CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises♣

Paul Grefen
University of Twente
The Netherlands
grefen@cs.utwente.nl

Karl Aberer
GMD-IPSI
Germany
aberer@darmstadt.gmd.de

Yigal Hoffner & Heiko Ludwig
IBM Research Zürich
Switzerland
{yho,hlu}@zurich.ibm.com

Abstract

In this report, we present the approach to cross-organizational workflow management of the CrossFlow project. CrossFlow is a European research project aiming at the support of cross-organizational workflows in dynamic virtual enterprises. The cooperation in these virtual enterprises is based on dynamic service outsourcing specified in electronic contracts. Service enactment is performed by dynamically linking the workflow management infrastructures of the involved organizations. Extended service enactment support is provided in the form of cross-organizational transaction management and process control, advanced quality of service monitoring, and support for high-level flexibility in service enactment. CrossFlow technology is realized on top of a commercial workflow management platform and applied in two real-world scenarios in the contexts of a logistics and an insurance company.

♣ The work presented in this paper is supported by the European Commission as ESPRIT project No. 28635.

Table of Contents

1. Introduction

In today's businesses, the application of workflow management systems (WFMSs) is widespread. The use of WFMSs ensures a well-structured and standardized management of processes within organizations [Geor95]. Traditionally, the emphasis of workflow management has been on homogeneous environments within the boundary of a single organization. In the context of close cooperation between companies, where companies combine their efforts and become virtual enterprises, processes crossing organizational boundaries have to be supported [Lud99a]. This implies extending the functionality of workflow support so that workflow management systems in different organizations can be linked to manage integrated cross-organizational processes. The advent of business-to-business electronic commerce adds a dynamic dimension to this. The extended workflow support must be able to deal effectively with heterogeneous workflow environments, well-specified levels of autonomy of partners in a virtual enterprise, and dynamic formation of new and dismantling of existing collaborations.

This report gives a detailed overview of the conceptual and technical approach in the CrossFlow project, which aims at developing support for cross-organizational workflow management in dynamic virtual enterprises using an outsourcing paradigm. The CrossFlow application scenarios are briefly decribed in this report. Below, we first outline the context of the CrossFlow project. Next, we present the details of the CrossFlow project. At the end of this section, we discuss the structure of the sequel of this report.

1.1 Dynamic service outsourcing in virtual enterprises

Nowadays, many organizations form dynamic partnerships to effectively deal with market requirements. Companies focus on their core business and outsource secondary activities to other organizations. Growing complexity of products requires co-makership relations between organizations. Value chains require a tight cooperation between companies participating in these chains. As a result, the creation of virtual organizations has become a major issue. To enable their operation, the information processing infrastructures of the participating organizations need to be linked. The workflow management systems that control the processes in the individual organizations are a key element here. Linked workflow systems should allow one organization (the service consumer) to start a process (a service) on its behalf in another organization (the service provider) and receive the results of this process. As black-box processes are too coarse for tightly cooperating organizations, advanced monitoring and control mechanisms are required to support fine-grained interaction between these organizations, while preserving their autonomy as much as possible.

1.2 The CrossFlow project

CrossFlow is a European research project of the 4th ESPRIT Framework that researches and develops cross-organizational workflow support for virtual enterprises.

The prime contractor in CrossFlow is IBM, participating with its e-business group of the Zürich Research Laboratory in Switzerland, its development laboratory in La Gaude, France, and its workflow software development group at its site in Böblingen, Germany. Technology providers in the consortium are GMD-IPSI in Darmstadt, Germany, and the University of Twente in the Netherlands, who contribute their experience in groupware [By98] and workflow management [Gre99a]. User partners are KPN Research in Groningen, the research division of the largest telecom operator in the Netherlands, and Church & General, an Irish insurance company that is part of the Allianz Group. Sema Group in Madrid, Spain, acts as industrial observer in the consortium.

The CrossFlow project started in September '98 and is planned to run for two years. The project covers the complete spectrum from requirements analysis to prototype assessment.

1.3 Structure of this report

In Section 2, we discuss the CrossFlow approach to cross-organizational workflow management, in which contracts play a central role. Section 3 presents the CrossFlow contract model and language. The CrossFlow architecture is discussed in Section 4, outlining the support for basic cross-organizational workflow management. In Section 5, we present the advanced aspects in workflow support, called the cooperation support services. Work related to the CrossFlow project is discussed in Section 6. We end this report with conclusions and a brief look into the future.

2. The CrossFlow approach

The CrossFlow project develops information technology for advanced process support in dynamic virtual organizations. Four main aspects characterize the CrossFlow approach:

Dynamic service outsourcing. The cooperation between partners is based on a dynamic service outsourcing paradigm with service consumers and service providers. Compatible business partners find each other through a matchmaking facility.

Contract-based service specification. A detailed service specification in the form of a contract is the basis for a tightly-linked cooperation implementing the service provision from service provider to service consumer. The definition of the interaction in the contract is independent of the specific enactment technology of the organizations.

Fine-grained, advanced interaction. The interaction level between service consumer and provider is at a fine grained and a high semantic level, enhanced by the availability of a set of advanced cooperation support services.

Contract-dependent generation of enactment infrastructure. The enactment infrastructure that connects the information systems of service provider and consumer is dynamically set up according to the contract and a specification how the contract is to be implemented and supervised.

Figure 1 illustrates the CrossFlow approach. In this figure, we see how the service consumer outsources its activities D and E to a service provider that can perform these activities with an additional value (hence D+ and E+). The contract is the basis for the cooperation that, apart from service invocation and result reception, also encompasses detailed monitoring and control of the outsourced activities.

The first three aspects above are of a conceptual nature. In the sequel of this section, we will discuss these in more detail. The fourth aspect is of an architectural nature and hence discussed in Section 4.

Note that the trading-based approach to service outsourcing means that CrossFlow can be considered a project investigating the intersection of workflow management and electronic commerce technology.

2.1 Dynamic service outsourcing

The CrossFlow approach to cross-organizational workflow management is based on a dynamic service consumer/provider paradigm. This means that an organization that wants a service to be performed on its behalf (the service consumer) outsources this service to an organization that can perform this service (the service provider). This outsourcing is performed dynamically, which means that the decision to outsource is taken during the execution of the process requiring the service and that the provider is chosen dynamically.

The dynamic quest for compatible business partners is performed through a matchmaking facility, which plays the role of a service marketplace. Service providers advertise their services in this facility. Service consumers query the facility for required services. Matchmaking of services is based on the fact that in many markets standard business practices, standard languages and ways of describing services, and standard legal forms and processes have evolved, resulting in common Contract Templates [Law98].

The interaction between service consumers and providers is based on contracts, as described below. Service providers advertise their services in contract templates, which are completed to individual contracts by service consumers. In Section 4, we discuss the contract-based interaction between consumer and provider in detail.

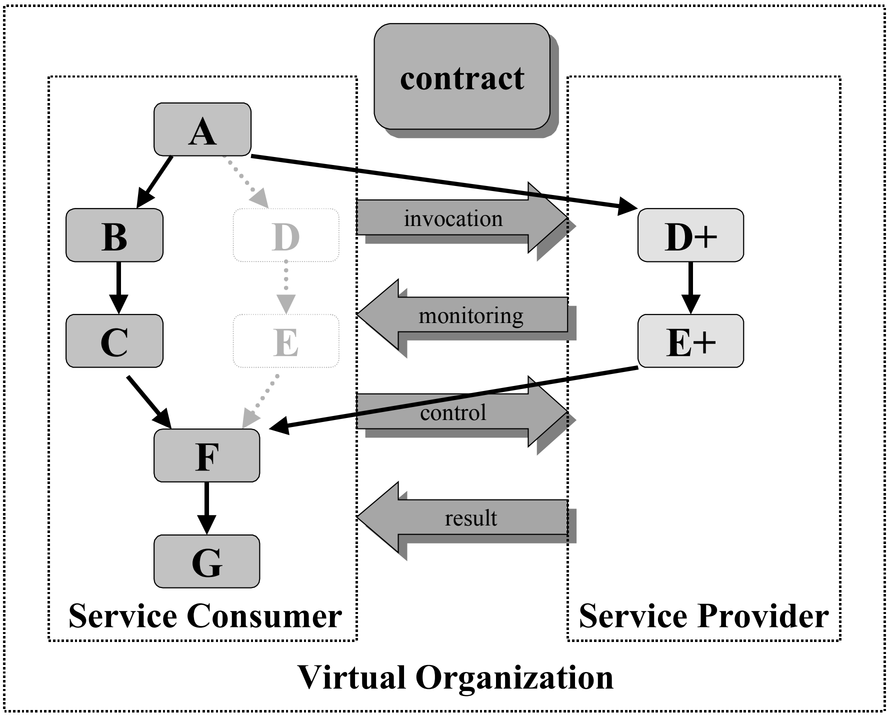

Figure 1: Dynamic service outsourcing in a virtual organization

2.2 Contract-based service specification

In the CrossFlow approach, the interaction between service consumer and service provider is completely specified in a contract. The contract defines all relevant details of the service provision. Traditionally, this is limited to an identification of the service and all parameters required to execute the service. CrossFlow contracts, however, also entail a specification of the process used to execute the service. Specification of this process allows for further integration of consumer and provider processes than a mere black-box process would allow. This high level of integration is essential for the close partnerships found in virtual organizations.

