes have to be executed, the intra-organizational process model does not suffice and a cross-organizational process model is required in which the specific aspects of cross-organizational workflow processes are captured.

3.3 Cross-Organizational Process Model

In a cross-organizational setting based on the consumer/provider paradigm, two organizations are involved in the execution of the cross-organizational workflow process. The provider organization executes a workflow process on behalf of a consumer organization. However, the provider organization does not want to disclose all details of the workflow process it executes. In addition, most consumer organizations do not even want to know the details of that workflow process, but would rather have an abstract view of it. To encapsulate the details of the process, the provider only discloses aspects of the process it is willing to make publicly known and that are of interest to possible consumer organizations. This encapsulation of the

process creates a common view and is called the contract level. As the contract level of the process is specified in the contract that has been agreed upon, it is visible to both the consumer and provider organizations, while the internal process structure of the provider process is encapsulated.

For example, in the logistics process shown in Figure 1, the logistics company implementing the process can act as a provider organization, but is willing to show only the top four activities or subprocesses, which constitute the contract level. The other activities or subprocesses are encapsulated by the contract level and are thus not visible to a consumer organization. That part of the process is called the internal level.

Note that the activities or subprocesses that are part of the contract level are specified by the provider organization. These activities or subprocesses can exist on any refinement level of the internal provider process, thereby forming a partition of the entire provider process.

To incorporate the outsourced process in the consumer process, the placeholder concept is introduced. The placeholder is a special kind of activity within the consumer representing a subprocess that is executed under the control of another organization and on another WfMS, i.e. the outsourced process.

3.4 Example Cross-Organizational Process Scenario

The cross-organizational example scenario describes a virtual enterprise that has been established by, and consists of, a telecom company acting as the consumer organization and a logistics company acting as the 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. This example scenario is based on one of the real-world scenarios used within the CrossFlow project [Dame00], but has been simplified in this report for reasons of clarity and brevity.

In the example, shown in Figure 2, 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, in parallel to the continued execution of the consumer process, the outsourced process is started (shown as a double lined rectangle, i.e. the placeholder). In the provider organization, the GSM phone is taken from stock and the serial number of the GSM phone is sent to the consumer (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. At the same time 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, the telecom company checks up on the client for marketing and customer satisfaction purposes.

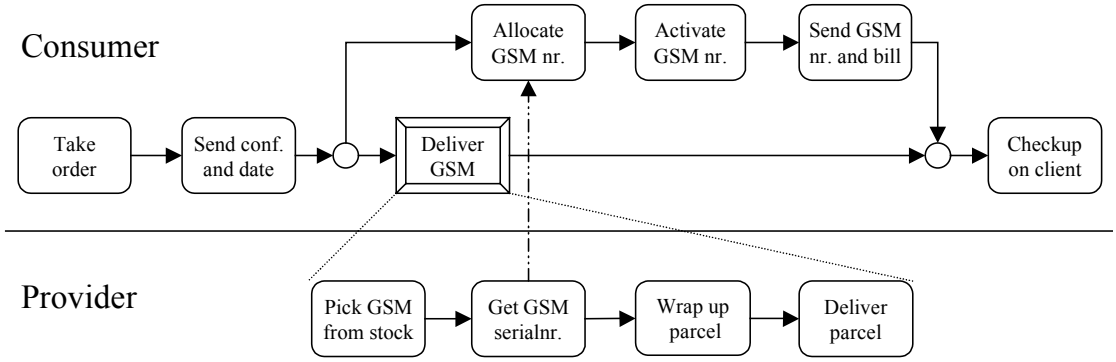

Figure 2: Cross-organizational process with outsourced logistics process

4. Basic Architecture for Cross-Organizational Workflow Management

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 in its interface 4 standard [WfMC96]. In a cross-organizational setting, autonomous organizations are involved that each have their own business rules and culture. Besides this, as explained previously, both organizations have a common view on the contract level and do not want to show the details of their core business processes. The contract contains all specific information necessary to perform the cross-organizational process including the rights and obligations of both organizations, which must be enforced by the cross-organizational workflow management architecture. This requires a mechanism that maps the contract to an infrastructure that controls all cross-organizational aspects of the process. From this, it follows that the architecture is multi-layered. One layer deals with the internal processes of the consumer and provider organizations and consists of the individual WfMSs of the involved organizations. This architecture layer is static because it is also used to execute the intra-organizational processes of the organizations. A second layer deals with the processes on the contract level, which is only required for the time that the contract is valid and hence is a dynamic layer. A third layer is an isolation layer that shields the other two layers from each other.

The basic architecture is shown in Figure 3. The upper half of the architecture is the dynamic cross-organizational infrastructure, which is configured using the agreements specified in the 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 will be different, the cross-organizational infrastructure will be configured differently for both organizations. The cross-organizational infrastructure consists of the necessary modules to provide dedicated support for different aspects of the cross-organizational workflow process execution, called cooperative support services (CSSs) and a proxy-gateway that provides a communication mechanism to handle all communication between the involved organizations.

