RC22378 (W0203-061) March 19, 2002
Computer Science

IBM Research Report

A Framework for E-Services: A Three-Level Approach towards Process and Data Management

Paul Grefen, Samuil Angelov
Computer Science Dept.
University of Twente
The Netherlands

Heiko Ludwig
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

A Framework for E-Services: A Three-Level Approach towards Process and Data Management

Paul Grefen[♣] grefen@cs.utwente.nl
Heiko Ludwig[♠] hludwig@us.ibm.com
Samuil Angelov[♣] sangelov@cs.utwente.nl

[♣]Computer Science Dept.
University of Twente
The Netherlands

[♠]IBM TJ Watson
Research Center
USA

Abstract

Service outsourcing is the business paradigm, in which an organization has part of its business process performed by a service provider. In dynamic markets, service providers can be selected on the fly during process enactment. The cooperation between the parties is specified in a dynamically made electronic contract. This contract includes a process specification that is tailored towards service brokering and cross-organizational process enactment and hence has to conform to market and specification standards. Process enactment, however, relies on intra-organizational process specifications that have to comply with the infrastructure available in an organization for process and data management. In this report, we present a three-level process and data specification framework for dynamic contract-based service outsourcing. This framework relates the two process specification levels through a third, conceptual level. This approach is inspired by the well-known ANSI-SPARC model for data management. We discuss an abstract architecture for dynamic service outsourcing based on the three-level framework. We show how the framework and architecture can be placed in the context of existing infrastructures for cross-organizational process support. As service outsourcing is used more and more for core business processes requiring reliable execution, we pay special attention to transaction management.

TABLE OF CONTENTS

1 Introduction 1
- 1.1 Dynamic service outsourcing 1
- 1.2 The e-service concept 2
- 1.3 An example scenario 2
- 1.4 The need for a framework 3
- 1.5 Aim and structure of this report 3

2 A three-level process framework 5
- 2.1 The need for three specification levels 5
- 2.2 The three-level framework 6
- 2.3 Private and shared specifications 6
- 2.4 The ANSI-SPARC connection 7

3 The conceptual level 9
- 3.1 The nature of the conceptual process specification level 9
- 3.2 Process specification techniques 9
- 3.3 An example conceptual process structure specification 9
- 3.4 Process-relevant data in conceptual process specifications 11
- 3.5 Transactional aspects 12
- 3.6 Designing conceptual process specifications 13
- 3.7 Analyzing and verifying conceptual process specifications 15

4 The external level 17
- 4.1 The nature of external process specifications 17
- 4.2 Process specification techniques 18
- 4.3 Mapping the conceptual level to the external level 18
- 4.4 Example external process specification 18
- 4.5 Process-relevant data in external process specifications 20
- 4.6 Transactional aspects 21
- 4.7 External process specifications in electronic contracts 21

5 The internal level 23
- 5.1 The nature of internal process specifications 23
- 5.2 Process specification techniques 23
- 5.3 Mapping the conceptual level to the internal level 23
- 5.4 Example internal process specification 24
- 5.5 Process-relevant data in internal process specifications 27
- 5.6 Transactional aspects 28

6 Reference architecture 29
- 6.1 Global architecture 29
- 6.2 Process design 30
- 6.3 Service contracting 31
- 6.4 Service enactment 32

7 Specification management 35
- 7.1 Specification storage 35
- 7.2 Specification data model 36

8 Related work ... 39
8.1 Standards ... 39
8.2 Proprietary systems for cross-organizational process integration ... 42
8.3 Research projects ... 42
9 Conclusions and outlook ... 45
Bibliography ... 47

1 INTRODUCTION

In this section, we provide an introduction to the framework for e-services that we elaborate in this report. First, we explain our view on dynamic service outsourcing, the business paradigm that underlies our notion of e-services. Next, we discuss our notion of the concept of e-service in more detail and place it in contrast with the concept of monolithic web service (also called e-service by other authors). Then, we illustrate the paradigm with an example scenario from the logistics domain that we use throughout this report. We continue with showing why the framework we propose is indeed necessary. We end this section by explaining the aim and structure of this report.

1.1 Dynamic service outsourcing

Nowadays, many organizations focus on their core business process and buy services from partners in the market to perform the additional parts of the process required to reach their business goals. We call this the service outsourcing paradigm. In this paradigm, the outsourcing organization is referred to as service consumer, the service implementing organization as service provider. The details of service outsourcing are specified in a contract between both parties.
The combination of service consumer and service provider can be seen as a virtual enterprise that presents itself to a third party (for example a customer) as a single entity. Traditionally, these virtual enterprises have a more or less stable character over time. In dynamic e-commerce settings, however, players in a market and competitive situations change that fast, that a more dynamic approach is required to service outsourcing to create or retain a competitive position. This means that in service outsourcing, service consumers dynamically determine which service providers to use in the enactment of their business processes.
In Figure 1, dynamic service outsourcing is illustrated. On the left, we see a service consumer that wants to outsource a part of its business process (steps D and E). On the right, we see a number of service providers offering compatible services (D+ and E+). Process enactment at the consumer side is dynamically linked to process enactment at the provider side.

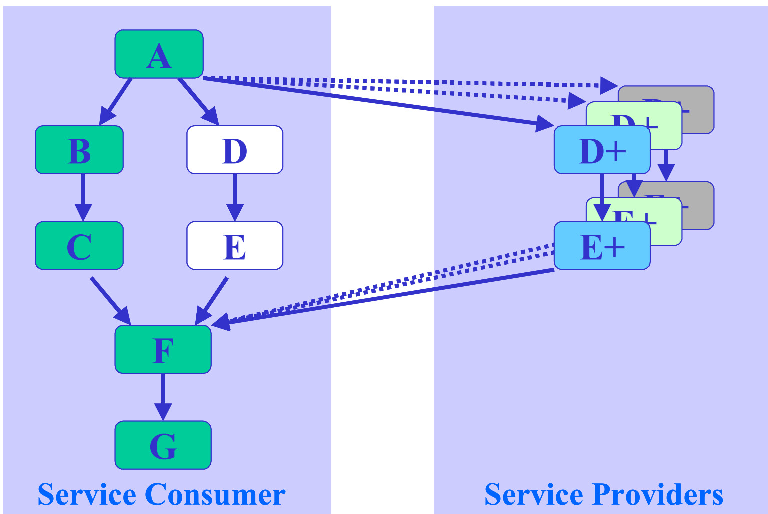

Figure 1: dynamic service outsourcing

1.2 The e-service concept

In the context of the above, we define an electronic service – or e-service for short – as follows:

> *An e-service is a non-trivial part of a business process that is executed by a service provider on behalf of a service consumer through electronic means.*

By a non-trivial part of a business process, we mean a part that has an externally visible internal process structure. Consequently, to support the execution of e-services, inter-organizational process and data management aspects have to be addressed. This is the focus of this report: we propose a framework for specification and execution of e-services.

Other service types whose internal process structure is not shared between provider and consumer are not addressed by the work in this report. Monolithic web services (sometimes also called e-services, see e.g. [Mec01]) are an example of this class. Cross-organizational execution of a web service relies on a simple call-and-return interaction paradigm – the internal process structure of the web service is a black box to the consumer party.

We assume that the outsourcing of e-services can be performed just when the need arises if both outsourcing and actual enactment of the service is implemented by means of an automated process and data management infrastructure. We call this dynamic outsourcing of e-services.

In the sequel of this report, we use the term service as shorthand for e-service as defined above.

1.3 An example scenario

Service outsourcing can be found in many market segments. An example is an insurance company that focuses on the core insurance process and outsources secondary activities like customer call handling and insurance claim loss assessment to other companies. In this case, customer call handling may be outsourced statically to a fixed business partner and claim loss assessment dynamically to one of a set of possible business partners.

Other examples can be found in the logistics domain, where companies outsource their logistic subprocesses to specialist organizations in this field [Dui00]. We elaborate a scenario from this domain as a running example throughout this report. The scenario is based on a telecom operator and logistics providers, as depicted in Figure 2. We see on the left hand the business process of selling mobile telephones (GSM phones) to customers of the telecom operator. At the operator, telephones are sold, their sale is handled (among another things, the telephone is connected to the network), and the sale is finalized – all activities that are core activities for the telecom operator. The physical logistics part of the process, i.e., actually delivering the telephone to the customer, is not a core activity of a typical telecom operator. Therefore, this activity is dynamically outsourced to a specialized logistics provider. The logistics provider is selected in the course of the sales process, based on shipping requirements of the telecom operator and current offers of the logistics providers. We elaborate the processes in more detail in Section 3.

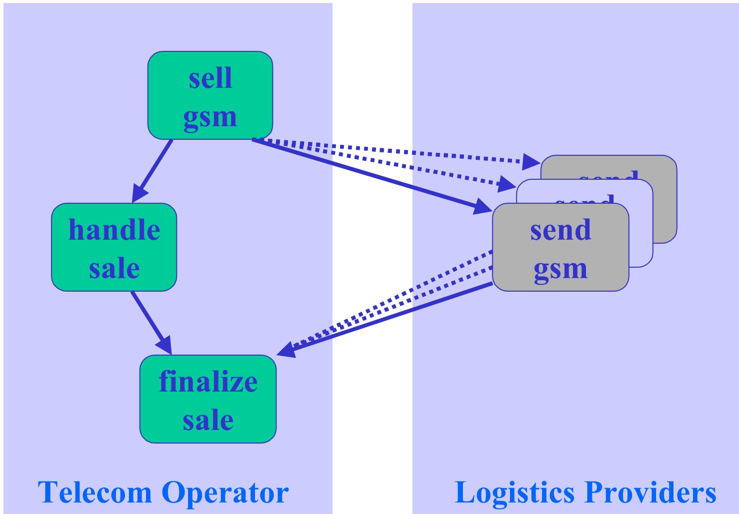

Figure 2: example scenario

1.4 The need for a framework

Dynamic service outsourcing as explained above implies a number of steps: services have to be identified and defined, compatible business partners have to be found in an efficient way, contracts have to be made dynamically, process enactment and data management infrastructures have to be set up and coupled, and the process has to be actually enacted. Requirements to speed and costs dictate that all of this takes place electronically, preferably in a fully automated way.
To enable this, clear process specifications are necessary that describe process structures and the process-relevant data structures for the above-mentioned steps in service outsourcing. Given the fact that these steps are quite diverse in nature, a single process specification will not be adequate for all these activities. Hence, we propose a conceptual framework of multi-level process specification for dynamic service outsourcing. This framework is linked to the concept of electronic contract [An01b, Ang02, Gre02], which serves to formally define the business relationship between organizations entering into a dynamic virtual enterprise.
The multi-level framework we propose is inspired by the ANSI-SPARC model [Tsi78] that is well known in the database community for describing three levels of data management. The general approach to service outsourcing used in this paper is inspired by the approach taken in the CrossFlow project [Gre00, Hof01].