In virtual organizations, however, a partner does not require full operational details of other partners. Rather, a well-defined abstraction of their operation should be used to obtain an effective view on both data and processes. As partners in a virtual organization often have different IT platforms, a heterogeneous environment exists. This heterogeneity should be addressed by abstraction of technical details of partners. For both reasons, CrossFlow contracts define the interaction between organizations not in terms of their workflow systems, but on an abstraction level above these systems. Section 3 provides a detailed discussion of CrossFlow contracts.

2.3 Fine-grained, advanced interaction

As explained above, the CrossFlow approach is focused on tightly integrated service consumer and provider processes. For this reason, a common service process specification is included in CrossFlow contracts. To support the tight coupling of processes, however, advanced notions of interaction are required. These notions are operationalized in so-called cooperation support services (CSSs). A broad spectrum of CSSs is relevant for cross-organizational workflow management, like remote process monitoring and control, cross-organizational transaction management, automatic service remuneration, trust and security management, etc. The design of these services should be such, that they can be selected and combined in a modular way, depending on the application context.

In the context of the CrossFlow project, three areas of advanced cooperation support services are addressed. The selection of these three areas is based on the interest and background of the project partners. Quality of Service monitoring allows tracking the progress of outsourced services, both online during service execution and offline to provide aggregate information. Level of Control enactment provides means for high-level cross-organizational transaction management and consumer-controlled process control over outsourced services. Flexible Change Control allows for dynamic changes to execution paths of services during their execution.

3. Contracts for cross-organizational workflow management

Contracts between service consumer and service provider organizations form the basis for cross-organizational workflow cooperation in the CrossFlow project. Using the requirements derived from user scenarios, a conceptual contract model has been developed. We discuss this model in Section 3.1. As an operationalization of the model, an XML-based contract specification language has been defined. This language is presented in Section 3.2. A more elaborate discussion of contract model and language can be found in [Koet99]. In Section 4, we show how contracts are enacted.

3.1 Contract model

The CrossFlow contract model provides the conceptual structure that describes the tight collaboration of service consumer and provider in a virtual organization. The model has been designed to fit the requirements that follow from the approach outlined in the previous section. More specifically, the design of the model includes concepts for representing the structure of the outsourced service process described by the contract, high level concepts for monitoring and controlling this process in a cross-organizational context, and concepts for flexible use of contracts. Below, we discuss the global contract model and elaborate one submodel to illustrate the detailed structure of the model.

Global contract model. A modular contract structure has been chosen that allows easy adaptation and extension to specific environments. The model consists of the following parts:

Concept model. All concepts that are used in the contract must be defined clearly, 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 contract.

Process model. The process model describes the internal structure (schedule) of the workflow process implementing the service at the contract level. The process schedule is composed of process elements, i.e., the individual activities and transitions. The process is modeled in a way that allows the provider to map it to its internal process and allows the consumer to understand the sequence of events for monitoring purposes and make decisions based on this knowledge for control purposes.

Enactment model. The enactment model provides concepts to represent the advanced cooperative support that is offered during service enactment. Cooperative enactment support can be composed of a number of elementary services, like service execution monitoring, service execution control, remuneration support, authentication support, etc. The required set of services is application-dependent. Details of cooperation support are discussed in Section 5.

Usage model. The usage model defines manners in which the contract can be used. 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.

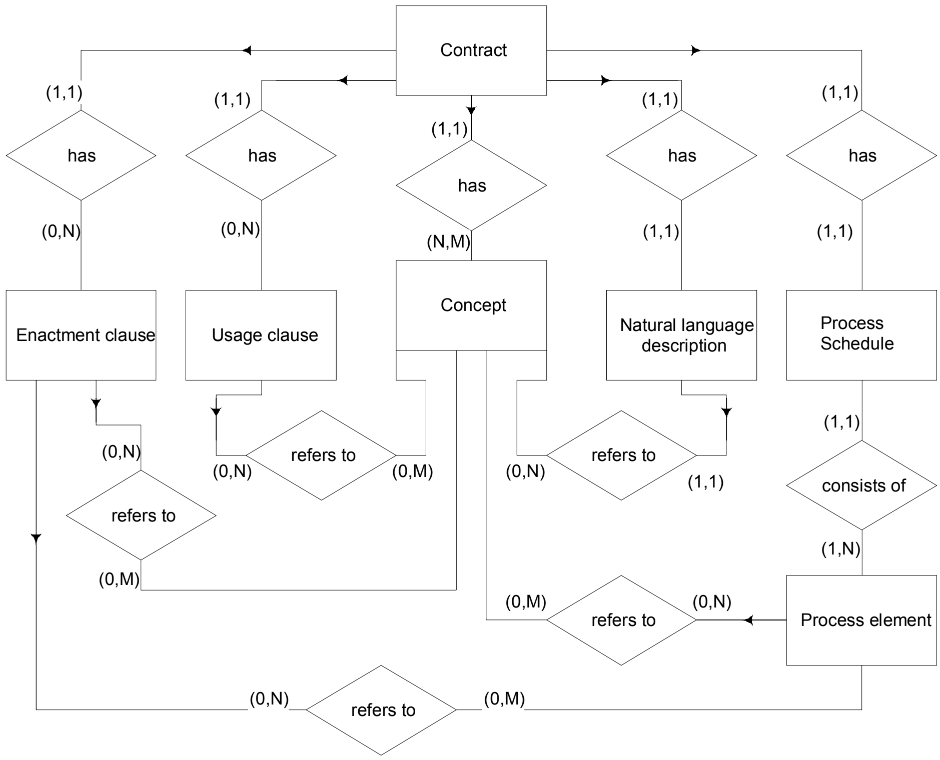

Figure 2: Global structure CrossFlow contract model

Figure 2 shows the global structure of the contract model in an EER diagram. This diagram clearly shows the modular structure of the contract model. The model includes set structures for contract concepts, enactment clauses, usage clauses, and process elements. This caters for flexible and extensible contract structures. The various reference relations in the model cater for semantic coupling between the elements.

Process submodel. The various elements in the contract model are further detailed to describe their internal structure. To illustrate this, we show a simplified version of the internal structure of the process submodel in Figure 3. This submodel provides the necessary structures for the specification of the service process at the contract level. As processes are modeled in an abstract fashion, the process model is relatively simple when compared to full-blown workflow models. The process model is mapped to concrete workflow models in consumer and provider workflow processes.

3.2 Contract language

For easy specification and communication of contracts, a contract specification language is required that provides a textual format for the contract model introduced above. In the CrossFlow project, XML has been chosen as the basis for this language. Choosing a standard meta-language enhances general acceptance of and interoperability with the CrossFlow contract language.

XML is a markup language for documents containing structured information [XML, Grah99]. A Document Type Definition (DTD) is used to define the structure of a document, consisting of the tags that can be used for the document type, the order of these tags and the attributes that can accompany those tags. A complete XML document can be mapped to a tree structure, with hierarchically ordered elements. This hierarchic structure enables easy access to relevant data for the contract enactment infrastructure.

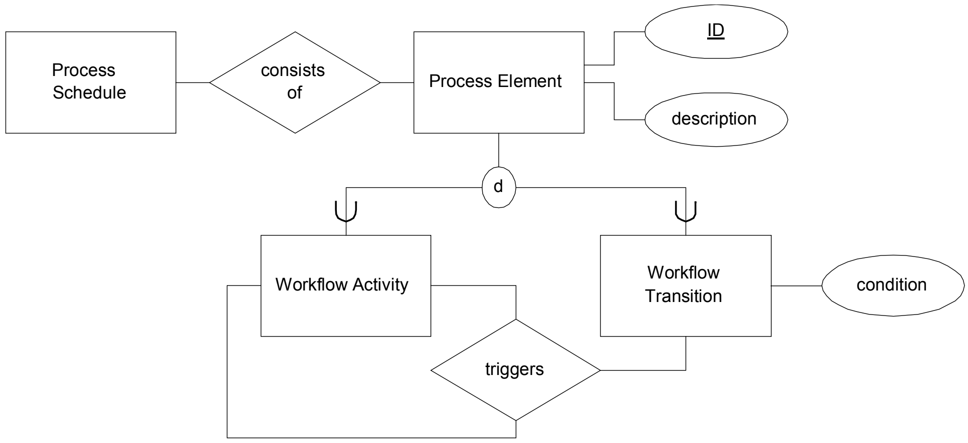

Figure 3: Process submodel of CrossFlow contract model

Below, we discuss the overall contract language first, then elaborate the process specification sublanguage.