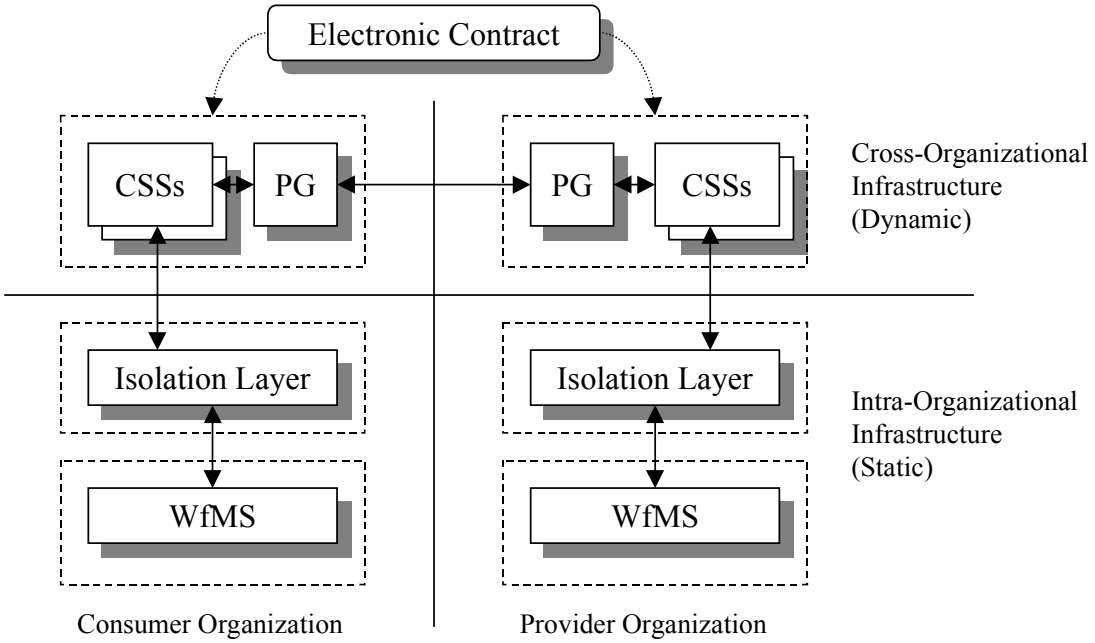

Figure 3: Basic Cross-Organizational WfM Architecture

The bottom half in the figure is the static intra-organizational infrastructure. It consists of the local workflow management system (WfMS), which executes the separate processes of the organization, and the isolation layer that shields the cross-organizational infrastructure from WfMS specifics. The isolation layer maps the WfMS independent contract level processes to the WfMS dependent internal level processes. This way the cross-organizational infrastructure is independent of the underlying WfMS, which can be different for each organization.

5. The Cross-Organizational 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.

5.1 Intra-Organizational Transaction Model

Various advanced transaction models [Elma92] have been proposed to overcome the problems related to the long-livety of intra-organizational workflow processes by relaxing the atomicity and isolation constraints. Instead of inventing yet a new transaction model from scratch, we have taken the existing WIDE transaction model [Gref99b] as a basis and extended it to handle the specific requirements imposed by cross-organizational workflow processes. The WIDE transaction model is in turn based on the saga transaction model as proposed in [Garc87], extended with the safe-point concept that allows flexible process rollbacks.

Long-running workflow processes are divided into smaller, relatively short running, process steps that commit the results after the step completes. As these steps are related to the intra-organizational workflow process, they are called I-steps. Each I-step has a compensating counterpart specified for it that semantically undoes the effect of the original I-step. In case of failures, the process is rolled back by executing the compensating steps in the reverse order in which their original counterpart I-steps have been executed. Marking I-steps as a safe-point indicates that a rollback could be stopped at those steps, because a consistent state in the process has been reached. This means that every committed I-step executed after the safe-point is compensated, but not the safe-point itself. Whether the rollback actually stops at those safe-points is determined by the compensation algorithm and depends mainly on the occurrence of parallel executed activities. A formal description of the compensation algorithm is presented in [Gref99a].

If an activity in the process execution fails and a rollback is required, it is up to the performer of the activity or the process manager to issue the rollback in which he can indicate the rollback mode. The rollback mode can be either partial or complete. In the first case, the rollback will compensate or undo the process until a suitable (set of) safe-points 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.

5.2 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 global 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 is even more severe if the consumer/provider paradigm is extended to include more than two organizations. Thus, to preserve the autonomy of the involved organizations, cross-organizational workflow processes require transaction support that offers loose transaction properties.

Similar to dividing the intra-organizational processes, 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. Because it must be possible 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 and is executed by the provider organization. Therefore, it is the provider organization that has to specify these contract level compensating activities. To inform the consumer about how the outsourced process

will be compensated by the provider, the contract level compensating 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 3, 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 placeholder, i.e. the activity in the consumer process that represents the outsourced process, is a regular I-step and must therefore have a compensating activity specified for it. The outsourcing level is only visible to the consumer organization.
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.

5.3 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. It 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 safe-points, it is also possible to specify X-steps as safe-points. This way, the flexibility in rollback handling offered by partial rollbacks is offered at the contract level as well. However, an consistency constraint exists between the contract level safe-points and the internal level safe-points. As the X-steps, i.e. the activities at the contract level, are an encapsulation of the detailed internal process of the provider, contract level safe-points depend on the

safe-points that are specified on the internal level. An X-step can only be marked as a safe-point when the last I-step of that X-step is marked as a safe-point.