1.5 Aim and structure of this report

Main aim of this report is to paint the overall picture of our approach to multi-level process support for service outsourcing. This means that not all details are covered – part of these are subject of further research. We do pay special attention to transaction management aspects in this report. This attention is caused by the observation that service outsourcing is used more and more for core business processes that require a high level of reliability in their execution.
In Section 2, we first present our three-level process model, explaining what the global purpose of each of the levels is. In Sections 3 to 5, we discuss each of the three levels and the mappings between them in more detail. The levels are illustrated by means of the running example introduced in this section. We place the process model in the context of process support infrastruc-

tures by discussing architecture aspects in Section 6. We base this discussion on a reference architecture that provides a blueprint for concrete architectures. Data management aspects from the architecture are elaborated in Section 7. In Section 8, we discuss related work, paying attention to both research projects related to our approach and standardization efforts in the domain of our work. We end the report with conclusions.

2 A THREE-LEVEL PROCESS FRAMEWORK

In this section, we introduce our three-level process framework. We start with explaining why three levels of process specification are indeed required. Then we describe and illustrate the framework – the central process management paradigm in this report. Next, we show how shared and private process specifications exist at the three levels and how shared specifications come into existence. We end this section with showing how our framework relates to the well-known ANSI-SPARC framework from the field of database management.

2.1 The need for three specification levels

In service outsourcing, we require process specifications that specify all necessary details of process and data structures with respect to these services and their enactment. These specifications exist on two levels from an operational point of view.

Firstly, we require process specifications that can be shared between multiple organizations (service consumer and service providers). These specifications should allow for service brokering between partners when setting up a virtual enterprise on the one hand, and service enactment in a running virtual enterprise on the other hand. Brokering is used to relate service consumers and providers, either through simple matching of specification characteristics or through more advanced matchmaking. In service enactment, services are actually executed in an inter-organizational setting. Interoperability is the main characteristic of this specification level. The nature of this level is often imposed by market standards. If an organization operates in multiple markets, it may be confronted with multiple standards.

Secondly, we require process specifications at the implementation level of the process enactment systems employed by each of the organizations cooperating in a virtual enterprise. Executability is the main characteristic of this specification level. The nature of this level is often imposed by current process and data management systems – be it of an up-to-date or legacy nature. Again, an organization having a heterogeneous infrastructure may be confronted with multiple specification types here.

None of the above two levels is focused on conceptual process specification, however, that is independent from both practical collaboration and implementation aspects. A conceptual specification is required, though, for three main reasons:

- A conceptual specification is necessary to guard the quality of processes through proper design, analysis, and validation of process structures. Certainly, in the field of e-business where automated processes are the backbone of an organization, quality of these processes is of the highest importance.
- A conceptual specification is used as a ‘semantic bridge’ between the standards used at the interoperability level on the one hand and the implementation level on the other hand. This bridge provides a separation of concerns with respect to specific details of semantics at the two levels.
- A conceptual specification is also required as a unifying level to bridge multiple specification types at both interoperability and implementation levels. As such, it is a basis for portability of process specifications across multiple markets with diverse standards on the one hand and multiple process and data management systems with diverse technologies on the other hand.

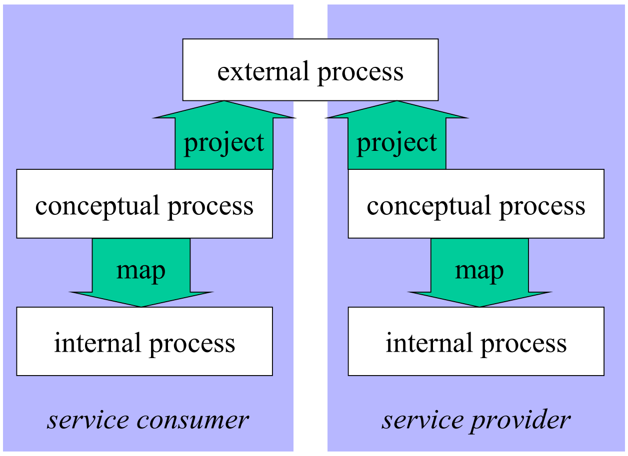

Figure 3: three-level framework

2.2 The three-level framework

For the reasons explained above, we introduce a third, conceptual process specification level between the external and internal levels. Consequently, we arrive at the following three-level process specification framework:

External level: The external level is geared towards communicating a process specification between different organizations. It can be considered a projection of the conceptual level, where projection uses hiding and translation operations.

Conceptual level: The conceptual level is the centerpiece of process specification. It is independent from external use and internal implementation. It is used for conceptual reasoning about the process, e.g. for design and analysis purposes. The conceptual level is a combination of abstraction and aggregation of the internal level.

Internal level: The internal level is geared towards enactment of processes in the context of a specific organization, e.g., by means of workflow management systems. The internal level is a mapping of the conceptual level, where mapping is a combination of translation (specialization for a specific platform) and refinement.

The relation between the three process levels in service outsourcing between consumer and provider organizations is illustrated in Figure 3.
In Sections 3 to 5, we discuss the levels of the framework in more detail. We start with the conceptual level, as this level is the pivotal point in our framework. Next, we move 'up' to the external level, paying attention to the nature of this level and the projection from the conceptual level. After that, we move our attention 'down' to the internal level and to its mapping from the conceptual level. At all three levels, we pay attention to both process and data management aspects. As explained in the introduction, transaction management receives special attention.

2.3 Private and shared specifications

In Figure 3, we see that two organizations each have local conceptual and internal process specifications, but that they share a common external specification. In terms of our running example, this means the following. Both the telecom operator and the logistics provider have a private conceptual process specification that describes their logical process. This includes the process interface to the other party, so in the specification of the telecom operator the way its logistics are outsourced to an external party and in the specification of the logistics provider the way services requested by external parties are handled. Each organization has a private mapping of its specification to the internal level. Two collaborating parties share a common external specification describing the way they work together.

The private specifications can be constructed by each organization individually. The common external specification, however, needs consensus among parties. This consensus can be based on several mechanisms:

- It can be based on a process standard in a specific market that is imported by both organizations. In this case, the standard is often constructed by a standardization body (see also Section 1).
- It can be based on a model exported by one organization and imported by the other. In this case, it is often exported by the service provider. It can be exchanged either directly between provider and consumer, or through a service broker. In the latter case, exporting actually means advertising.
- It can be constructed in a process of bilateral negotiation between service consumer and provider. Clearly, the latter alternative implies a sophisticated mechanism for inter-organizational process construction.

2.4 The ANSI-SPARC connection

Note that the three above levels coincide more or less with the well-known ANSI-SPARC three-level model in the database community [Tsi78, Elm94]. The ANSI-SPARC model distinguishes between external, conceptual and internal levels for data management. Aims of the conceptual and internal level are similar to that in our approach – except of course for the data versus process management perspective. Main difference is the use of the external level. In the ANSI-SPARC model, the external level contains multiple views on a database that cater for different user groups in an organization. In our model, the external level contains multiple 'views' on a process to cater for different organizations in an electronic market.

3 THE CONCEPTUAL LEVEL

The conceptual process specification level is the center point in our framework for cross-organizational process support. It is used for the design, analysis and verification of cross-organizational processes.

3.1 The nature of the conceptual process specification level

The process exhibited by an organization to the outside world is usually less detailed than the implementation of the same process actually enacted by the organization. Consequently, several process aggregation levels exist. The mapping between these aggregation levels is dealt with by a process refinement hierarchy.

As the conceptual specification level is the design and analysis level of our process specification framework, we place the process refinement levels that have a conceptual meaning within this level. Refinement that is related to mapping to process enactment infrastructures is dealt with in the mapping between conceptual and internal process specification – see Section 5.3.

To properly support design and analysis tasks at the conceptual level, an adequate process specification technique has to be selected. As we deal with discrete business processes, we require a discrete process modeling technique. As unambiguous reasoning about process specifications should be possible at this level, we require a specification technique with a formal background. Clearly, the chosen technique should allow to be mapped to both the external and the internal process specification levels. We discuss process specification techniques in Section 3.2. The technique should allow the addition of transaction characteristics of processes. We pay attention to transaction specification at the external level in Section 3.5.

Apart from the process structure specification, we require a conceptual specification of data elements that are relevant for process enactment. Most important in this context is the data that is used in the control of the process execution, e.g., in the evaluation of pre- and postconditions of process steps or in routing decisions for processes with alternative process flows. We call this data process-relevant data. In other contexts, it is referred to as process variables, in the workflow management world as case data. We discuss the data aspect in Section 3.4.

3.2 Process specification techniques

In this report, we do not advocate the use of specific process specification techniques or languages. Below, we briefly discuss some alternatives – this mainly for illustrative purposes.

Variants on Petri Nets are widely accepted in the business process reengineering and workflow management domains (see for example [Aal98, Ada98]), both from a research and from an industrial perspective. Their discrete, token-based, semantics makes them applicable for modeling business processes. Also, there is a substantial amount of work available on general design and analysis of Petri Nets (e.g. [Pet81, Rei92, Deh01]). We illustrate the use of basic Petri Nets in Section 3.3.

As UML is widely used in information system specification, UML activity diagrams are also an option for specification of automated business processes [Es01a].

3.3 An example conceptual process structure specification

Our example is based on a real-world logistics scenario in the telecom industry developed in the CrossFlow project [Dui00, Gre00]. In this scenario, a telecom operator sells mobile phones

(GSMs) and network subscriptions for these telephones. The subprocess of delivering the telephones to clients is dynamically outsourced to a logistics company. The example process at the telecom operator is specified in an abstract technique at the conceptual level in Figure 4.

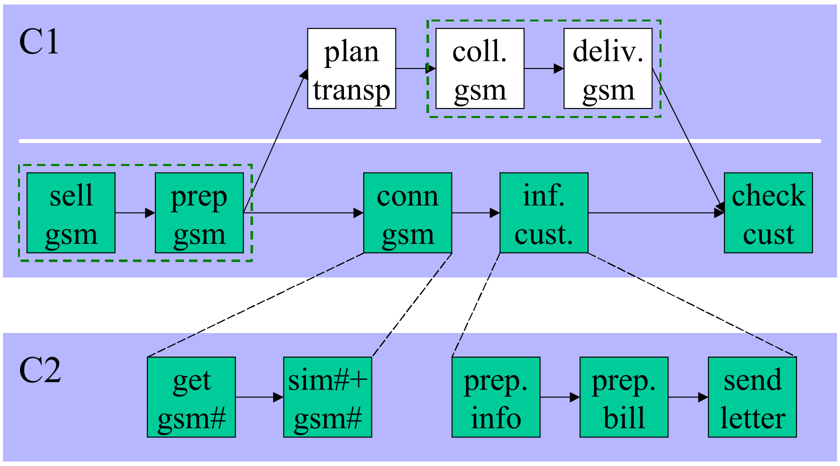

Figure 4: example conceptual process in abstract technique

In the upper part of the figure, we see the level 1 conceptual process model. Below the white line, we have the internally handled part of the process, above the white line the outsourced part of the process. The telecom operator sells a GSM, using telephone or web communication with its client. It then prepares the GSM for transport. After that, two parallel process branches are executed. In-house, the GSM is connected to the network and the customer is informed by mail about the connection, the personal identification code and the subscription. The process of physically delivering the GSM to the customer is outsourced to a logistics operator, who performs the delivery in three sequential steps. After completion of the two parallel branches, the telecom operator checks up on the customer by telephone.