Global contract specification. The contract model has been mapped to a contract DTD, establishing the content and structure for all contracts. The top level of the contract DTD has the following specification, coinciding with Figure 2:

```
<!ELEMENT Contract (DataSection, NaturalLanguageDescr?, WorkFlow,
                     Enactment, UsageClauses? )>
```

This DTD fragment establishes that the contract consists of a data section, a natural language description, a workflow specification, an enactment specification, and optional usage clauses. Each element in the top level of the contract is specified into detail in turn, defining a complete structure that all contracts must conform to.
However, a DTD alone cannot offer adequate constraints for specifying a complete contract language. Additional constraints have been identified for the contract language, for example constraints that relate the attribute values of an element to its content. XML in itself cannot impose constraints on text content within elements. These additional constraints form together with the DTD the complete contract language.
In the mapping of the contract model to the contract language explicit design decisions have been made to make the contract operationally more suitable. Some features that are defined as attributes in the contract model have been implemented as individual elements in the XML structure to enable easy processing by the CrossFlow enactment infrastructure. More specifically, we have defined explicit contract tags to prevent searching through tree structures for specific value in a contract specification.

Process specification. Below the DTD specification of the process model is given. This structure is obtained by mapping the conceptual process submodel (the simplified version of which is depicted in Figure 3) into XML.

```
<!ELEMENT WorkFlow (Activity | Transition )+ >
<!ELEMENT Activity (Name, Description?) >
<!ATTLIST Activity ActID ID #REQUIRED>
<!ELEMENT Name (#PCDATA) >
<!ELEMENT Description (#PCDATA) >
<!ELEMENT Transition  (Name, Description?, To, From, Condition?)>
<!ATTLIST Transition  TransID ID #REQUIRED>
<!ELEMENT To   (ActRef) >
<!ELEMENT From (ActRef) >
<!ELEMENT Condition (#PCDATA | ParamRef | ActRef)*>
<!ATTLIST Condition type (parameter |workflow.state |workflow.event |
                          workflow.time | CDATA) #REQUIRED>
<!ELEMENT ParamRef EMPTY>
<!ATTLIST ParamRef ParamID IDREF #REQUIRED>
<!ELEMENT ActRef   EMPTY >
<!ATTLIST ActRef   ActID   IDREF #REQUIRED>
```

Because of the fact that processes are described at an abstract level, this is a relatively simple DTD fragment. Activities have an identifier attribute, to enable referring to them in the rest of the document. Conditions are attributed with a type in order to enable configuring the proper evaluators in the infrastructure. The conditions can refer to both activities and parameters to form expressions that refer to a particular state of an activity or a parameter value. A separate condition language is one of the additional elements of the contract language, defining the constructs that can be used. To avoid too complex XML structures, the condition language is not based on XML but on a straightforward expression syntax.

4. The CrossFlow architecture

In the previous section, we have discussed model and language for specifying CrossFlow contracts. Now we turn to the CrossFlow architecture that handles contract-based cross-organizational workflow management. The CrossFlow architecture supports both contract making and contract (service) enactment. The architecture is based on commercial workflow management system technology, shielded from the CrossFlow technology by an interface layer. In the project, IBM's MQSeries Workflow (formerly FlowMark) workflow product is used [IBM00].
The lifecycle of a service outsourcing consists of four phases:

1. Contract establishment;
2. Dynamic infrastructure configuration;
3. Contract enactment;
4. Dynamic infrastructure disposal.

We describe each of the four phases in detail below. We conclude this section with a discussion of technical details of the prototype implementation.

4.1 Contract establishment

Advertising and searching for a matching business partner in CrossFlow is supported by an advanced service trader (matchmaking facility) based on the CORBA Trading Service [OMG96]. The trader provides an extension of the CORBA trading service's 'property and constraint' language by which what each party offers to and demands from the other party can be expressed. Unlike the CORBA trading service, it supports a bi-directional matchmaking process where both customers and providers can describe what they offer and require of each other [Hoff99a]. This is particularly suitable for service contract matchmaking where both the provider and the consumer offer certain service characteristics while at the same time placing demands on the other party [Hoff99b].
The following describes a typical sequence of events that leads to the establishment of a contractual relationship between the provider and consumer organizations – illustrated in Figure 4.
When the provider WFMS is ready to receive requests for enactment of a process on behalf of a consumer organization, it notifies its Contract Manager of its readiness. A Workflow Module (WM) acts as an interface layer to shield the Contract Manager from details of specific WFMSs. The Contract Manager selects a pre-existing Contract Template that describes the service and its associated QoS guarantees, work schedule, monitoring and control points as provided by the service, etc. Appropriate values for these service guarantees including the cost of the service must then be determined. These will be decided according to the capabilities of the enactment infrastructure, the resources that the provider is willing to assign to the enactment, and the price associated with the resources. In addition, the requirements that the provider places on the consumer within the terms of the Contract Template are also specified. The service description and the demands are translated into the property and constraint language of the matchmaking facility. The result is then advertised into the trader that serves the specific market. In a competitive market, several provider organizations will advertise the same service with the same associated service contract but with different values describing QoS, scheduling and other guarantees, and the price of the service.

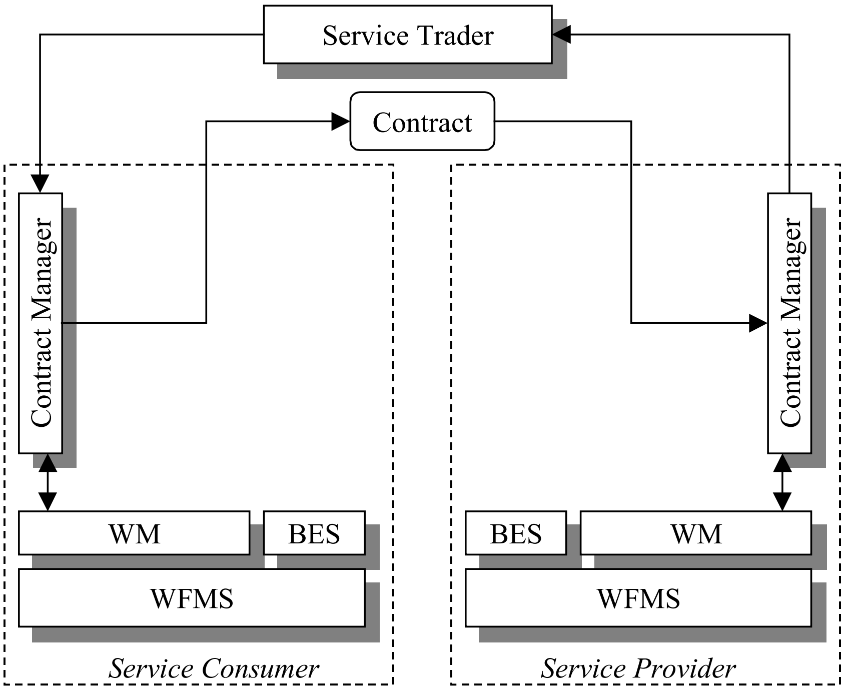

Figure 4: Contract establishment in the CrossFlow architecture

When the consumer WFMS reaches a task that it wishes to have enacted on its behalf externally, it notifies its Contract Manager (again through a Workflow Module). The consumer Contract Manager selects a pre-existing Contract Template that describes the service it is looking for in terms of the QoS guarantees, work schedule, monitoring, and control points it wishes to have associated with the provided service. Unlike the provider who specified those parameters as properties, the consumer can place demands in terms of the speed by which it wishes to have the work completed and the maximum price it is willing to pay for it, for example. The consumer must also describe what it offers in terms of its willingness to pay and the means by which it can pay, for example. The consumer's promises and demands are translated into the property and constraint language of the trader. The result is then sent as a search query into the trader serving the market.

The trader compares the promises and demands made by the consumer against the offers previously posted in it by market providers. The matching offers are then sent back to the consumer. The consumer Contract Manager can then compare the offers and select the one that suits its requirements best. By notifying the selected provider, the consumer in effect makes a counter-offer that the provider can accept or reject. The acceptance of the counter-offer signifies an agreement between the two organizations. Although this is outside the scope of the CrossFlow project, the agreement between the organizations can be digitally signed, making the contract an explicit legal entity.