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

1. At the 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 counterpart activity of the placeholder is executed, which is specified by the consumer and does not necessarily involve the provider organization that has executed the outsourced process. Just like any other I-step, the placeholder can also be specified as a safe-point
2. At the contract level. A rollback on the contract level will involve only the X-steps 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.
3. At the internal level. A rollback will involve I-steps that are internal to the provider process and are thus not visible to the consumer.

The operational semantics of a rollback in a cross-organizational workflow process is described in the next subsection.

5.4 Cross-Organizational Operational Transaction Semantics

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

In the previous subsection is described that rollbacks can occur on any of the three transactional levels of the cross-organizational process model. However, when a

rollback takes place at the internal level, the consumer 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. To offer the consumer more insight in the provider compensation process, the internal level compensation is 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 contract level activity is specified in the contract the consumer can trace the compensation process while the provider is executing it.

Rollback starts at: / Rollback Scope and mode	**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 safe-point 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 safe-point 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 rolled back it is rolled back by the provider.	Not possible: Either the process is rolled back partially to a safe-point in the provider process (which is thus intra-organizational) or the rollback is cross-organizational and thus the provider process is rolled back completely.

As an example, suppose the activities ‘Get GSM’ and ‘Adjust stock’ must be compensated. These activities are not visible to the consumer, but the activity ‘Pick GSM from stock’ is visible to the consumer. Instead of compensating both internal level activities, the same effect is reached when the contract level activity is compensated.

5.5 Rollback Examples

To illustrate the effects of different rollback modes and rollback scope in a rollback request, the example scenario introduced in Section 3 is used. Figure 4 shows the

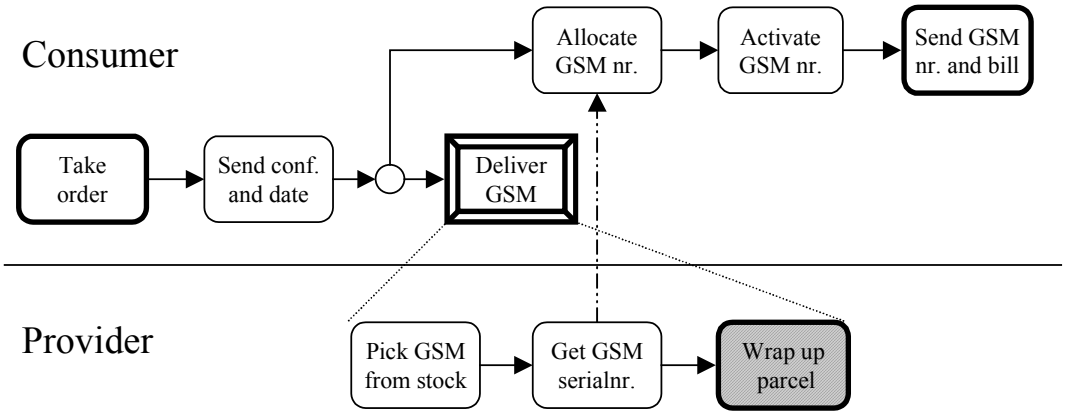

Figure 4: Rollback KPN/TNT example

X-transaction of the process, 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 safe-points that are specified in the process.

For a first example, suppose that the running activity at the provider fails, as indicated by the shaded rectangle in the figure, because during packaging it is discovered 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. For this, the entire process needs to be rolled back, thus the rollback mode is complete. Both 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 cross-organizational, which implies that the consumer process will also be rolled back (in partial mode, see table). The resulting compensation process, consisting of the compensating counterpart activities, is shown in Figure 5.

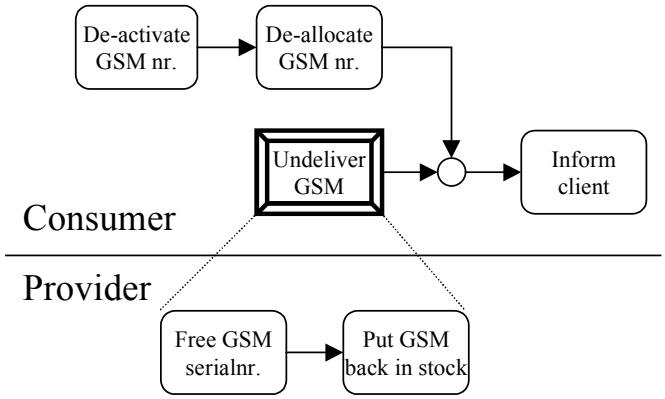

Figure 5: Example compensation process

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') and the original process is restarted again with activity 'Send conf. and date'.

As a second example, suppose that the last activity in the cross-organizational workflow process 'Checkup on client' 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) or to compensate the process by itself. 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, i.e. de-activating and de-allocating the GSM number, also its own compensating activity of the delivery of the GSM, 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.

6. Transaction Architecture for Cross-Organizational WfM

In this section, the basic cross-organizational workflow management architecture presented in Section 4 is extended to offer support for the X-transaction model as it is discussed in the previous section.

6.1 Cross-Organizational Transactional Architecture

The transactional architecture is shown in Figure 6. The cross-organizational infrastructure of Figure 3 has been expanded and the cooperative support service modules have been replaced with the specific cooperative support modules that are required to facilitate cross-organizational transaction support. In addition to this, an extra module (ITM) is introduced in the intra-organizational infrastructure that provides the intra-organizational transaction support.