Apart from the control flow of the process, certain execution characteristics are required, e.g. with respect to transactional properties or timing of processes. These characteristics can be seen as process quality attributes. Two examples are drawn as dotted boxes in Figure 4 (following the idea of spheres from [Der01]). The box around 'sell gsm' and 'prepare gsm' indicates a transactional property: these two activities should be executed atomically – we discuss transactional aspects in Section 3.5. The box around 'collect gsm' and 'deliver gsm' indicates a timing property. These two activities should be completed within a certain timeframe (i.e., after the phone has been picked up at the telecom company, it should be delivered at the client in say three days). Note that in practice, more execution characteristics will be specified.

In the lower part of Figure 4, we see (part of) the refinement of the level 1 into the level 2 process model. Activity 'connect gsm' has been refined into two activities for retrieving a GSM phone number respectively coupling this in the administration to the SIM number of the phone. Activity 'inform customer' has been refined into three activities.

In Figure 5, the process from Figure 4 is depicted as a Petri Net. The mapping from the abstract specification is straightforward – the only difference being the fact that the parallelism between the branches is now modeled explicitly (two tokens leaving transition 'prepare gsm').

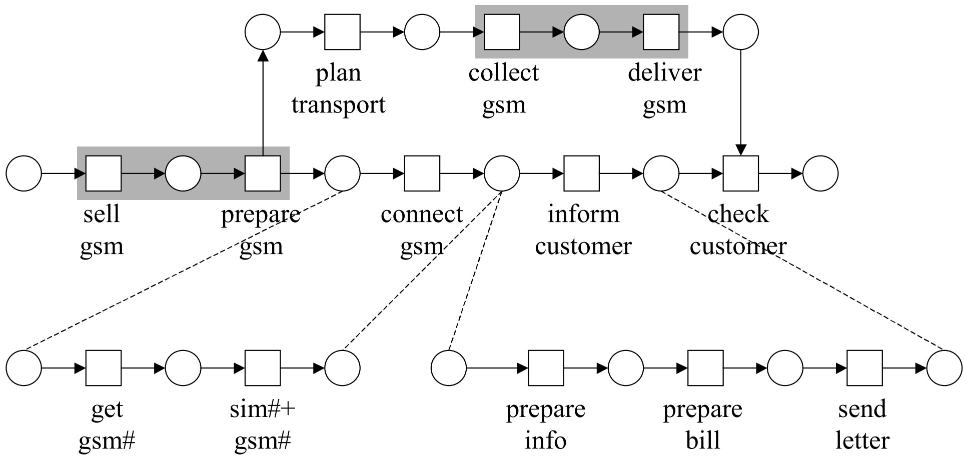

Figure 5: example conceptual process in Petri Net

3.4 Process-relevant data in conceptual process specifications

In the course of their execution, processes read and modify data. Process-relevant data in general serves two purposes:

- It is used by the individual activities associated with the steps of a process.
- The data is used by the process management system as parameters in conditions for branching, starting a step and completing a step.

In many cases, one data item is used for both activities and process management.

As expressed in the term "process-relevant data", the primary interest in data items in the context of a conceptual-level process model lies in its usage by the elements of the process flow. It is not in the interdependencies of data as they are represented in conceptual data models, e.g. using an entity-relationship diagram. According to the two purposes of data items in conceptual process models, data items are used by activities and transition conditions of the process flow model.

Figure 6 illustrates the usage relationship between a process flow model and its associated process-relevant data. All data items are part of the process-relevant data. In the first case, the "sell gsm" step accesses the data items "Customer Address", "Customer Payment Inf." and "Device Data" - data that is being collected in the course of a sale of a gsm service. In the second case, the transition condition to the "connect gsm" step uses the phone number. The transition condition expresses that the step can only be carried out when the phone number assigned to this service has been set by a prior activity.

Each step and each transition condition has a usage relationship to the process-relevant data, which may be empty in some cases. Data items of the process-relevant data can be used by multiple steps and transition conditions, even in parallel branches of the process flow. It is up to the execution environment of the process management system to ensure consistency in the case of concurrent access. The granularity of the data item depends, for this purpose, on the granularity of the usage relationship of process flow elements.

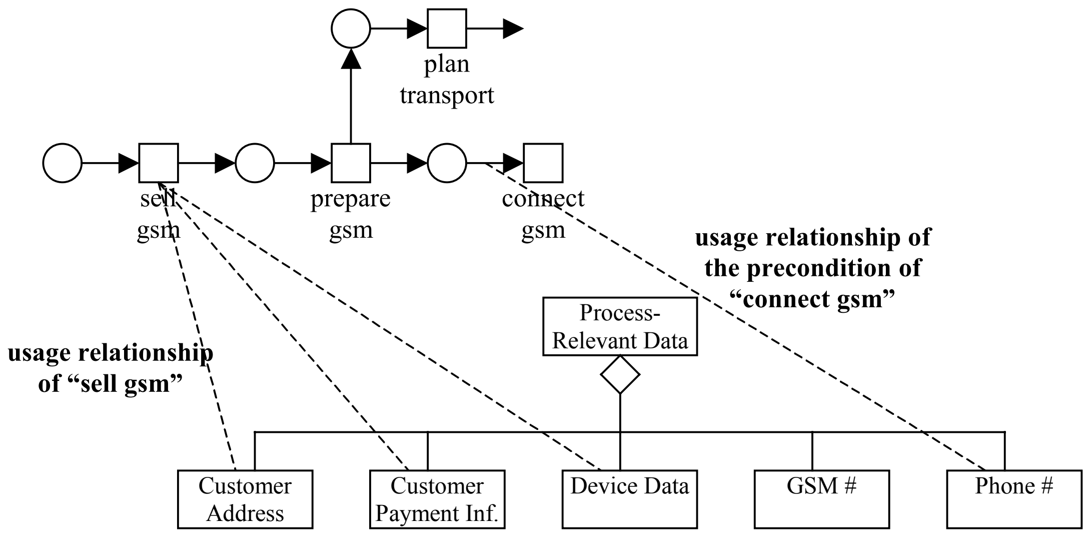

Figure 6: data usage in conceptual process

3.5 Transactional aspects

Depending on process types to be supported in service outsourcing, various types of transactional characteristics of processes have to be specified at the conceptual level: atomicity requirements of parts of a process, isolation requirements of parts of a process, and integrity constraints with respect to a process[1]. Atomicity requirements can be of two kinds: strict atomicity and loose atomicity requirements.

To specify strict atomicity, parts of a process are indicated that are to be executed in an atomic (all-or-nothing) fashion. A first approach is to strictly partition a business process into atomic subprocesses, which may be referred to as business transactions [Gre97]. This means that every process step is part of a business transaction. A second approach is to annotate arbitrary (non-overlapping) subprocesses as atomic, which may be called atomicity spheres [Der01]. This means that not all process steps are part of atomic subprocesses. Figure 4 contains an example of an atomicity sphere around steps 'sell gsm' and 'prep gsm'. If we consider all other steps atomic by default, the process is partitioned into business transactions (most of which are single-step).

Loose atomicity is usually specified by means of compensation patters. These patterns specify how the execution of parts of a business process can be undone by executing inverse subprocesses. Technology-independent compensation mechanisms for entire processes are formally specified in [Gre01]. In [Ley95], an approach based on joint spheres of compensation is described, which specifies subprocesses that have to be compensated upon rollback.

Isolation requirements can be modeled by specifying parts of a process that do not share intermediate results with other parts of the same process (or other processes). Specification can be based on a notion of isolation spheres, i.e., subprocesses that are isolated. If so required, isolation requirements can be parameterized with respect to the data objects isolation is defined on. Note that specification of isolation is heavily related to process-relevant data aspects as dis-

[1] Note that these three categories correspond with the first three categories of the well-known ACID transaction properties from the database world [Elm94]: atomicity, correctness and isolation. The fourth category (durability) is not of interest here.

cussed in Section 3.4. Isolation requirements are also heavily influenced by the underlying data management paradigm: shared data spaces or explicit data flow.
Integrity constraints usually pertain to process-relevant data, as discussed in Section 3.4. The constraints can be specified as predicates over these data. In this report, we do not go into the details of integrity control, as they are not specific to the aspects we discuss[2].

3.6 Designing conceptual process specifications

Design of processes takes place at the conceptual level, preferably in a top-down fashion. In a service outsourcing scenario, this means that at the highest aggregation level of the conceptual level, the conceptual interaction between service consumer and service provider is determined. Here, a number of aspects require due attention: scope of the conceptual specification, level of detail of the conceptual specification, and control flow interface at the conceptual level.

3.6.1 Scope of the conceptual specification

At the conceptual level, the service consumer specifies its conceptual cross-organizational business process in conceptual terms. This process contains activities that are enacted in-house, which are refined in the conceptual level if necessary. It also contains activities to outsource, which are not further refined, but enacted by a service provider.
The service provider also specifies the service it offers in conceptual terms at this level. The service specification may be extended with additional activities that are internal to the service provider, for example administrative activities. All activities can be refined within the conceptual process specification level.

3.6.2 Level of detail of the conceptual specification

Refinement of processes at the conceptual level should stop at the level where independence from execution infrastructures cannot be guaranteed. This means, for example, that process properties of specific workflow management systems should not be reflected at the conceptual level.
Summarizing the above, there is a highest process aggregation level at the conceptual specification level – determined by collaboration characteristics – and a lowest aggregation level – determined by platform independence. We call these levels the conceptual upper and lower bound. This is illustrated in Figure 7, where we see three conceptual refinement levels (C1 to C3) between external (E) and internal (I1 and I2 – explained later) levels.

[2] See [Gre93] for a general overview of integrity management aspects in database systems.