4.2 Dynamic infrastructure configuration

Once a contract has been made between service consumer and provider, a dynamic contract and service enactment architecture is set up in a symmetrical way for both partners, as illustrated in Figure 5. For this purpose, the Contract Manager activates the Configuration Manager. The configuration of this enactment infrastructure is based on the contract and requires a number of components:

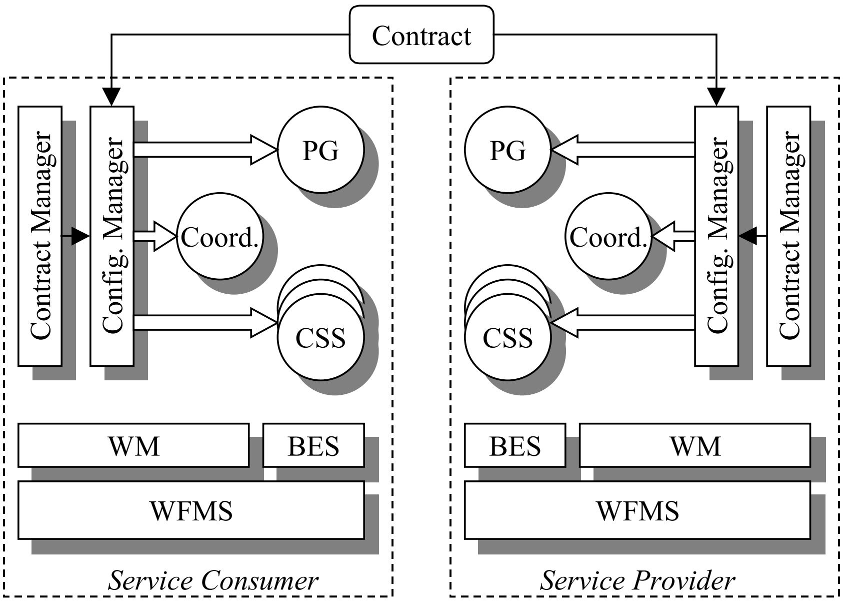

Figure 5: Dynamic configuration of CrossFlow architecture

Cooperation Support Services (CSS) implement the advanced cooperation support introduced in Section 2. Three areas of cooperation support are addressed by the CrossFlow project. Details of CSS modules are discussed in Section 5.

Proxy-Gateways (PG) deal with the crossing of domain boundaries (see section 4.1), by facilitating the interaction between the organizations' systems, by translating between the internal-external and organizational differences on a syntactical level, and by monitoring and controlling exit-entry to protect the organization's integrity and security.

Coordinators (Coord.) are used at each site to connect the various components such as the CSSs, the PG, and the WfMS through the WM.

The functionality of contract and service enactment components is largely dependent on the contents of the contract and the manner in which each organization sees fit to carry out their part of the enactment.

The Internal Enactment Specification (IES) is the organization-specific blueprint that specifies how the contract is to be enacted. It defines which internal resources can be used in which way. For this purpose, the IES describes in which components are needed to enact the service and, in addition, it describes the contract implementation policy for each of the deployed CSS components. It also provides the mapping between the external and internal details of the business process and its related data.

Using the contract and the according IES, the Configuration Manager instantiates, configures, and links a coordinator, a PG and a set of CSS components to enact the contract.

4.3 Contract enactment

When the set-up described above is ready, the consumer can initiate the enactment of the outsourced business process by contacting the provider. For this purpose, the various dynamically constructed components are linked together as illustrated in Figure 6 (in a simplified way).

Any monitoring information agreed upon in the contract to be provided from the provider to the consumer can either be sent as a notification or requested by the consumer. As a result of the progress update, the consumer may wish to request the provider to modify the enactment of the business process. This may include a change of parameters or a change in the process direction

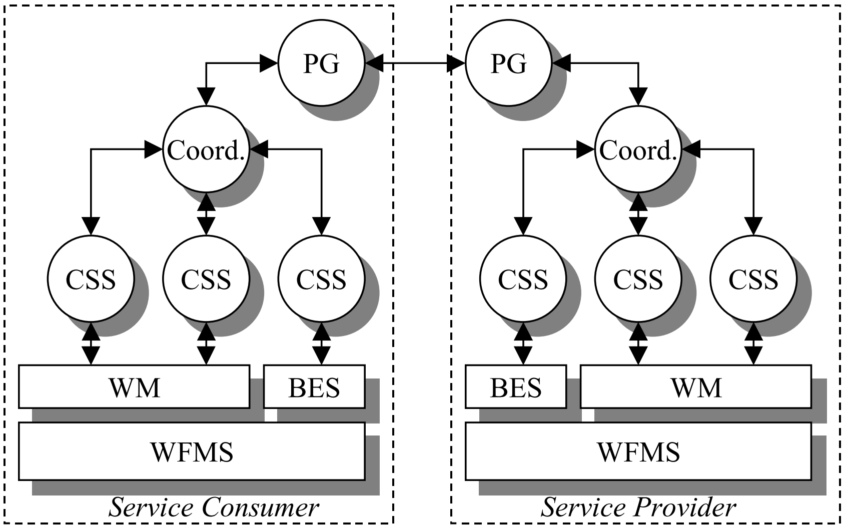

Figure 6: Service enactment in CrossFlow architecture

or structure, depending on the contract. Further monitoring information may pass as a result and more changes may be initiated where necessary. Ultimately, an indication of the completion of the process and its results will be passed to the consumer.
Where appropriate, the enactment infrastructure can access Back End Systems (BES) for specific services. These systems offer CrossFlow services on a permanent basis (not related to the enactment of a single contract) and other more general services. We will see example of these systems in Section 5.

4.4 Dynamic infrastructure disposal

When all the administrative processes have been completed and both sides are satisfied with the provision and consumption of the service, the infrastructure created earlier can be dismantled. This means that coordinator, CSS modules, and proxy gateways relating to the service can be deleted.

4.5 A technical view on the architecture

The CrossFlow system architecture faces a number of requirements to implement its general applicability for companies that dynamically establish process relationships. The most important of these are:

- The architecture must be able to accommodate different WFMSs and an open set of back-end systems (BES), as they are required.
- Long-running components such as the WM must be integrated with components that only run while establishing or enacting a particular contract, e.g. the coordinator.
- The architecture must cater for different forms of contract managers, i.e., automatic ones that are driven by the outsourcing request identified in the WFMS and user-driven interactive applications.

Figure 7 provides an overview of the CrossFlow system architecture. We distinguish four different environments that are independent system parts that can run on separate computers:

- The *enterprise system environment* comprises the WFMS of the organization, other enterprise resources, and the long-running CrossFlow components that access them. These are in

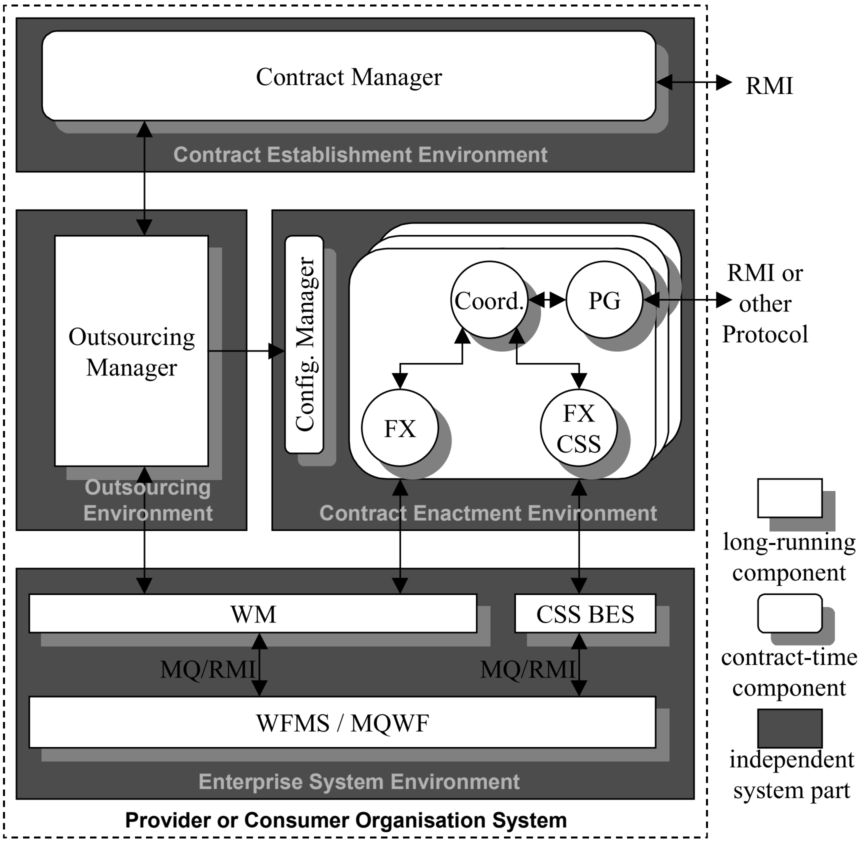

Figure 7: Technical view on the CrossFlow architecture

particular the Workflow Module and the long-running parts of the CSSs. Long-running in the context of this system architecture means beyond the lifetime of a single contract.