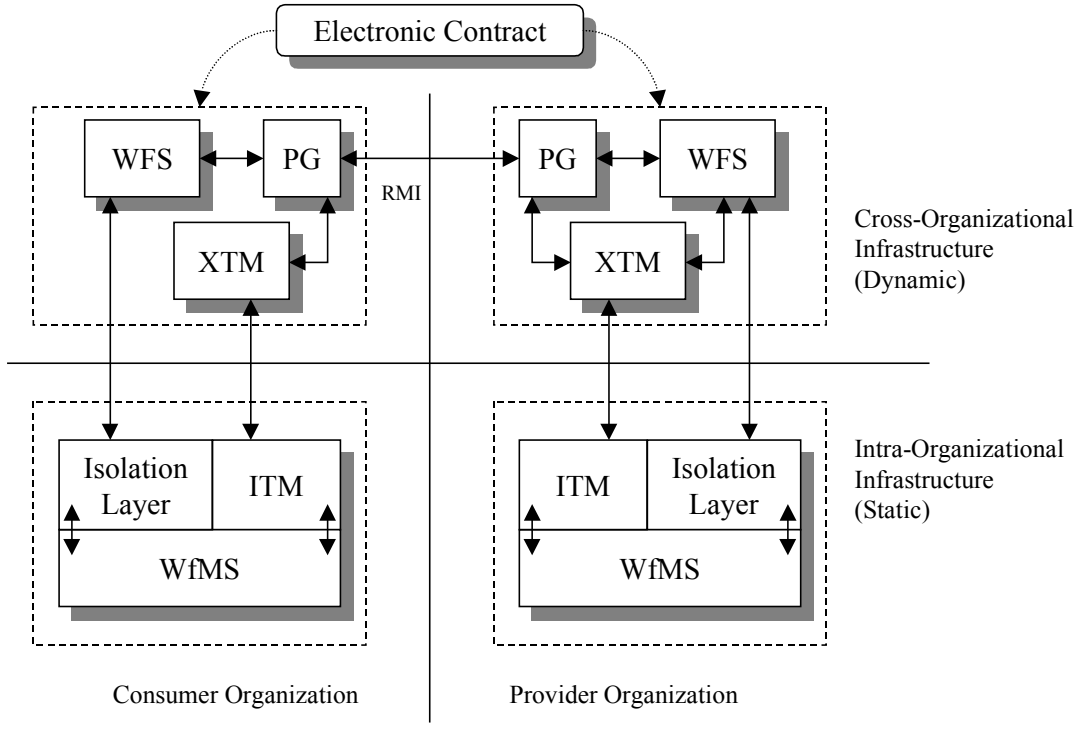

Figure 6: Transaction architecture

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, which means that the consumer side XTM and WFS usually differ from the provider side XTM and WFS, see also Section 4. It also means that the XTM and WFS are only required when the outsourced process starts and are no longer required when the contract expires.

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 as the internal level activities and process.

The XTM provides the cross-organizational transaction support. It has the necessary algorithms to calculate compensating workflow processes to undo executed X-steps. To determine which X-steps have been executed that together form the workflow execution history on the contract level, it uses the information provided by the WFS. Although the cross-organizational architecture includes a XTM module for both organizations, only the provider side XTM requires the algorithms to calculate compensating processes, because the contract level activities are executed by the provider and must be compensated at the provider. The consumer side XTM determines whether the contract level process must be compensated whenever a rollback is started within the consumer process itself.

The proxy-gateways (PG) interact with each other using Java RMI and handle the communication between the different organizations, thereby providing security mechanisms to protect the organizations.

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 transactional functionality, as required by workflow management, see Section 5, 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.

Note that, when a rollback occurs that involves a combination of the three transactional levels as described in Section 5, 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 are, however, tightly related to each other.

6.2 The CrossFlow Context

As stated before, the work presented in this report is part of the CrossFlow project [Gref00]. The CrossFlow project aims to develop the support 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 is covered, but also the dynamic creation of a virtual enterprise based on the consumer/provider paradigm. Organizations find each other in an electronic marketplace where organizations offer their services or search for services that are offered in the marketplace (business-2-business e-commerce). Using a matchmaking facility [Hoff99], compatible organizations form a virtual enterprise, the cooperation in which is described and established in an electronic contract [Koet99]. The electronic contract not only contains the specification of the service, but also the rights and obligations that the service entails. To facilitate a smooth cooperation between the organizations in a virtual enterprise, the business processes that are to be performed in the involved organizations must be interconnected. Merely connecting the workflow management systems of the organizations does not suffice. The architecture presented in Section 4 is developed to handle the specific issues that arise as a consequence of the cooperation between the different organizations. Specific cooperation support services are developed that each deal with a different aspect of the cross-organizational workflow management requirements. The cooperative support services covered in CrossFlow are transaction management as presented in this report, Level of Control, Quality of Service and Flexible Change Control [Klin00].