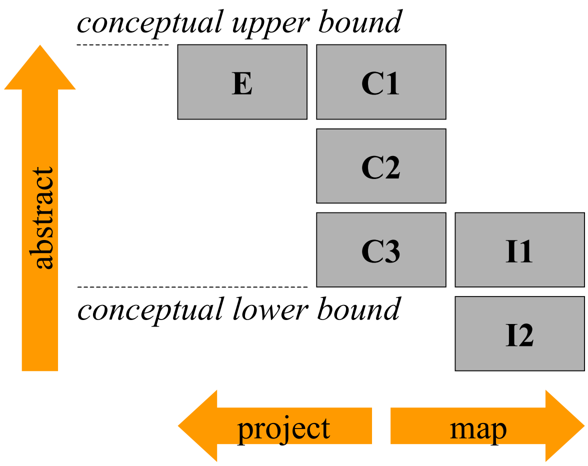

Figure 7: conceptual upper and lower bound

3.6.3 Control flow interface at the conceptual level

A special point of attention at the conceptual level is the control flow interface between in-house and outsourced part of a process, as this heavily influences the process execution autonomy of involved parties. We illustrate this below.
In Figure 8, we show a modified version of the top level of the process from Figure 4. We have changed two control flow elements: a telephone is now connected after its transport has been planned and the customer is informed before the telephone is delivered. In the original process from Figure 4, the in-house part of the process starts the outsourced service and synchronizes with it the very end (at task 'check cust'). As such, the control flow of the service provider is a black box to the service consumer. In the modified process, there is both incoming and outgoing synchronization between service consumer and service provider (as shown by the control flow arcs crossing the white line in Figure 8). This limits the process execution autonomy of both parties.

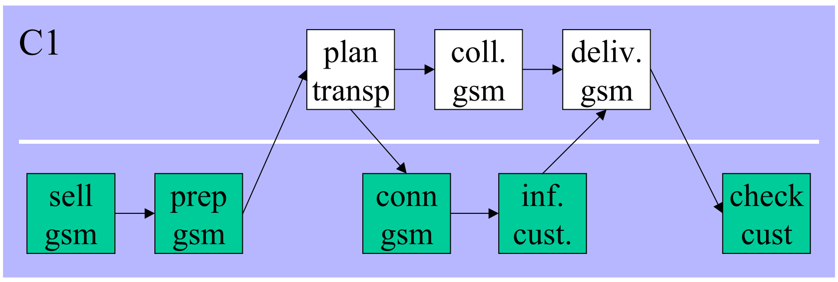

Figure 8: process with modified control flow interface

Given our asymmetric model of service outsourcing with service consumer and service provider, we distinguish between four control flow interface classes:

Black box: with a black box interface, the service consumer observes the service process at the provider as a black box. This means that the consumer has no information about the way the service is executed. In our three-level specification approach, a black box service is modeled by a single outsourced activity at the conceptual (and hence external) level. This interface class is not interesting in the context of this report – it is actually excluded by our definition of service as given in Section 1.

Glass box: with a glass box interface, the service consumer can observe the internal state of the outsourced process, but does not synchronize with it through control flow. This is the case in the process as depicted in Figure 4: there are no control flow relations between in-house and outsourced activities except for start and end of the outsourced service.

Half-open box: with a half-open box interface, the state of the outsourced process is synchronized with that of the consumer's in-house process through one or more control flow relations, i.e., arrows going from consumer to provider, but not in the opposite direction. This means that the progress of execution at the provider is influenced by that of the consumer, and that the provider's autonomy is consequently reduced.

Open box: in the open box interface class, there can be arbitrary control flow relations between consumer and provider. The execution progress of both parties depends on one another and the execution autonomy of both parties is reduced. The process in Figure 8 is an example of this class: arrows go in both directions between in-house and outsourced activities.

Clearly, the chosen interface class is heavily dependent on the collaboration paradigm that is chosen in service outsourcing.

3.7 Analyzing and verifying conceptual process specifications

As discussed in Section 2, one of the purposes of the conceptual process specification level is to allow analysis and verification of process specifications. Analysis of a specification is used to assess qualitative or quantitative characteristics of the specified process. Verification of a specification is used to check whether a specification possesses a specific characteristic.

Qualitative analysis may, for example, focus on transactional characteristics of business processes (see [Der01] for an example). In this type of analysis, the specification can be analyzed with respect to its atomicity or isolation behavior.

Analysis can also have a quantitative character. Quantitative properties of a specification can be established statically or dynamically. Dynamical analysis can be performed by using simulations of process runs.

Verification techniques may be used to prove certain characteristics of processes, e.g., reachability or termination characteristics. Analysis and verification lead to the assessment of quality characteristics of services that may be required as attributes at the external process specification level.

4 THE EXTERNAL LEVEL

Interoperability of processes is the main focus of the external level. Below, we discuss the nature of external process specifications and their mapping from the conceptual level. We illustrate this level with an example. We pay attention to data and transaction management aspects. Finally, we discuss the role of external process specifications in electronic contracts.

4.1 The nature of external process specifications

Process specifications at the external level are used for two purposes:

Service brokering: The external process specification is used in brokering between service consumers and service providers, i.e., in setting up a virtual enterprise. The external process required by a consumer should match with an external process offered by a provider in order to link their processes.

Cross-organizational process control: The external process specification is used for controlling the cross-organizational service enactment, i.e., it is the basis for cross-organizational process monitoring and control.

Note that process specifications are generally used differently in the above two aspects. Service brokering will usually be based on some level of predefined services, i.e., brokering will usually take place at a high abstraction level of the external specification[3] or with respect to specific attributes of the specification. In process control, however, all details of an external service specification have to be taken into account. In this report, the focus is on process control aspects.

Translation of the conceptual process specification is necessary to conform to common market standards[4] or bilateral standards – this to allow interoperability between partners in brokering and process control. Where the conceptual process specification uses a formal process model – as explained in the previous sections – the external model is based on a process specification standard that is common within a certain market. We discuss specification techniques at the external level in Section 4.2.

A process specification at the external level can be a high-level abstraction of a complex process that on the conceptual level is considerably refined into separate subprocesses. Still the process specifications at the external level should not be too general as they may turn the process of service outsourcing in a black box process, thereby not allowing the fine-grained cooperation that is required in dynamic virtual enterprises. Clearly, this is closely related to the discussion of the conceptual upper bound in the previous section. Also, some constraints have to be observed when mapping the conceptual to the external level – we discuss this in Section 4.3.

The external level includes service execution characteristics that define how the service is executed. These characteristics can be of different natures. An important class of characteristics is the specification of the transactional behavior of service execution, determining, for example, how failures and concurrent access to shared resources are handled. A second class is the specification of monitoring and control points in the service process and available control primitives – this to specify how a service consumer can observe and influence the enactment of an out-

[3] The highest level of abstraction is – of course – only the service identification. In not too advanced situations, this may often be the most practical approach.

[4] We treat market standards in more detail in our discussion of related work in Section 8.

sourced service. Quality of service attributes, e.g., execution times or success probability, form another important class.

4.2 Process specification techniques

By their very nature, external process specifications will be shared among organizations. This not only requires a common process meta-model but also a description format that can be used to exchange specifications among organizations. In the context of business-to-business integration, we find, mostly, XML-based languages as process specification techniques at the external level.

In the industrial domain, business-to-business standard frameworks such as ebXML [eb01a], Web services [W301], and RosettaNet [RN01] receive increasing acceptance. Each of those frameworks defines an approach to define processes in addition to messaging and other, lower-level interoperability layers. Based on ebXML Business Process Specification Schema and RosettaNet, specific industry groups define standard external-level processes. We discuss these approaches in more detail in Section 8.

In the research domain, many XML-based languages are currently under development for cross-organizational service specification. An example is the contract language developed in the CrossFlow project. We will spend some more words on this in Section 8.

In this report, we use ebXML as an example approach for the external level. Consequently, we use UML activity diagrams as process specification technique (see Section 8.1.1 for an explanation of the ebXML framework and the role of UML in ebXML) – we illustrate this in the sequel. The semantics of activity diagrams in workflow specification is discussed in [Es01a].

4.3 Mapping the conceptual level to the external level

The external specification is a projection of the conceptual process specification, where projection is a combination of hiding and translating – both of process and data structures.

Parts of the conceptual process can be hidden in the external process because they are not relevant for a cooperation partner. A service consumer might hide everything of its conceptual process but the subprocess to be outsourced. A service provider will usually show most of its conceptual process, but might hide some administrative tasks at the end of a service specification. Process steps may also be renamed from conceptual to external level, as internal names may not coincide with standard names in a market.

A similar approach is used for mapping process-relevant data from conceptual to external level. Part of the data may be hidden in the mapping and part of the data may be renamed or transformed. In data management terms, the mapping coincides with a complex projection operation (in some cases combined with a selection operation).

Translation is necessary to transform primitives from the conceptual process specification technique into the process specification standard used at the external level. Also, execution quality characteristics determined at the conceptual level have to be translated into service attributes at the external level. As discussed before, transactional properties of services are important aspects in this context – we treat these in Section 4.6.

4.4 Example external process specification

In Figure 9, we show how the top-level conceptual process (bottom part of the figure) is projected onto the external level (on part of the figure), using an abstract specification technique. Only the activities to be outsourced are represented at the external level in the external representation language. We see that steps in the process are renamed. Where the conceptual specification uses telecom-specific names, the external specification uses names generally used in the logistics industry: ‘collect gsm’ is for example renamed into ‘collect parcel’.

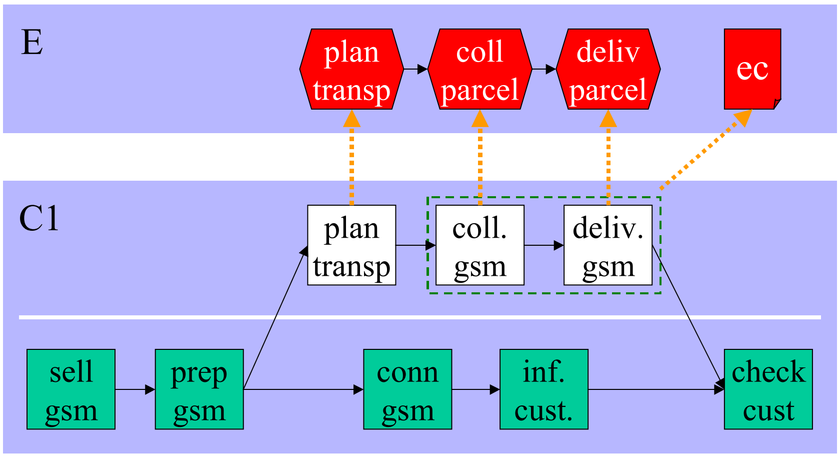

Figure 9: example external process (top), projected from conceptual (bottom)

Enactment characteristics specified in the conceptual process specification are mapped to separate enactment clauses at the external level (shown in the upper right corner in the figure). Both process specification and enactment clauses must be integrated into one transferable process specification.
In Figure 10, we see the mapping from conceptual to external level in two concrete specification techniques. The bottom part shows the conceptual process in a Petri Net as in Figure 5. The external level specification in the top part of Figure 10 is based on ebXML - hence we use an UML activity diagram as the graphical process specification technique (see also Section 8.1.1). The mapping between Petri Net and activity diagram is straightforward in this simple case: each transition in the Petri Net is mapped onto an activity in the activity diagram. In [Es01b], a comparison is made between the two techniques, elaborating the differences between them in the context of workflow management.