- The *outsourcing environment* contains the outsourcing manager. The outsourcing manager provides the link between the long-running enterprise system environment and the contract-time components. It has to interact with the contract manager to obtain a service contract and its related IES that it can pass on to the configuration manager that uses it to create the contract enactment infrastructure for this particular contract.
- The *contract establishment environment* comprises the contract manager. As discussed before, this component can be embodied in many different ways. The two major approaches are (on the consumer side) to either have an automatic decision-making mechanism that can simply be invoked by the outsourcing manager, or to have a contract manager that is a workflow client application that supports an employee in the decision-making for a service provider.
- The *contract enactment environment* includes the configuration manager and the enactment infrastructures for the respective contracts.

The separation into different environments allows system configurations that put the enactment infrastructures on a "service gateway" [Lud99b] computer while the contract manager can be an applet or a GUI application that resides on a decision maker's PC. All environments are written in Java and interact using RMI [Farl98]. Within the enterprise systems environment the CrossFlow components use the necessary means to access WFMS or ERP systems. In the case of MQWF this means MQ Series through an RMI gateway.

5. Cooperation support services

The cooperation support services in CrossFlow provide functional support to enhance the interaction of the cooperating autonomous workflow processes. They provide means to observe and control the behavior of the processes with which an interaction is performed based on a contractual relationship. Those behavioral aspects can relate to the technical execution of the service, like transactional properties or quality of service, as well as to the legal and economic aspects of service execution, like security management, authentication and trust management, and remuneration support. The decision to focus on issues relating to the behavioral aspects at the technical level of process execution has been made on the basis of available manpower and background of the project partners. The CrossFlow approach and architecture allow easy addition of more support types.

Three areas of cooperation support are addressed by the CrossFlow project: Level of Control support, Quality of Service monitoring, and Flexible Change Control. We discuss each of these in detail below.

5.1 Level of Control

The Level of Control (LoC) cooperation support service caters for process control in cross-organizational workflow execution. For this purpose, it addresses both implicit and explicit process control:

Implicit process control is offered in the form of advanced cross-organizational transaction management to provide reliable cross-organisational workflow executions. Transaction management is based on the X-transaction model that we discuss below.

Explicit process control is offered as support for process control primitives that provide means for the consumer to have control over the provider's workflow execution. Process control support relies on transaction management for its operation.

We discuss both transaction management and process control models below and explain how these are supported in the CrossFlow architecture as presented in Section 4.

X-transaction model. Transaction management in CrossFlow is based on a three-level transactional workflow process model, called the X-transaction model (see Figure 9). This model distinguishes three levels in a cross-organizational workflow process that have transactional semantics. These three levels are the outsourcing level, the external level and the internal level. The external process level defines the process as it is described in the contract. As such, it is the common view shared by service consumer and provider. The outsourcing process level defines the process at the consumer side in which the process in the contract is embedded. Activities in the contract are represented by placeholders (dummy activities) at the outsourcing level. The internal process level at the provider side defines the implementation of the process at the external level, i.e., it describes the process in the contract at a lower level of abstraction and at a higher level of detail. The internal level is encapsulated towards the service consumer by the external level.

To provide process rollback, the X-transaction model is based on the compensation principle as originally described in [Gar87]. Compensating activities that semantically undo the effects of the originally executed activities, are used to undo (roll back) completed activities of a workflow execution. Upon a rollback request, a compensating process is dynamically generated from the predefined compensation activities. A rollback request can indicate which parts of a workflow need to be undone, i.e. partial or complete, intra- or cross-organizational. The X-transaction model can be considered an elaboration of the WIDE global transaction model [Gre99a, Gre99b] with distribution aspects [Vonk99] to cover the cross-organizational dimension.

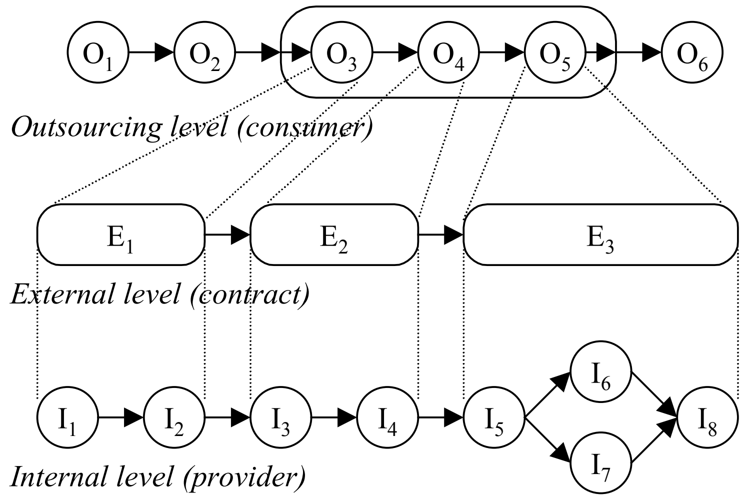

Figure 8: CrossFlow three-level process model

Compensation in X-transactions can be requested at each of the three process levels. Depending on the nature of the compensation, a compensation request on one level can initiate compensation at other levels. This is illustrated by the following example, based on the process in Figure 9. If the provider requests a rollback at the internal level from activity I_6 to I_3, the rollback has to be reflected at the external and outsourcing levels. If the rollback would be to activity I_5, the effect could remain encapsulated within E_3.

PCP support. The consumer organization has a number of Process Control Primitives (PCPs) available to allow control over the outsourced workflow process that is being executed by the provider. Allowed modes and locations of control with respect to the process are explicitly specified in the contract. The following process control primitives are offered by PCP support:

Stop: the execution of the outsourced process is requested to stop.

Continue: the execution of the outsourced process is requested to resume (after it has been stopped).

Abort: the execution of the process should is requested to be aborted, but the decision what to do with the already executed part is left to the provider.

Rollback: (part of) the outsourced workflow process is requested to be compensated.

Change variable: the value of a process variable (defined in the concept model) is changed.

Because of the distributed nature of cross-organizational workflow processes, additional parameters for the PCPs are necessary to indicate when the PCPs should be executed at the provider side.

Architecture. To cater for intra- and cross-organizational rollbacks, the CrossFlow transaction manager consists of an Intra-organizational and a Cross-organizational Transaction Manager module (ITM respectively XTM). The XTM is realized as a dynamic CSS module, as its functionality is closely related to contract execution. The ITM, on the other hand, has functionality of a more persistent nature. Therefore, the ITM is implemented as a Back End System (see Section 4). PCPs are supported by a separate dynamic CSS module. All three modules have a close cooperation through the Coordinator module as shown in Figure 6.

5.2 Quality of Service

Quality of service parameters on the execution of workflow processes relate to different dimensions, like the time needed to execute a service, the quality of the results produced by the serv-

L_1 = [*A*(0,2), *B*(2,3), *D*(3,4)]
L_2 = [*A*(0,4), *C*(4,8), *D*(8,10)]
L_3 = [*A*(0,4), *C*(4,8), *B*(8,11), *D*(11,12)]
L_4 = [*A*(0,3), *C*(3,7), *B*(4,8), *D*(8,10)]

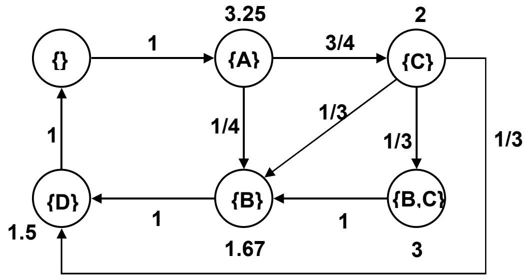

Figure 9: A set of logs and the derived Markov model

ice, and the cost in resources or money incurred by the service execution. The parameters always relate to the state of the process execution, not to data related to the invoked applications. Constraints on those parameters are specified within the contracts, for example by limiting the allowed parameter range.
A workflow that is outsourcing part of its activity needs to know about the proper execution of the outsourced workflow part. This is achieved by observing the actual values of the quality of service parameters. From the observation it can

- verify the correct performance of the outsourced workflow part with regard to the constraints agreed in the contract,
- decide on appropriate reactions in case deviations occur, and
- predict from earlier service monitoring the future behavior of services.