6.3 Implementation Issues

In the CrossFlow project, a prototype has been build with which it is possible to test and demonstrate the described X-transaction model. The underlying workflow management system (WfMS) in the prototype is IBM's MQ Series Workflow [MQWF00]. The intra-organizational transaction support, i.e. the ITM as shown in Figure 6, is based on the transaction manager built in the WIDE project. An additional

module has been implemented that resets process states and activity states so that the workflow processes that are compensated are brought into the correct state. This module is required because its functionality, i.e. resetting states of (partly) executed workflow processes, is currently not offered by any commercially available workflow management system, and thus also not offered by MQ Series Workflow. The XTM module is a dynamic event-based software module 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. The entire prototype is built in Java using RMI as the communication mechanism [Java99]. 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.

A graphical User Interface has been realized that is used to issue rollback request, an example of which is shown in Figure 7. For a selected process that is being executed (shown at the top in the GUI), it is possible to specify the required rollback mode and rollback scope and, in case of a partial rollback, the activity that fails and from which the compensation should begin. The appendix contains a more elaborated rollback example using the CrossFlow system within the real-life logistics scenario.

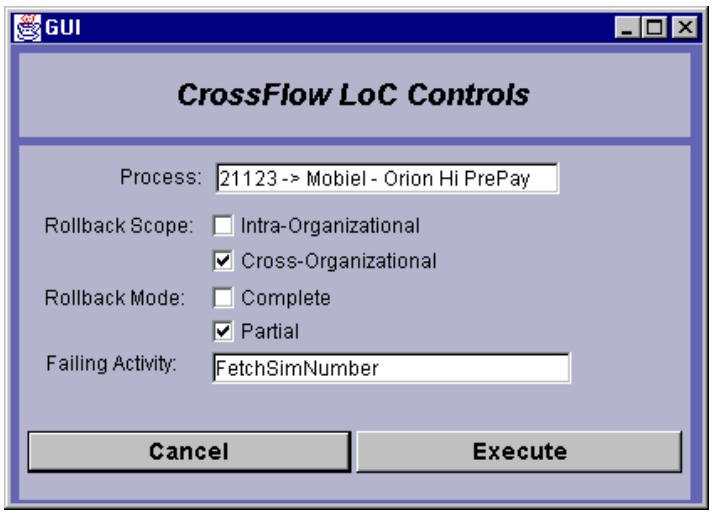

Figure 7: Graphical User Interface

7. Conclusions and Future Work

This report describes an advanced transaction model and architecture to support cross-organizational workflow process executions within 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 (three transactional levels) and rollback mode (safe-points) 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. 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 will be tested using two real-world scenarios. One scenario is an extended version of the scenario presented in Chapter 3. The other scenario is a motor damage claim handling process of an insurance company, in which the daily administrative and financial subprocesses are outsourced to an other organization that is specialized in these kind of processes.

Further developments with respect to cross-organizational transaction management can include an extension of the consumer/provider paradigm so that more than two organizations can be involved in the outsourcing of processes. This offers the possibility of nested outsourcing, in which provider organizations can again outsource part of their processes, thereby acting as a consumer organization as well. It also offers the opportunity for a consumer organization to outsource multiple parts of its

process to multiple provider organizations, which implies that multiple contracts have to be established. Both situations require an extension of the X-transaction model to offer a refinement of the rollback scope so that it can be indicated which organizations are involved in the rollback. Because multiple organizations are involved that can all issue a rollback request simultaneously, the X-transaction model must be further extended to deal with concurrent rollback requests, similar to the work described in [Vonk99].

Acknowledgements

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 report.

References

[Aals99] W. M. P. van der Aalst; *Interorganizational Workflows – An approach based on Message Sequence Charts and Petri Nets*; Systems Analysis – Modelling – Simulation, 34(3), 1999.

[Alon96] G. Alonso, D. Agrawal et al.; *Advanced Transaction Models in Workflow Contexts*; Procs. Int. Conference on Data Engineering; USA, 1996.

[Alon99] G. Alonso, U. Fiedler, C. Hagen, A. Lazcano, H. Schuldt, N. Weiler*; WISE: Business to Business E-Commerce*; Procs. Int. Workshop on Research Issues in Data Engineering; Australia, 1999.

[Barb96] D. Barbará, S. Mehrotra, M. Rusinkiewicz; *INCAs: Managing Dynamic Workflows in Distributed Environments*; Journal of Database Management – Special Issue on Multidatabases 7(1), 1996.

[Dame00] Z. Damen, W. Nijenhuis, M. Verwijmeren; *Transport Scenario Description*; Public CrossFlow Deliverable, 2000. (available via: `http://www.crossflow.org`)

[Dan00] 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.

[Elma92] A. K. Elmagarmid (Ed.); *Database Transaction Models for Advanced Applications*, Morgan Kaufmann; USA, 1992.

[Garc87] H. Garcia-Molina, K. Salem; *Sagas*; Procs. 1987 ACM SIGMOD Int. Conf. on Management of Data,; USA, 1987.

[Garc91] H. Garcia-Molina; *Nested Sagas*; in [Kuma98]

[Geor96] D. Georgakopoulos, M. Hornick, F. Manola; *Customizing Transaction Models and Mechanisms in a Programmable Environment Supporting Reliable Workflow Automation*; IEEE Transactions on Knowledge and Data Engineering 8 (4), 1996.