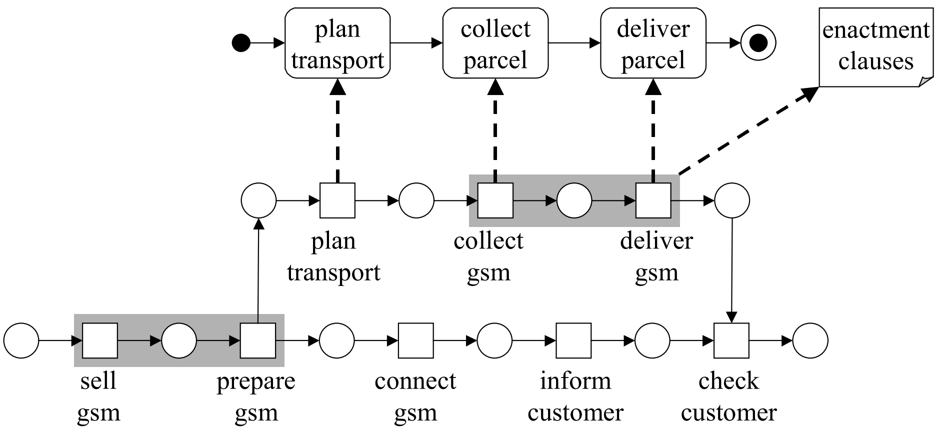

Figure 10: example external process (top), projected from conceptual (bottom)

4.5 Process-relevant data in external process specifications

The issue of a process model on an external level, i.e., the model of a process that an organization expects another organization to perform or that is performed for another organization, also encompasses the process-relevant data that is accessed on this level. Process-relevant data on the external level can be both data that a made available by the service provider to its customer and vice versa.
Similarly to the process flow model, the process relevant data of the conceptual level is projected and translated to the external level. This operation is guided by the usage relationship of the elements of the process flow model of the external level.

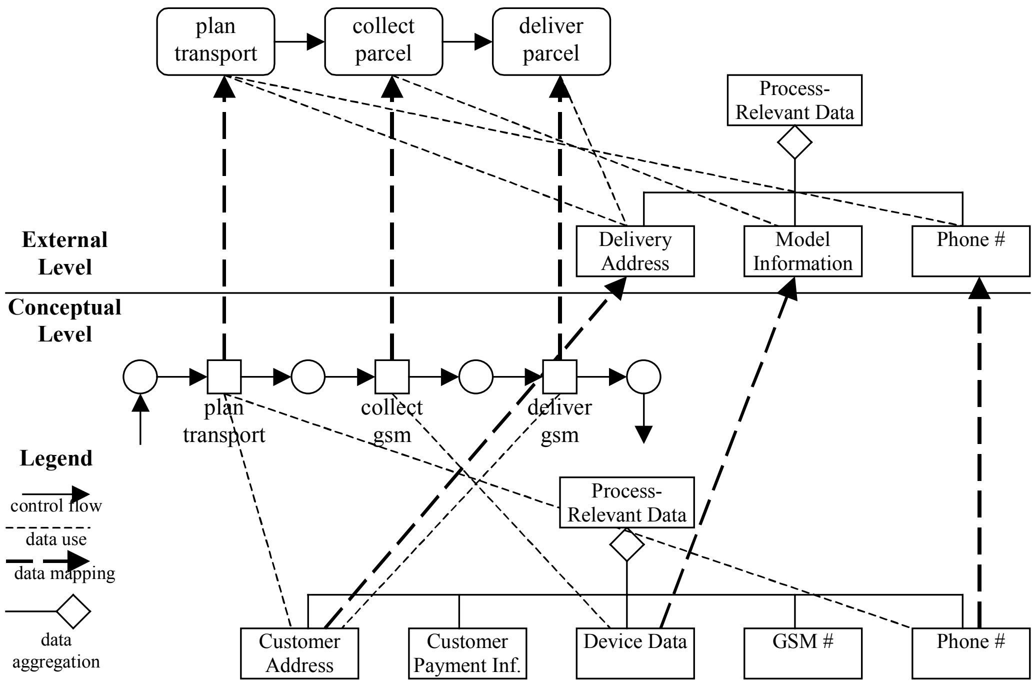

Figure 11: data usage in external process

Figure 11 shows an example of the relationship of process-relevant data on conceptual and external level – the relationship can be as complex as the picture looks. The conceptual level shows the process-relevant data items and the fragment of the conceptual process flow model that is mapped to an external level. The usage relationship between the steps and the data items is represented as light dotted lines. (The usage relationship of transition conditions is skipped at this point.) Steps and transitions of the process flow model are mapped to the external level as illustrated in the previous subsection (bold dashed arrows). The process-relevant data of the external-level process is created according to the usage relationship of the mapped conceptual process flow model elements. The mapped elements use only a subset of the process-relevant data items on the conceptual level. The external process-relevant data is a projection of this data structure and some data items are translated.

4.6 Transactional aspects

Transactional properties of a process specified at the conceptual level have to be mapped to the external level to obtain the required cross-organizational transactional behavior of service enactment. More specifically, this means:

- A part of a process at the external level that corresponds with an atomic subprocess at the conceptual level has to be atomic as well (be it strictly or relaxed atomic).
- A part of a process at the external level that corresponds to a subprocess at the conceptual level that has specific isolation properties must have isolation properties as well. These properties are with respect to data items related to (given the mapping discussed before) those involved in the isolation property at the conceptual level.

Many specification techniques at the external level do not include process primitives for the specification of transactional properties. For this reason, an external process specification often has to be extended with specific transaction management clauses. In Figure 9, this is illustrated by the 'ec' part of the external specification. This approach is elaborated for example in the CrossFlow project [Von00] – we discuss this in more detail in Section 8.3.

4.7 External process specifications in electronic contracts

Electronic contracts form the basis for the automated formation of dynamic virtual enterprises. One of the most important aspects of this formation is the alignment of the business processes of the cooperating organizations. For this reason, the external process specification is a key element in electronic contracts in the domain of service outsourcing [An01a, Koe00].

On the external level, there are two types of process specifications that are relevant to electronic contracts: establishing contracts between service consumer and provider and enacting contracts based on established contracts. As the main topic of this paper is e-service design and enactment, we focus on the specification of processes related to contract enactment, i.e., processes for enacting outsourced services. For reasons of completeness, however, we briefly describe both types of processes below.

The first type of process specification describes the establishment of an electronic contract, i.e., the process of negotiation and exchanging preliminary data between two organizations. This process is specified in an electronic contract offer and subsequently – if necessary – in an electronic contract. The reason to include this process specification in a contract is for example to achieve contract reusability. The established contract can be stored and subsequently, when new business relations between the parties appear, the preliminary process can be repeated in an automated manner and will lead to a new instance of the contract. Contract templates can be used here to facilitate the reuse of contract specifications [Hof00]. This kind of process specification can also be reflected at the conceptual and internal specification levels.

The second type of process specification concerns a specification of the outsourced service. This specifies the process execution chronology and components, and supports the cross-organizational process monitoring and control. Specification of these processes at the right level of granularity at the external level is of high importance for the proper execution of the services accompanying contract enactment, e.g., contract progress monitoring[5], contract enforcement, etc. The granularity is determined in the design process at the conceptual level, as described in Section 1. The design determines which process specifics can remain hidden from the contractual relation between a service consumer and a provider. These observations for the process

[5] In this report, we focus on process aspects of contracts. See for example [An01b] for a discussion of more general *provisions* in contracts.

specifications embedded in the electronic contract are an underpinning for the need of a conceptual level that will play a mediating role and will allow independence between the internal and external levels.
In [An01b, Ang02], we distinguish four main parts of a contract:

Who: this part describes which parties are involved in the contract; in the context of this report, this part describes the service consumer and service provider.

Where: the *where* part describes the context of the contracts, e.g., the geographical and legal contexts.

What: the *what* part describes what is traded in the contract; in the context of this report, it is the service being outsourced – the second type of process described above.

How: the *how* part describes how a contract is made – the first type of process as described above.

In the context of service outsourcing, the *what* part includes the external process specification as discussed in this section.

5 THE INTERNAL LEVEL

The internal process specification is used to have local parts of cross-organizational processes enacted by process support systems. Below, we first describe the nature of internal process specifications and spend a few words on specification techniques at this level. Next, we discuss the mapping of specifications from the conceptual level to the internal level. Then we present our running example at the internal level. We treat data and transaction management aspects in some detail.

5.1 The nature of internal process specifications

Internal process specifications are used to describe enactment processes by organizations participating in cross-organizational processes.
Workflow management systems are a general infrastructure for the automated support of business processes enactment. Often, these systems are separate entities in an information system infrastructure, sometimes they are embedded in other systems, like ERP systems. Hence, the internal process specification level can be based on a workflow process definition language (WFPDL). WFPDLs are discrete in nature and often use Petri Net based process models, so they match the requirements as discussed at the conceptual level. They usually contain primitives allowing easy specification of various process constructs, for example various forms of process splits and joins. We show an example WFPDL in the sequel of this section.
Data specification is usually performed by means of an underlying database schema. Some WFPDLs, however, provide also means for explicit data flow specification. We treat data-related aspects in more detail in Section 5.5.
Most commercial WFPDLs contain little support for transaction specification. Some languages resulting from research efforts are better equipped in this respect. We address these issues in Section 4.6.
The internal process specification contains a mapping of the activities in the process to the enactment resources available in the organization. These resources include the employees in the organization and the back-end information systems employed in the business processes. WFPDLs contain primitives that cater for this specification.

5.2 Process specification techniques

As the focus of the internal level is process enactment, the process specification techniques used at this level have to be geared towards process and data management systems.
In practice, a WFPDL is usually WFMS-specific: each WFMS has its own WFPDL. This heterogeneity is one of the reasons for having a separate conceptual specification level. Although most of the languages conform to standard concepts [WF01] defined by the Workflow Management Coalition [WM01], portability across systems is usually poor.
In Section 5.4, we will show two process definition languages of commercial WFMSs as an illustration.

5.3 Mapping the conceptual level to the internal level

Abstract process models defined at the conceptual level have to be mapped to workflow models in a WFPDL. This requires a syntactical translation, but also the addition of resource specifications as mentioned in Section 5.1. Given the global similarity of the natures of process primi-

tives in conceptual specification language and WFPDL, process translation is in most practical cases not problematic. In specific cases, however, the precise semantics of constructs at both levels have to be analyzed to obtain a correct translation. Clearly, quality characteristics identified at the conceptual level have to be taken into account in the translation – such that guarantees specified in contracts at the external level are in fact delivered at the internal level.
In the lower part of Figure 12, we see part of the internal process mapped from the example conceptual process specification introduced before. In the upper part, we see the relevant parts of levels 1 and 2 of the conceptual specification (for reasons of brevity, we do not show the entire processes). Essential in the mapping from conceptual to internal process is the fact that the internal level is determined by platform specifics, whereas the conceptual level does not. To illustrate this, we assume that the platform used to execute the internal level does not support subprocesses, i.e., cannot handle nested process structures. Hence, we have to flatten the levels of the conceptual process (C1 and C2) into one single internal process level (I1).