On-line monitoring. To observe the quality of service parameters, the contract specifies the events observable by the consumer workflow. The observable events are activity state transitions occurring in the workflow execution. Thus an observable event is described by the unique activity identifier, the activity state identifier, the activity input and output parameters and a timestamp when the event has occurred. Alternatively, instead of a timestamp a Boolean value indicating whether the event occurs at the current point of time can be given. The basic activity state model consists of the states *not active, active, terminate*. It can be extended within the contract if required. Monitoring conditions can be specified that restrict the monitoring of certain activity parameters. By default, monitoring is performed in a pull mode by the consumer. To enable a push mode, notifications can be specified in the contract by means of simple ECA rules.
The information obtained during on-line monitoring can be either used for immediate reactions or be stored for later analysis. Immediate reactions can be performed by informing the Flexible Change Control and Level of Control modules. In the following, we discuss the advanced analysis functions provided within the CrossFlow monitoring component allowing a posteriori analyses based on the log files of observed events.

Off-line monitoring. In the log file the externally observable events of a service together with time stamps are collected. From this information, a stochastic model of the observed workflow is built. This model is used for making predictions on future executions of the service. These predictions are required by the planning algorithm of the Flexible Change Control (discussed in Section 5.3).
A standard way to model stochastic processes in time are continuous time Markov Chains. Markov chains describe state transition systems by assigning state transition probabilities that are independent of the previous history. Continuous Markov chains extend this model with mean residence times for the states.
For deriving a continuous time Markov model, we have to first to identify the states of the Markov model. Since a workflow can have activities executing in parallel, we have to use sets

of (observable) activities as process states. Thus the current state of the continuous time Markov model indicates which workflow activities are active at certain point in time. From the event log the transition probabilities and the holding times can be derived for each state. This is done by incrementally building up a matrix of transition probabilities between the states and a list of holding times for the states. An incremental approach is crucial to limit the complexity of the algorithm, since only a small fraction of all possible states (i.e. subsets of observable workflow activities) will occur. Details of the algorithm are found in [Klin99].

Besides using this model in future outsourcing decisions, the model can also be readily used for analyzing a number of time-related properties of workflows and subworkflows. Depending on the log information used, this can be done at different aggregation levels, e.g. for services of a specific provider or across all providers of a service.

5.3 Flexible Change Control

In general, business processes have certain goals that can be qualitative and quantitative in nature. Qualitative goals are the performance of certain tasks observing dependencies among different tasks. Quantitative goals are the performance of certain tasks within a given time and cost frame and with a given quality. The models used by traditional workflow management systems are centered around the structural aspects of processes. Thus they are very suitable to explicitly represent the qualitative goals of a business process. Quantitative goals, however, can only be represented implicitly.

In the setting of cooperative workflows two factors come into play that lead us to a reconsideration of the traditional workflow modeling approach. Firstly, often alternative service providers with the similar or same functional but different performance characteristics are available from which the most suitable can be chosen. Secondly, due to the cooperation with autonomous partners, much less organizational control can be exerted on the execution of services by an autonomous provider, such that, for example, errors and delays are more frequent. Both factors contribute to a multiplication in the number of execution alternatives that can be considered during the workflow execution and thus have to be explicitly represented in a structural workflow model. In addition when choosing from the alternatives more often quantitative criteria are used as a decision basis than qualitative criteria. Moreover, while the quantitative information on the service performance and alternative service providers is local to the current state of a workflow execution, the quantitative goals of a business process are typically global in nature, for example, the total cost or time needed for the overall workflow.

Thus we find a situation where local decisions on the next step in the workflow execution have to be frequently made based on quantitative considerations that are related to the overall execution characteristics of the workflow. To address this problem, CrossFlow provides a flexible workflow model. This model allows to express explicitly the global goals of the business process, modeled by the workflow as part of the workflow specification (Quality of Service Goals), and to identify the places in the workflow, where alternative execution paths are possible, and where the choice of the alternatives depends on quantitative aspects of the current state of the workflow execution only (flexible workflows)

Taking the actual decision, i.e. optimally selecting the next steps for reaching the global workflow goals under a given workflow state is then done by the FCC support modules, that provide efficient planning algorithms and can exploit available knowledge on the requested services. This knowledge is derived both from the specifications given in the contracts and the performance models derived from off-line monitoring. In the following, we give some more technical details on the approach.

Flexible Workflows. We base the flexible workflow model on a standard workflow model, providing the usual constructors, including OR-split, OR-join, AND-split and AND-join. This model is extended with additional constructors that allow the provision of execution alternatives [Klin00]. These alternatives are specified in the FCC enactment clauses of a contract. We call

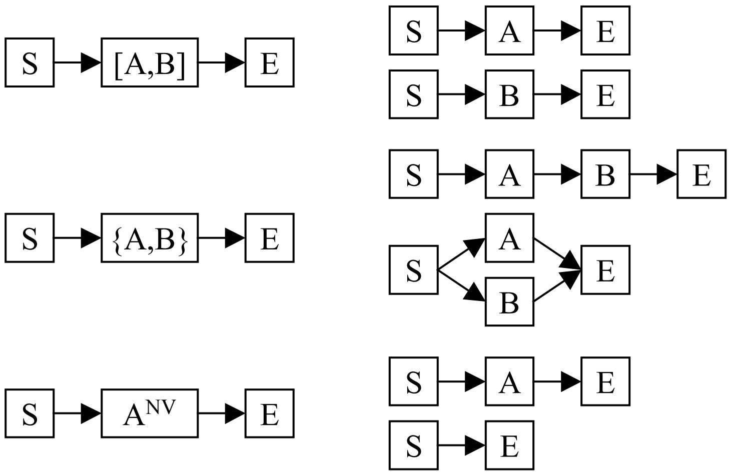

Figure 10: Resolution of flexible workflows

the additional constructors flexible elements. A flexible element is a construct that allows different execution alternatives, without requiring decision predicates for the selection of the alternatives (in this sense it is different from OR-split and OR-join). Each alternative represents a correct execution of the workflow. The following flexible elements are provided:

Alternative activities: This flexible element allows specifying different alternatives for activities of which exactly one can be chosen. Those alternatives can for example be used to provide time expensive high-quality options as well as quick low-cost options, which are selected then based on the execution state of the workflow instance.

Non-vital activities: This flexible element allows specifying that a specific element can be omitted in the workflow execution. Non-vital activities are those that in extreme situations are beneficial to be sacrificed in favor of other higher priority goals.

Optional execution order: This flexible element allows specifying a preferred ordering of elements that can be overridden. Similarly, as with non-vital activities re-ordering can be beneficial if in this way other goals of higher priority can be achieved.

In Figure 10 we depict the new constructs and the possible resolutions of the flexible workflows to standard workflows. Which resolution is chosen is part of the decision algorithms of the FCC module that are described in the following.

Quality of Service Goals and Planning. Quality of Service goals can be expressed for all quality service parameters along the different quality of service dimensions. As described earlier, quality of service parameters can relate to time, e.g. the duration of an activity or the duration of a workflow, monetary cost of executing an activity, or result quality. We distinguish between *elementary quality of service parameters*, i.e. those that can be directly observed from a running workflow instance using the monitoring capabilities described in Section 5.2, and *derived quality of service parameters,* that are composed from other elementary or derived quality of service parameters by using a composition function. An example of a derived quality of service parameter is the total execution time of a workflow. A quality of service goal is then given by three parameters:

- The quality of service parameter used.
- A fulfillment function as a mapping from the domain of the quality of service parameter to the interval [0,1], specifying to which extent a certain value of a quality of service parameter satisfies the quality of service goal.

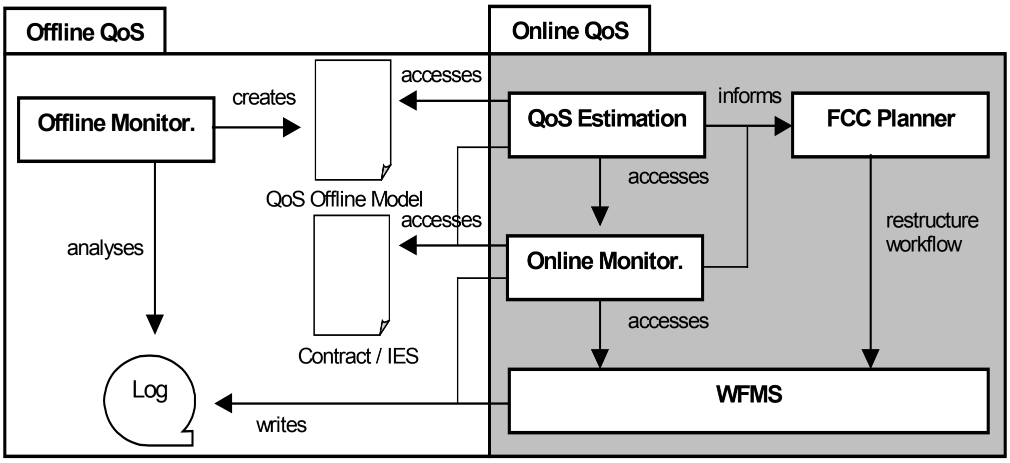

Figure 11: Interaction of QoS components

- A weight factor, specifying the relative importance of the quality of service goal with respect to other quality of service goals.