[Gref99a] P. Grefen, J. Vonk, E. Boertjes, P. Apers; *Semantics and Architecture of Global Transaction Support in Workflow Environments*; Procs. Int. Conf on Cooperative Information Systems (CooPIS), United Kingdom; 1999.

[Gref99b] P. Grefen, B. Pernici, G. Sánchez (Eds.); *Database Support for Workflow Management – The WIDE Project*; Kluwer Academic Publishers, 1999.

[Gref00] P. Grefen, K. Aberer, Y. Hoffner, H. Ludwig*; CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises*; Int. Journal of Computer Systems Science & Engineering, Vol. 15, No. 5, 2000.

[Hoff99] Y. Hoffner; *Supporting Contract Match-Making*; Procs. IEEE 9th Int. Workshop on Research Issues in Data Engineering; Australia, 1999.

[Jajo97] S. Jajodia, L. Kerschberg (Eds*.); Advanced Transaction Models and Architectures*; Kluwer Academic Publishers, 1997.

[Java99] P. Deitel, H. Deitel; *Java: How To Program v1.2*; Prentice Hall, 1999.

[Klin00] J. Klingemann; *Controlled Flexibility in Workflow Management*; Procs12th Conf. on Advanced Information Systems Engineering, Sweden, 2000.

[Koet99] M. Koetsier, P. Grefen, J. Vonk*; Contracts for Cross-Organizational Workflow Management*; Proc. 1st Int. Conf. on Electronic Commerce and Web Technologies (EC-Web), United Kingdom, 1999.

[Kuma98] V. Kumar, M. Hsu; *Recovery Mechanisms in Database Systems*; Prentice Hall, 1998.

[Leym95] F. Leyman; *Supporting Business Transactions Via Partial Backward Recovery In Workflow Management Systems*; Procs. BTW '95.

[Leym00] F. Leyman, D. Roller; *Production Workflow – Concepts and Techniques*; Prentice Hall, 2000.

[Ludw99] H. Ludwig, C. Bussler, M. Shan, P. Grefen*; Cross-Organizational Workflow Management and Co-ordination – WACC '99 Workshop Report*; ACM SIGGROUP Bulletin, Vol. 20, No. 1, 1999.

[Ludw00] H. Ludwig, 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.

[MQWF00] IBM Corporation; MQ Series Workflow web site; `http://www.software.ibm.com/ts/mqseries/workflow/`

[Stri00] C. Stricker, S. Riboni, M. Kradolfer, J. Taylor*; Market-based Workflow Management for Supply Chains of Services*; Procs. 33rd Hawaii Int. Conf. on System Sciences, Hawaii, 2000.

[Vonk99] J. Vonk, P. Grefen, E. Boertjes, P. Apers*; Distributed Transaction Support for Workflow Management Applications*; Procs. 10th Int. Conf. on Database and Expert System Applications (DEXA); Florence, Italy, 1999.

[Vonk00] J. Vonk, P. Grefen, M. Koetsier*; Cross-Organizational Transaction Support for Virtual Enterprises*; Procs. 7th Int. Conf. on Cooperative Information Systems (CoopIS); Eilat, Israel, 2000.

[WfMC96] Workflow Management Coalition; *Interface 4: Interoperability Abstract Specification v1.0*; 1996.

[WfMC99] Workflow Management Coalition; *Terminology & Glossary v3.0*; 1999.

[Wodt96] D. Wodtke, J. Weissenfels, G. Weikum, A. Dittrich; *The MENTOR Project: Steps Towards Enterprise-wide Workflow Management*; Procs. Int. Conf. on Data Engineering, USA, 1996.

Appendix A

In this appendix, an example is presented that clearly shows the effects of the developed transaction support on the actual execution of a cross-organizational workflow process within the implemented CrossFlow system.

Figure 8 shows the part of MQ Series Workflow (MQWF) that informs the user about the state of all 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. The reason that the consumer process and the provider process are both shown to the same user is that both processes are running on the same MQWF installation. This is done for practical reasons. In reality both organizations run their own workflow management system in which case one organization cannot see the processes of the other organization.

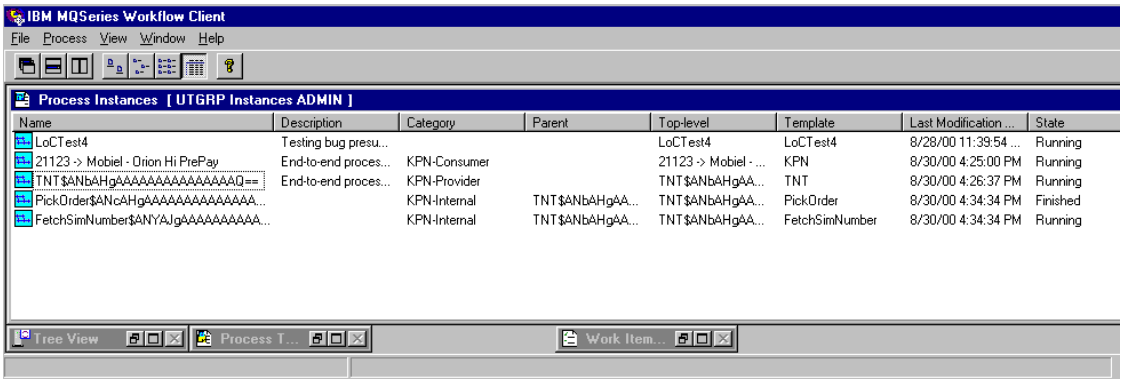

Figure 8: Process states before rollback request

As can be seen in the figure, both the consumer and provider processes are running.