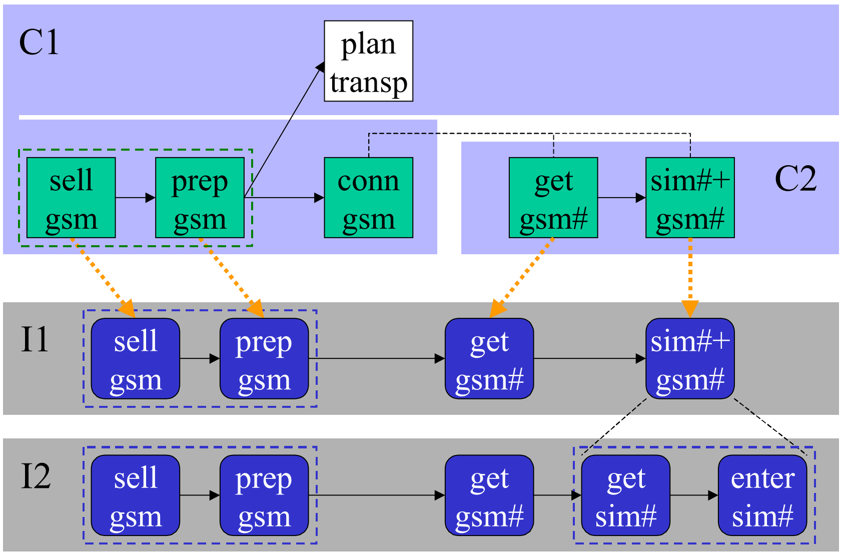

Figure 12: mapping conceptual to internal level

We also assume that the process enactment platform does not allow access to multiple back-end systems in one activity. Activity 'sim#+gsm#' (used to couple the SIM number to the GSM number) does require access to two back-end registration systems in our example, however. For this reason, this activity has to be exploded into two lower-level activities – together with the execution characteristic indicating that these two activities should be executed atomically. The fact that subprocesses are not supported, requires that the other activities are copied to this level as well.
Execution characteristics at the conceptual level have to be mapped to primitives offered by the enactment platform, as indicated by the dotted boxes in Figure 12.
Note again that refinement only takes place at the internal level if technology limitations dictate this. Refinement from a functional point of view takes place at the conceptual level.

5.4 Example internal process specification

For illustrative purposes, we present example internal process specifications in the context of two workflow management systems: Sema's FORO and IBM's MQSeries Workflow.

5.4.1 FORO WFMS

In Figure 13, we show level 1 of the internal process specification in the graphical version of the WFPDL of the FORO workflow management system [Sem01]. We see that the conceptual process specification can be mapped almost directly onto the internal specification. Start nodes, end nodes, splits and joins in the process are, however, modeled explicitly in the FORO FWPDL. Also, atomic tasks and subprocesses are distinguished between explicitly (note that tasks 'connect gsm' and 'inform customer' are represented by a different symbol). For reasons of brevity, we do not illustrate the addition of resource mappings here.

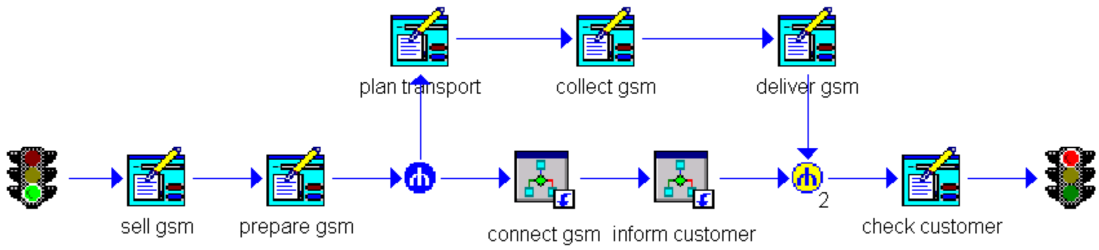

Figure 13: example internal process in FORO WFPDL

Note that the example mapping from conceptual to internal specification is rather straightforward. Depending on the functionality of the process enactment platform, the mapping may be more complex. If the process enactment platform would not support nested processes (subprocesses), for example, the nested structure of the conceptual specification would need to be 'flattened'. If the process enactment platform would not allow access to multiple back-end systems in one task, the task 'sim#+gsm#' at the conceptual level would have to be split into two tasks 'get sim#' and 'enter sim#'. The resulting process specification at the internal level is shown in Figure 14.

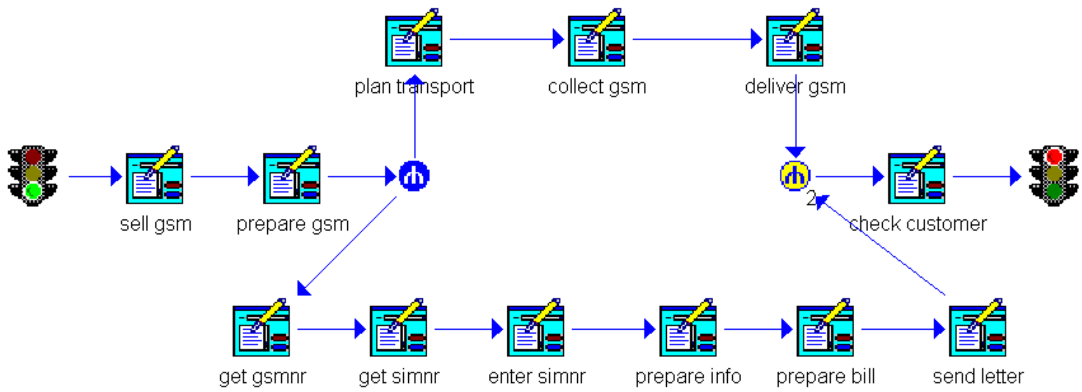

Figure 14: example internal process specification for limited platform

Note that this level is a refinement of levels 1 and 2 of the conceptual level. This refinement is made at the internal level, as it is driven by platform specifics, not by conceptual process functionality. As such, the distinction between refinement at conceptual and internal process levels caters for both a separation of concerns and portability of conceptual process specifications across implementation platforms.

5.4.2 MQSeries Workflow WFMS

In Figure 15, we show the example process of Figure 13 in the graphical WFPDL of MQSeries Workflow [IBM00, IBM01]. We see that splits and joins are not explicitly modeled in the graphical syntax (they follow from transition conditions on the control flow connectors). Also, start and end of the process are not modeled explicitly. Note that MQSeries Workflow does support explicit data flow – to be consistent in the running example, we have not used this feature here (see also Section 5.5).

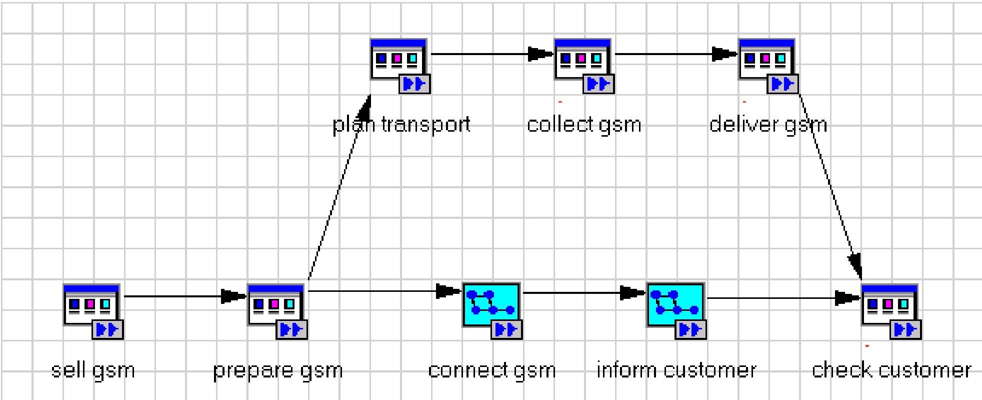

Figure 15: example internal process in MQSeries Workflow WFPDL

Figure 16 shows the same example process in (highly simplified) textual notation. The figure shows the mapping to one resource class (application software) – mapping to actors is left out.

```
PROCESS 'E-Service Example'
  PROGRAM_ACTIVITY 'sell gsm'
    START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
    PROGRAM 'SellGSM'
  END 'sell gsm'
  PROGRAM_ACTIVITY 'prepare gsm'
    START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
    PROGRAM 'PrepareGSM'
  END 'prepare gsm'
  PROCESS_ACTIVITY 'connect gsm'
    START AUTOMATIC WHEN AT_LEAST_ONE CONNECTOR TRUE
    PROCESS 'ConnectGSM'
  END 'connect gsm'
  PROCESS_ACTIVITY 'inform customer'
    START AUTOMATIC WHEN AT_LEAST_ONE CONNECTOR TRUE
    PROCESS 'InformCustomer'
  END 'inform customer'
  PROGRAM_ACTIVITY 'check customer'
    START MANUAL WHEN ALL CONNECTORS TRUE
    PROGRAM 'CheckCustomer'
  END 'check customer'
  PROGRAM_ACTIVITY 'plan transport'
    START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
    PROGRAM 'PlanTransport'
  END 'plan transport'
  PROGRAM_ACTIVITY 'collect gsm'
    START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
    PROGRAM 'CollectGSM'
  END 'collect gsm'
  PROGRAM_ACTIVITY 'deliver gsm'
    START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
    PROGRAM 'DeliverGSM'
  END 'deliver gsm'
  CONTROL FROM 'sell gsm' TO 'prepare gsm'
  CONTROL FROM 'prepare gsm' TO 'connect gsm'
  CONTROL FROM 'connect gsm' TO 'inform customer'
  CONTROL FROM 'inform customer' TO 'check customer'
  CONTROL FROM 'prepare gsm' TO 'plan transport'
  CONTROL FROM 'plan transport' TO 'collect gsm'
  CONTROL FROM 'collect gsm' TO 'deliver gsm'
  CONTROL FROM 'deliver gsm' TO 'check customer'
END 'E-Service Example'
```

Figure 16: example internal process in simplified MQSeries Workflow WFPDL

5.5 Process-relevant data in internal process specifications

On the internal level, we need to map the process-relevant data of the conceptual level to the right internal data management models that are used by a particular process management system and the implementations of the particular steps of the process. In currently available process management systems we can find two approaches to represent process-relevant data on the internal level.

In one approach, the central data store approach, all process-relevant data is stored in a central database and both process management systems and implementations of the individual steps access the data there. This database can follow any suitable data model, e.g. relational or object-relational. A representative for this approach is Foro [Sem01].

The other approach is called the data flow approach. Here, the internal process model is amended by a data flow graph that outlines how data flows from one step to subsequent ones. Each step has an in-container and an out-container. The data flow between two steps connects the out-container of one step with the in-container of another. This approach explicitly outlines the usage relationship of the conceptual level. IBM's MQ Series Workflow follows this approach [IBM01].