During the execution of a flexible workflow that is equipped with quality of service goals, the FCC module chooses those execution alternatives for flexible workflow elements, that optimize the weighted sum of the fulfillment functions of all quality of service goals. To compute this value for the already executed activities the quality parameter values can be obtained from online monitoring, while estimations for the future alternatives can be obtained from the performance models derived from offline monitoring.
Choosing the right alternatives poses in fact a complex optimization problem. Different optimization strategies are developed in CrossFlow to provide a solution to this. These strategies include deterministic search, which is inherently a NP-hard problem in which some efficiency improvements can be obtained with pruning, and heuristic algorithms that trade accuracy of prediction for faster execution time. Details on these algorithms are presented in [Klin00].

The QoS/FCC management architecture. We summarize the interplay between the CSS components for quality management in Figure 11. The *QoS estimation component* provides predictions of behavior of the currently running workflow instances. These estimates are based on performance models given as continuous time Markov models and produced by the *offline monitoring component*, which analyzes past executions of workflows, and on the behavior observed by the *online monitoring component*. The information on the current state of the workflow in conjunction with the performance model allows calculating probabilities for the future fulfillment of QoS parameters. In case the QoS estimation detects significant delays or deviations, the *planning component* of the FCC CSS is informed. It then has different options to restructure the workflow in order to achieve the most important QoS goals. The restructured workflow is then executed by the underlying WFMS.

6. The CrossFlow application scenarios

In the CrossFlow project, two real-world application scenarios are used to demonstrate the approach: an insurance scenario and a logistics scenario. We present both scenarios briefly in this section.

6.1 Insurance Scenario

The Church and General scenario focuses on the Motor Damage claims function, which demonstrates a real life application of cross flow of work between contractual parties, due to the number of outsourced parties involved. The relationship between Church and General and the other players in the scenario is based on the operation of a scheme known as the Emergency Service which is available to private motor policyholders. This scheme operates via an emergency helpline and a network of approved garages and loss assessors/adjusters. The scheme is coordinated on a day to day basis by a consulting engineers firm.

The application of CrossFlow technology in the insurance context is beneficial in the use of third parties such as motor assessors, property loss adjusters and solicitors. These parties provide a service in the claims function where it is important that the associated activities are executed in a time efficient manner. The contract matchmaking streamlines the initiation of the appointment of such parties and the linking of two workflow systems allows greater monitoring facilities, highlighting areas where delays are occurring and corrective action is required. The application of the matchmaking element depends greatly on the complexity and specialty associated with the contract. However, specified criteria within the service consumer organization can be used to dictate the use of this element of the technology.

In an industry where the application of electronic tools is used to reduce inefficiencies and create competitive advantage, a CrossFlow application would achieve both of these objectives whilst also allowing the insurance company greater control over the process in terms of monitoring capabilities.

6.2 Logistics Scenario

KPN Research investigates for TNT Post Group how TNT can take advantage of CrossFlow technology to drive the competitive edge. Information-intensive and highly standardized basic logistic processes make TNT a good candidate for the introduction of workflow management systems to improve efficiency and cost-effectiveness. A major challenge is to increase the variety of the service portfolio without increase of costs. Also, integration with the systems of customers is an important challenge, e.g. a direct connection to their enterprise resource planning systems. Integration with workflow-enabled versions of these systems requires support for flexible, interoperable workflow management within TNT.

In the CrossFlow scenario, TNT acts as a service provider for KPN Telecom in the distribution of cellular phones. This requires a high-level and tight coupling of TNT's and KPN's workflow support, offering powerful monitoring and control functionality. KPN is in need of this control, because KPN's customers demand more control. Due to e-commerce developments, the end-customer wants to see how and when his order is handled. The end-customer could even interact with the running order processing, because he might now be contacted via the Internet. This requirement and challenge for TNT and KPN asks for a transparent integration of workflow systems and web stores. CrossFlow technology facilitates building a chain of transparently connected processes, across several involved organizations.

7. Related work

As CrossFlow is positioned at the crossroads of a number of recent developments, it can be placed into a broad context of related work. Below, we divide this context into four main areas: electronic commerce and trading, service specification and contracts, cross-organizational workflow management, and finally the combination of workflow management and electronic commerce.

Electronic commerce and trading. In recent years we have seen the development of a number of systems that enable the advertising and searching in an electronic form of goods and services - marketplace systems [Bako91]. Examples are ViMP [Hoff99b], Ariba (www.ariba.com), e-speak [HP99], or MIT's Kasbah prototype [Chav96]. Schmidt [Schm97] provides an elaborated categorization of electronic marketplace systems.
A number of related standardization efforts are underway, such as RosettaNet [Rose99], the eCo Framework [Smith99], and the set of standards of the Foundation for Intelligent Physical Agents (FIPA, www.fipa.org), which has been applied by [Shep99] for example. In the OMG ECDTF reference model [Sege00] currently under development, service, contract, and brokerage facilities are identified that address functionality related to that of CrossFlow.
The CrossFlow approach distinguishes itself from other developments in electronic commerce and trading by focusing on services of which the process specification is explicitly included in the trading paradigm and of which the execution can hence be controlled by the service buyer in a fine-grained fashion.

Service specification and contracts. In the past, a substantial amount of effort has been dedicated to electronic description and communication of structured information. Most notable are developments in the context of EDI [Cann93] and EDIFACT [UNE00]. This work generally concentrates on electronic exchange of product specifications, whereas our work focuses on trading, exchange and enactment of service specifications, including execution characteristics. The work in the XML community aims at structured information specification [Grah99]. We see this work as input to our work, as our contract specification language is based on XML.
Agreements and contracts are used differently in the area of transactions, workflow management and distributed systems, not always for the purpose of specifying mutual obligations between organizations, e.g. the ConTract approach for long-running transactions [Wäch91].
Contracts as a definition of a service relationship between organizations are used by a number of recent projects. The Coyote [Dan97] and the TOWEC approach [Verh98] provide mechanisms similar to ConTract, while explicitly taking into account that services as parts of transactions can be executed in different organizations.
Work is also being done on languages for describing contracts. An example is the 'courteous logic' approach, which is the basis of several concrete languages (e.g. Business Rules Markup Language - BRML) for expressing contracts between two parties [Gros99]. Another proposal is the documentary Petri net approach that can be executed by the InterProcs system [Lee98].

Cross-organizational workflow management. The developments in CrossFlow are related to standardization efforts in the development of advanced workflow support. Basic interoperability between workflow management systems has been addressed by the Workflow Management Coalition [WFM96], but realistic virtual enterprise settings require more than this. The Simple Workflow Access Protocol [Swen98] addresses the same issue, and is currently being integrated in the WfMC's framework as the XML binding of interface 4. These interfaces help cross WFMS vendor boundaries, but they do not address the issues of organizational boundaries. To accommodate this, electronic commerce aspects have to be added to cross-organizational workflow management.

Workflow management and electronic commerce. A number of other projects address the combination of workflow management and electronic commerce, as CrossFlow does. The

FlowJet project at Hewlett-Packard aims at coupling various types of workflow systems in E-business contexts [Shan99]. Dynamic resource brokering is within the scope of the project, but explicit contracts for detailed service specification are not considered. The WISE project is comparable to FlowJet as it uses cross-organizational workflow management technology for business-to-business E-commerce scenarios [Alon99]. MariFlow follows a similar approach to the WISE project but is specifically targeted at the marine industry [Cing99]. The project aims at providing process management capabilities in a similar manner to WISE, enhanced by an advanced marketplace for service contracts. The COSMOS project is developing an architecture that allows organizations to offer and search for services in a catalogue, a negotiation platform, and facilities for contract signing [Griff98]. Once the contract is signed, workflow specifications are derived from the contract or encapsulated in the contract constituents of the offering party and a new workflow instance is started. Electronic trading of workflows is also considered in other research efforts, like [Ditt99]. The Virtual Enterprise Coordinator (VEC) [Lud99b] follows the buyer-seller approach where organizations can make process factories externally accessible and allow other organizations to pick their process from a catalogue and integrate it as an atomic step in their process.

Our work distinguishes itself from these other approaches by the use of contracts as fine-grain specifications of the workflows (or rather services) to be traded.

8. Conclusions

In this report, we have described the CrossFlow project. This project aims at the support for cross-organizational workflow management in dynamic virtual organizations. The support for dynamic cooperation with a service consumer/provider paradigm introduces elements of electronic commerce into the field of workflow management. The CrossFlow approach is characterized by contract-based cooperation with advanced, fine-grained process monitoring and control.

The conceptual results of the CrossFlow project are reflected in a prototype based on a commercial workflow management platform. This prototype will be applied and assessed in two real-world scenarios. This will result in feedback for the improvement and extension of concepts and models.