Figure 9 presents the MQWF process monitor applied to 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 activities ‘WelcomeCustomer’ and ‘FetchSimNumber’ are running, while the preceding activities are finished and the succeeding activities are not yet started.

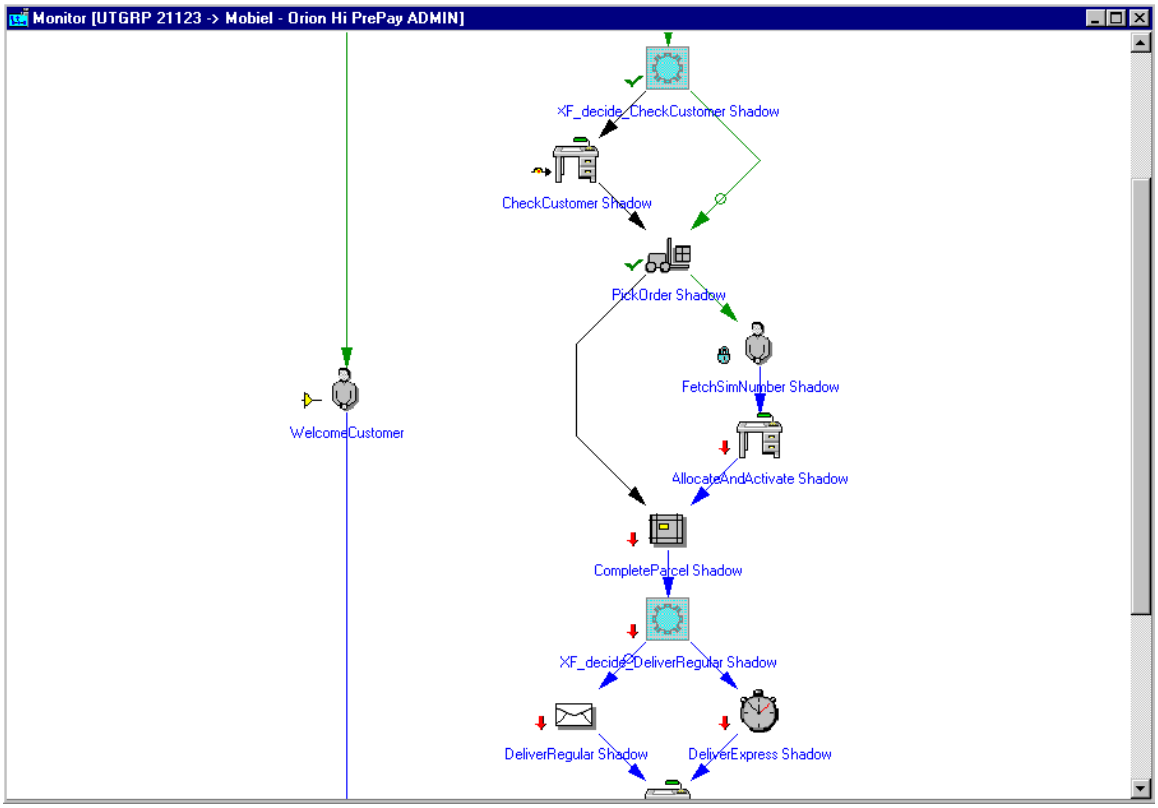

Figure 9: Process monitor before rollback request

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 7 (but with different parameters). Dealing with the rollback request, the transaction support system will first suspend the involved processes, in this case it will suspend the KPN-Consumer process and the KPN-Provider process.

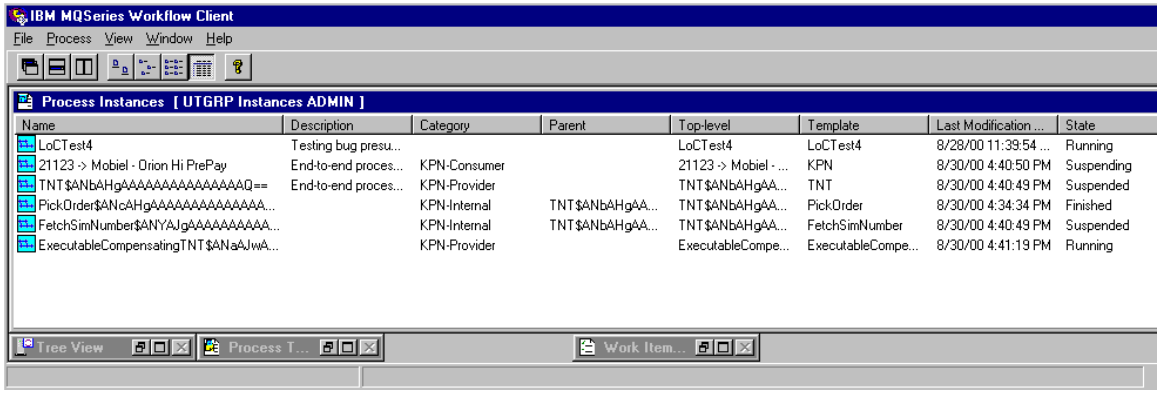

Figure 10: Process states during rollback execution

After the involved processes are suspended, a compensation process is created by the transaction support system, as explained in Sections 5 and 6, and executed by the workflow management system. Figure 10 shows the process states during the handling of the rollback request. Both the consumer and provider process are suspended and a new compensation process is running (called 'ExecutableCompensatingTNT$...'). The process category of the compensation process shows that this compensating process belongs to the provider organization and will therefore undo all activities that have been executed within the provider process.