For the central data store approach the process-relevant data items are mapped to the appropriate data structures, e.g. tables, and the process management systems as well as the step implementations need to understand how to access these data items. In the data flow approach, the process-relevant data items are mapped to data flows between steps.

5.6 Transactional aspects

Transactional properties of processes specified at the conceptual level have to be mapped to the internal level in order to be guaranteed during process enactment. The way this mapping takes place is highly dependent on the available transaction support by the used process enactment systems – as represented in the specification languages at this level.

Many internal level specification languages – especially the commercial class – provide little support for explicit specification of transactional aspects. Often, transactional behavior is implicit in the process specification, e.g., each activity in a process is an atomic and isolated unit. In some environments, transaction support is completely missing.

If transaction support is too limited, transaction properties can be explicitly coded into a process specification. In the Exotica approach, for example, compensation behavior is added to a process specification to provide relaxed atomicity features [Alo96]. To do so, a conceptual specification of transaction properties is translated to additional process constructs that provide the transaction management functionality.

Some prototype systems offer more flexible transaction specification facilities. The WIDE system [Gre99], for example, offers two-level transaction specification facilities for both long-running processes with relaxed transactional behavior [Gre97] and short-running processes (business transactions) with strict transactional behavior [Gre97, Boe98].

6 REFERENCE ARCHITECTURE

In this section, we present a reference architecture for e-process support. This reference architecture is intended to be a general blueprint for analysis or design of concrete architectures for dynamic service outsourcing[6]. Hence, we will build the architecture from abstract modules, not related to specific technology. Below, we first paint the global (overall) architecture. After that, we elaborate sub-architectures of this global architecture.

As our approach leads to the existence of multiple interrelated process specifications per process, management of process specifications is an important practical aspect. For this reason, we discuss specification management aspects of our framework in Section 7. A number of concrete architectures have been designed that address parts or aspects of our reference architecture. In Section 8, we will therefore relate the reference architecture to these existing concrete architectures.

6.1 Global architecture

In the previous sections, we have introduced a service framework with process specifications at the conceptual, external and internal levels. These process specifications are used in three different phases of a process lifecycle:

Process design: in this phase, a process specification is constructed at the conceptual level and mapped to the external and internal levels (as illustrated in Figure 3).

Service contracting: in the contracting phase, a virtual enterprise is dynamically formed by contracting a service, based on the process specification at the external level.

Service enactment: in the enactment phase, local parts of a service are enacted by process and data management systems in the participating organizations, based on their internal process specifications; synchronization of the local service enactment is performed on the basis of the external process specification agreed upon in the contracting phase.

To obtain a clear separation of concerns, we 'partition' the architecture for e-process support with respect to the above three phases.

The result is shown in Figure 17. We see a functional component for each of the three phases, linked by five types of specifications: one for each of the three levels of our framework and two that describe the mappings between them (again as illustrated in Figure 3). Note that solid lines in the figure depict data flow relationships and dotted lines depict control relationships.

[6] A comparable approach has been used for workflow management systems in [Gre98])

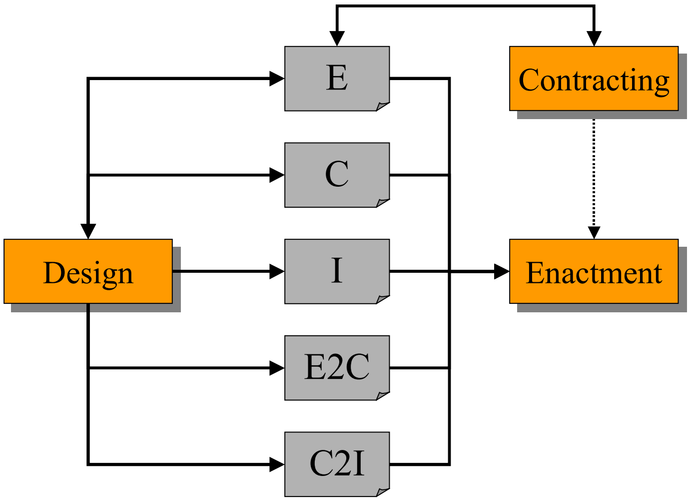

Figure 17: global architecture

Each functional component is discussed in a subsection below: design in Section 6.2, contracting in Section 6.3 and enactment in Section 6.4. The link between the phases is mainly formed by various specifications, i.e. digital documents residing in data stores. Hence, at the top level, the architecture can be classified as a repository-based architecture [Bas98].

6.2 Process design

The process design sub-architecture of the reference architecture is depicted in Figure 18. It consists of two groups of components: one for support of the activities related to the actual design of conceptual process specifications and one for translating these specifications to the external and internal levels.

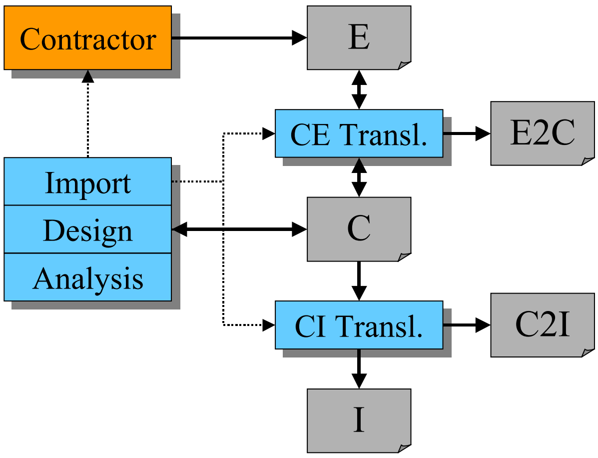

Figure 18: architecture – design

The components related to the actual design of conceptual process specifications are the following three:

Design module: this module provides the core functionality for the design of conceptual processes, as discussed in Section 3.4; its input are actions of the process designer and existing conceptual process specifications to be modified, its output are new conceptual process specifications.

Analysis module: this module provides the functionality for analyzing and verifying conceptual process specifications, as discussed in Section 3.7; its functionality is controlled by the design module.

Import module: the import module is used to import external specifications by means of the contractor module (discussed in Section 6.3) in case a logical process specification has to be based on a standard to be retrieved externally; the imported external specification is translated to the conceptual level by the CE translator module (see below); the import module is controlled by the design module and controls the contractor and translator modules with respect to the import functionality.

In our three-level approach, conceptual specifications are mapped to external and internal levels (as depicted in Figure 3). Hence, we distinguish the following two components for process specification translation:

CE Translator: the Conceptual-to-External (CE) translator translates process specifications from the conceptual to the external level and vice versa, as discussed in Section 4.2; when translating a conceptual to an external specification, it also has an E2C mapping as output, to be used in service enactment (see Section 6.4).

CI Translator: the Conceptual-to-Internal (CI) translator translates process specification from the conceptual to the internal level, as discussed in Section 5.3; it also produces a C2I mapping as output, to be used in service enactment (see Section 6.4).

6.3 Service contracting

In dynamic service contracting, service consumers and providers find each other through some kind of market place mechanism (usually through an intermediary) and engage in a business relationship.

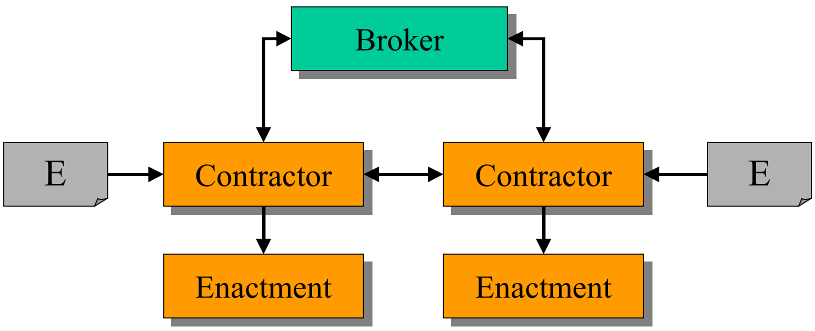

Figure 19: architecture - contracting

The abstract sub-architecture for service contracting is depicted in Figure 19. In the bottom of the figure, we see the enactment sub-architecture that we elaborate in the next subsection.
The main components for service contracting are the following two:

Contractor: the contractor module at a service consumer side finds compatible service offerings in a market through a broker; at a provider side, it advertises service offerings; the contractor uses an external process specification as the description of the service; two contractors can actually establish a contract for service enactment.

Broker: the broker is the intermediary in a specific service market; it brings together two contracting modules.

In this paper, we do not deal with the details of brokering – see for example [Hof01] for an operational approach to this subject. For brokering, standards like UDDI [UD00] may be used.

6.4 Service enactment

The abstract sub-architecture for service enactment (cross-organizational process execution) is depicted in Figure 20. In the left-hand side of the figure, we see the various specifications produced by the components of the design sub-architecture (as shown in Figure 18).

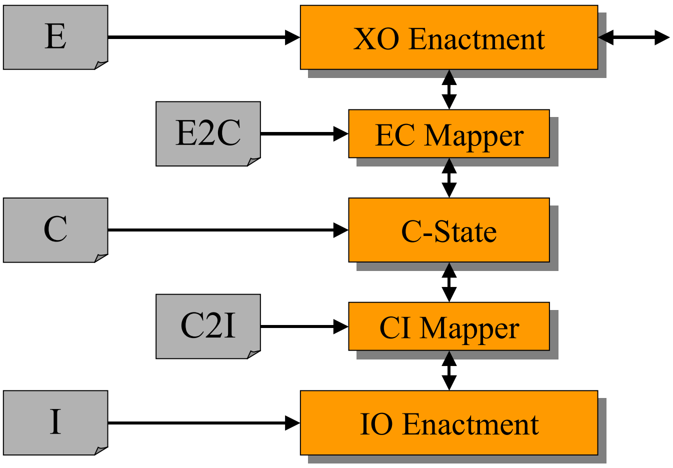

Figure 20: architecture – enactment

From bottom to top, we have the following components in the enactment sub-architecture:

IO Enactment module: the intra-organizational (IO) enactment module performs the actual process enactment within the bounds of one organization. To do so, it interprets the internal process specification. Events related to outsourced activities are communicated to the CI mapper. The IO enactment module can be based in 'traditional' workflow management technology.

CI Mapper: the conceptual-to-internal mapper maps process events between the conceptual and internal level. To do so, it interprets the C2I specification.

C-State module: the C-state module is a state machine that keeps the state of a process at the conceptual level. Upon reception of events from the CI mapper, it activates the EC mapper and vice versa. The C-state module includes functionality for observing the execution of the process at the conceptual level.

EC Mapper: the external-to-conceptual mapper maps process events between the external and conceptual level. To do so, it interprets the E2C specification.

XO Enactment module: the cross-organizational (XO) enactment module is responsible for the process enactment across the organizational boundary. It interprets the external level process specification. It communicates with its peer XO enactment module in the other organization. Events are internally communicated through the EC mapper.