A main direction of future work will be concerned with the contract framework. The use of fixed monolithic contract templates can be generalized to allow contracts to be built from units of smaller granularity. Reuse of contract templates can lead to a hierarchy of contract types coupled to a taxonomy market segments. Usage clauses in contracts have to be elaborated to specify in a flexible way how contracts covering multiple service cycles can be instantiated into single-cycle contracts. Also, the negotiation process may be extended to cover forms of matchmaking more complex than the current one-step offer-request paradigm.

Further research will be performed in the development of the cooperation support services that implement the fine-grained monitoring and control of the service provider by the service consumer.

Further information on the CrossFlow project can be obtained via the project's web site [CF00].

Acknowledgments

Justus Klingemann and Thomas Tesch of GMD-IPSI are acknowledged for their work on FCC respectively QoS aspects. Marjanca Koetsier and Jochem Vonk of the University of Twente are acknowledged for their work on contracts respectively LoC aspects. Wijnand Derks of KPN Research is acknowledged for his contribution to LoC aspects. Ceki Gülcü of IBM Zürich is acknowledged for his contributions to the technical aspects of the architecture. Sinead Browne and Michael Kellett of Church & General and Wim Nijenhuis, Matthijs Duitshof, and Zef Damen of KPN Research are acknowledged for their role in the definition of the scenarios. Jacques Saint-Blanquat of IBM La Gaude is acknowledged for his role as project administrator. Agustín González of Sema Group is acknowledged for his feedback on developments in the projcet. All other current and former members of the CrossFlow project are acknowledged for their contributions.

9. References

[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*, Sydney, Australia, 1999.

[Bako91] Y. Bakos, 'A Strategic Analysis of Electronic Marketplaces', *MIT Quarterly*, MIT, USA, 1991.

[By98] R. de By, W. Klas, J. Veijalainen (eds.); *Transaction Management Support for Cooperative Applications*; Kluwer Academic Publishers, 1998.

[Cann93] E.Cannon, *EDI Guide: a Step by Step Approach*, Van Nostrand Reinhold, New York, 1993.

[CF00] *CrossFlow Project Web Site*, CrossFlow Consortium, Europe, 2000, www.crossflow.org.

[Chav96] A. Chavez, P. Maes, 'Kasbah: An Agent Marketplace for Buying and Selling Goods', *Procs. 1st Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK, 1996.

[Cing99] I. Cingil, A. Dogac, N. Tatbul, S. Arpinar, 'An Adaptable Workflow System Architecture on the Internet for Electronic Commerce Applications', *Procs. Int. Symp. on Distributed Object Applications*, Edinburgh, 1999.

[Dan97] A. Dan, F. Parr, 'The Coyote Approach for Network Centric Service Applications: Conversational Service Transactions, a Monitor and an Application Style', *Procs. of the Workshop on High Performance Transaction Processing*, Asilomar, 1997.

[Ditt99] K. Dittrich, D. Tombros; 'Workflow Management for the Virtual Enterprise', *Procs. Int. Process Technology Workshop*, Villard de Lans, France, 1999.

[Farl98] J. Farley, *Java Distributed Computing*, O'Reilly, 1998.

[Gar87] H. Garcia-Molina, K. Salem; 'Sagas'; *Procs. 1987 ACM SIGMOD Int. Conf. on Management of Data*; USA, 1987.

[Geor95] D. Georgakopoulos, M. Hornick, A. Sheth, 'An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure', *Journal of Distributed and Parallel Databases*, Vol. 3, No. 2, 1995.

[Grah99] I. Graham, L. Quin, *XML Specification Guide*, Wiley Computer Publishers, New York, 1999.

[Gre99a] P. Grefen, B. Pernici, G. Sánchez (eds.); *Database Support for Workflow Management*; Kluwer Academic Publishers, 1999.

[Gre99b] P. Grefen, J. Vonk, E. Boertjes, P. Apers, 'Semantics and Architecture of Global Transaction Support in Workflow Environments', *Procs. 4th IFCIS Conf. on Cooperative Information Systems*, Edinburgh, Scotland, 1999.

[Griff98] F. Griffel, M. Boger, H. Weinreich, W. Lamersdorf, M. Merz, 'Electronic Contracting with COSMOS - How to Establish, Negotiate and Execute Electronic Contracts on the Internet', *Procs. 2nd Int. Enterprise Distributed Object Computing Workshop*, La Jolla, 1998.

[Gros99] B. Grosof, Y. Labrou, 'An Approach to using XML and a Rule-based Content Language with an Agent Communication Language', *Procs. IJCAI-99 Workshop on Agent Communication Languages*, 1999.

[HP99] *E-Services Broker Architecture. Beta Release 12/6/99*, white paper, Hewlett-Packard, 1999.

[Hoff99a] Y. Hoffner, 'Supporting Contract Match-Making', *Procs. IEEE 9th Int. Workshop on Research Issues in Data Engineering*, Sydney, Australia, 1999.

[Hoff99b] Y. Hoffner, C. Facciorusso, S. Field, A. Schade, 'Distribution Issues in the Design and Implementation of a Virtual Market Place', *Procs. 2nd IFIP WG 6.1 Int. Working Conf. on Distributed Applications and Interoperable Systems*, Helsinki, Finland, 1999.

[IBM00] *MQSeries Workflow Web Site*, IBM Corporation, USA, 2000; www.software.ibm.com/ts/mqseries/workflow/.

[Klin99] J. Klingemann, Jürgen Wäsch, Karl Aberer, 'Deriving Service Models in Cross-Organizational Workflows', *Procs. 9th Int. Worksh. on Research Issues in Data Engineering*, Sydney, Australia, 1999.

[Klin00] J. Klingemann, 'Controlled Flexibility in Workflow Management', *Procs. 12th Conf. on Advanced Information Systems Engineering*, Stockholm, Sweden, 2000.

[Koet99] M. Koetsier, P. Grefen, J. Vonk, *Contract for Cross-Organizational Workflow Management*, CTIT Technical Report 99-18, University of Twente, 1999.

[Law98] NJLRC – New Jersey Law Revision Commission, *Draft Final Report relating to Standard Form Contracts*, New Jersey Law Revision Commission, USA, 1998.

[Lee98] R. Lee, 'Distributed Electronic Trade Scenarios', *Int. Journ. of Electronic Commerce*, September, 1998.

[Lud99a] H. Ludwig, C. Bussler, M. Shan, P. Grefen, 'Cross-Organisational Workflow Management and Co-ordination – WACC '99 Workshop Report', *ACM SIGGROUP Bulletin*, Vol. 20, No. 1, 1999.

[Lud99b] H. Ludwig, K. Whittingham, 'Virtual Enterprise Co-ordinator - Agreement-Driven Gateways for Cross-Organisational Workflow Management', *Procs. of the Int. Joint Conf. on Work Activities Coordination and Collaboration*, San Francisco, 1999.

[OMG96] Object Management Group and X/Open, *RFP5 Submission: CORBA Trading Object Service*, Document orbos/96-05-6, 1996.

[Rose99] *RosettaNet Implementation Framework (RNIF) Specification, version 1.1*, RosettaNet, 1999.

[Schm97] B. Schmidt, 'Requirements for Electronic Markets Architecture', *Electronic Commerce*, Vol. 7, No. 1, 1997.

[Sege00] A. Segev, M. Bichler, 'Component-based Electronic Commerce', M. Shaw, R. Blanning, T. Strader, A. Whinston (eds.), *Handbook of Electronic Commerce*, Springer, 2000.

[Shan99] M. Shan, 'FlowJet: Internet-Based E-Service Process Management', *Procs. Int. Process Technology Workshop*, Villard de Lans, France, 1999.

[Shep99] J. Shepherdson, S. Thompson, B. Odgers, 'Cross Organisational Workflow Co-ordinated by Software Agents', *Procs. WACC Workshop on Cross-Organisational Workflow Management and Co-ordination*, San Francisco, 1999.

[Smith99] H. Smith, *CommerceNet's eCo Framework in a Nutshell!*, Ontology.org, 1999.

[Swen98] K. Swenson, *Simple Workflow Access Protocol (SWAP)*, Draft Proposal, 1998.

[UNE00] *The UN/EDIFACT Directories*, http://www.unece.org/trade/untdid/welcom1.htm.

[Verh98] E. Verharen, M. Papazoglou, 'Introduction to Contracting in Distributed Transactional Workflow' *Procs. 31st Annual Hawaii Int. Conf. on System Science*, 1998.

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

[Wäch91] H. Wächter, A. Reuter. 'The ConTract Model', A. Elmagarmid (ed.), *Database Transaction Models for Advanced Applications*, Morgan Kaufmann Publishers, 1991.

[WFM96] *Workflow Standard - Interoperability Abstract Specification*, Document Number WfMC-C-1012, Workflow Management Coalition,1996.