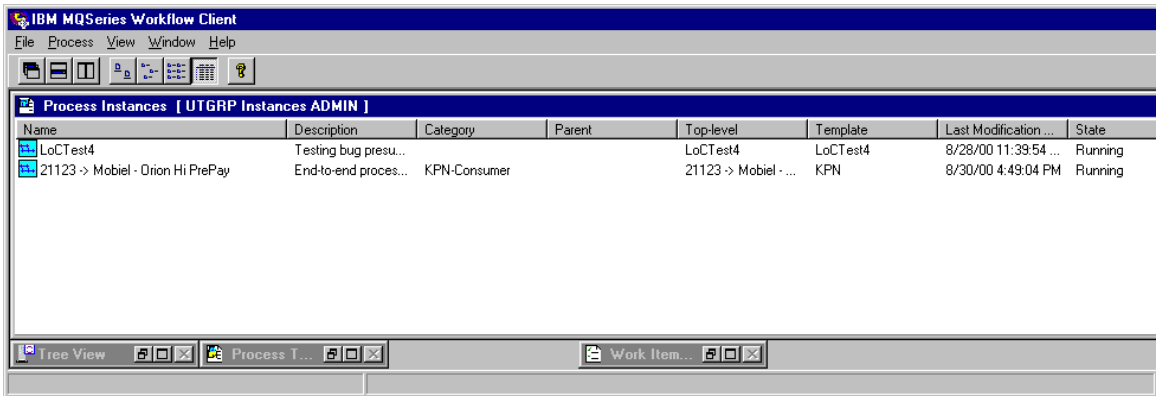

Figure 11: Process states after rollback has been performed

After the rollback request has been handled and, according to the table in Section 5, also part of the consumer process has been compensated, the original process is resumed again, see Figure 11. Because the compensation process has finished, it doesn't appear in the 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 compensation has been completed is shown in Figure 12. From this figure, it can be seen that only one activity is running ('XF-startpoint') and that all provider activities (right-hand side of the figure) have not been started yet.

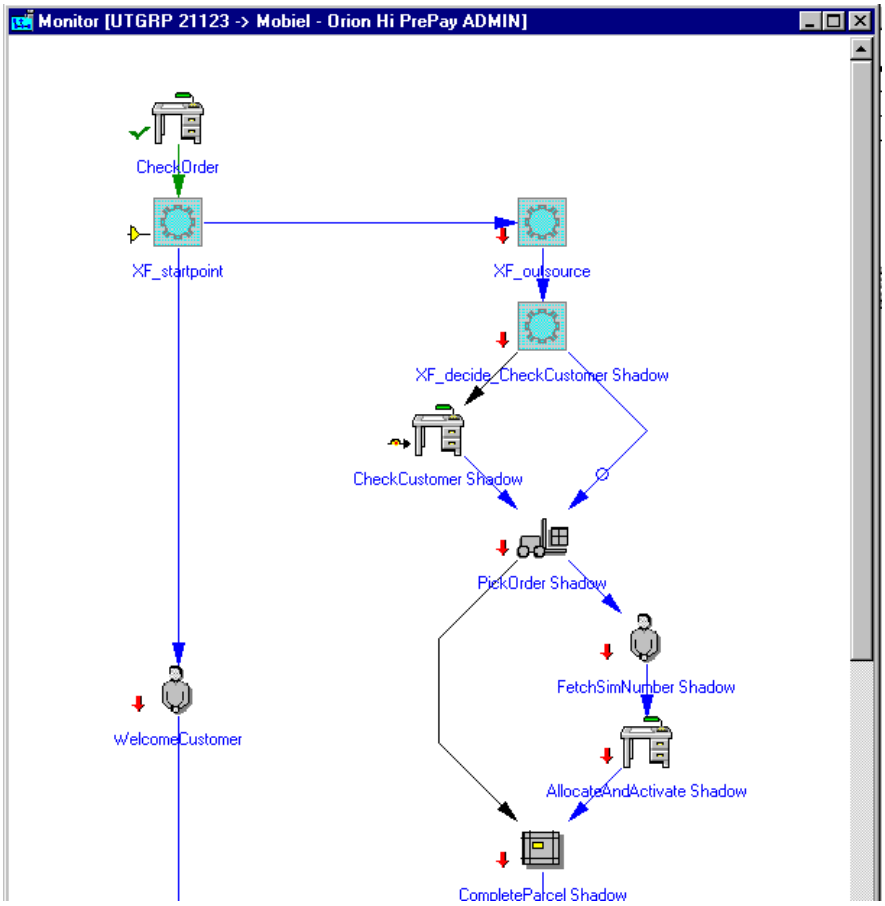

Figure 12: Monitor on process after rollback has been performed