Note that the architecture we advocate uses a two-step mapping between process events at the cross-organizational (external) respectively intra-organizational (internal) level. This two-step approach allows process monitoring at the conceptual level, separation of concerns when dealing with external and internal aspects, and the use of heterogeneous external and internal environments within the scope of a single logical process. A direct translation, on the other hand, may provide a less complex and more efficient solution in simple environments, however. The direct approach has been used for example in the CrossFlow project (see Section 8.1.2 for a discussion).

7 SPECIFICATION MANAGEMENT

The proposed approach to service outsourcing leads to multiple interrelated process specifications at the three levels for each service type. Interrelations include one-to-one and one-to-many mappings between specifications at the conceptual and external respectively the conceptual and internal levels. Usually, an organization is involved in the execution of multiple service types – depending on the type of organization this can be a few or a large number. Further, several versions of a service specification may exist simultaneously.

Efficiency in storage and retrieval of service specifications and effectiveness in guarding the consistency of service specifications is an important aspect in this context. Consequently, an infrastructure for specification storage and management is needed. In this section, we describe a high-level storage architecture and a first approach to an integrated data model for three-level specification management.

7.1 Specification storage

Storage of all specifications in central process specification database is an approach to match consistency and non-redundancy requirements. This database has a semi-structured data format with an XML basis. Below, we first discuss the specification storage architecture. Then, we pay attention to the relation of process specification formats to the XML basis of the architecture.

7.1.1 Storage architecture

The XML-based specification storage can be supported by a database management system (DBMS) with a general-purpose XML front end that provides the mapping between XML documents and storage structures provided by the DBMS – in practice usually relational tables[7]. Mapping XML documents to relational storage is currently being researched in several contexts, e.g. [Sha99, Flo99].

The XML front-end (FE) is coupled to specific filters for the specification languages at the three specification levels. The resulting architecture is shown in Figure 21. Note that the I-Filter translates both to the general XML format and directly to a relational representation – the latter is required for process management systems that read their process specification directly from a database.

[7] Of course, the use of a dedicated database management system for semi-structured data would also be possible (e.g. [McH97]), but coupling to standard database technology is more appropriate in practical settings.

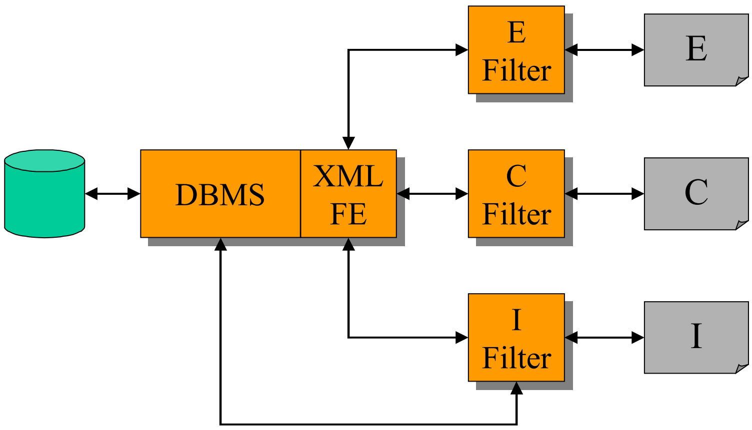

Figure 21: data management architecture

7.1.2 Process specification formats and XML

As discussed in this paper, specifications at the external level are usually stated in a business-oriented language based on XML, e.g., ebXML. Process-oriented XML-based languages are also being developed for inter-organizational use, e.g. XRL [Aal02]. Mapping to XML infra-structures is thus trivial at this level.

At the conceptual level, this requires a mapping of the used specification technique, for example Petri Nets, into XML. At a number of places, XML representations of Petri nets are currently being developed. Examples are the Petri Net Markup Language [PM01] and the before mentioned XRL [Aal02].

At the internal level, an XML representation of a WFPDL is required. The application of XML for workflow management currently finds some application in Wf-XML standard of the Workflow Management Coalition [WM01], although this language does not cover a complete WFPDL. Commercial workflow management systems, e.g. IBM's MQSeries Workflow [IBM01], are being developed towards XML support.

7.2 Specification data model

Figure 22 shows (part of) an approach towards a data model in EER notation for the three-level process specification storage. Entities on the external level are prefixed with 'e', on the conceptual level with 'c', and on the internal level with 'i'.

In this data model, we see a multiple inheritance relation. From left to right, we see a specialization from general process element to specific process element (functional specialization). From right to left, we see a specialization from general level to specific level (abstraction level specialization).

We have placed the mapping relations (indicated as 'map') between elements of the three levels between functionally most general entities, such that they can be inherited by more specific entities. Structural relations are placed between specific process element types that are generalized with respect to their level (for example the consists-of relation, indicated as 'c.o.', between sub-process and activity).

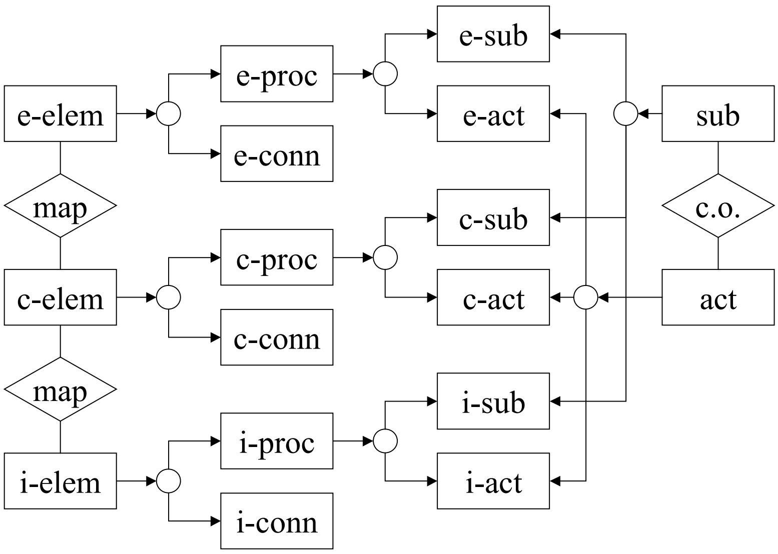

Figure 22: specification data model

8 RELATED WORK

In this section, we investigate how existing approaches support the definition and relationship of the proposed three-level model. The relationship between conceptual process models and internal models has been addressed to some extent in the fields of software engineering, information systems design and business process modeling [Bom97]. However, the relationship between external and internal specifications became more relevant in the recent years with the advent of widespread cross-organizational business process integration, albeit some EDI-based solutions being around since a number of years.
We treat related work in three categories: first we discuss relevant standards in Section 8.1, then we introduce products that are based on proprietary interaction in Section 8.2, and we end with Section 8.3 presenting a discussion of research projects.

8.1 Standards

It is difficult to keep up to date in the area of business-to-business standardization efforts. However, industry efforts mainly focus on ebXML [eb01a] and Web services [W301], which address the same issue of standardizing business-to-business interaction starting from messaging up to process integration. Based on these standards, industry-specific standard processes are defined by various organizations, e.g. RosettaNet [RN01] and BizTalk standards repository [Biz01].

8.1.1 ebXML

ebXML is an initiative governed by UN/CEFACT and OASIS, joined by major companies and standardizing consortia, that aims to provide a framework for establishment of business relations and subsequent execution of business transactions [eb01a]. A global overview of the ebXML approach is shown in Figure 23.

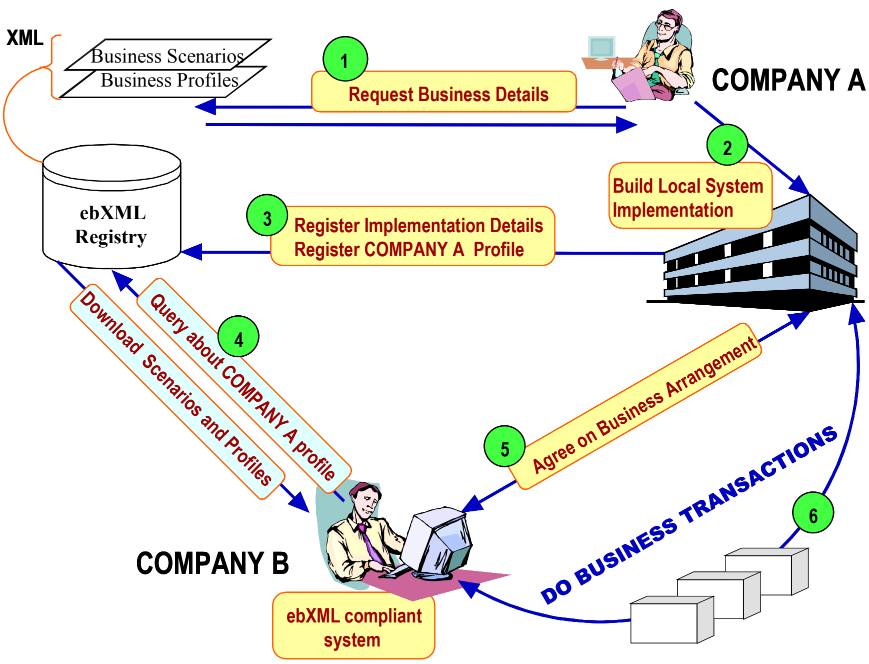

Figure 23: ebXML architecture overview (from [eb01a])

ebXML provides a process modeling technique based on the UN/CEFACT Modeling Methodology (UMM) that utilizes UML. The modeling technique employs UML activity diagrams and sequence diagrams in both a graphical and a textual, XML-based format. In Section 4.4, we have shown an example activity diagram.
The process modeling and specification stage in the ebXML framework relates to the conceptual level of our three-level process model. However, ebXML does not differentiate process specifications at the conceptual level from process specification at the external level. As these two levels are mixed up and there is no explicit attention for refinement of processes in the approach, detailed mappings to internal level process specifications may be hard to obtain.
As a result of process modeling in the ebXML approach, Collaboration Protocol Profiles (CPPs) are created at the service consumer and service provider sides that reflect the process specification at the external level of both parties. The CPP includes specifications of the supported processes, transaction chronology descriptions, etc. and is the contract offer specification described in Section 4.7. On the basis of the CPPs of both parties, a Collaboration Protocol Agreement (CPA) is created. The CPA is the electronic contract that specifies the agreed processes that will be performed, their chronology and possible interactions between parties.
The existence of ebXML as a global standard specification allows achieving the interoperability required on the external level.

8.1.1.1 ebXML Business Process Specification Schema (BPSS)

In ebXML, the alignment of the partners' business processes is depicted in the ebXML Business Process Specification Schema (BPSS) [eb01b]. BPSS provides a framework for business processes specification. A specification that conforms to the BPSS is called an ebXML Business Process Specification. The BPSS is a subset of the UN/CEFACT Modeling Methodology (UMM) and is provided to support direct specification of the elements, properties and semantics for the business collaboration definition. In an ebXML business collaboration, two or more parties interact through business transactions. Business transactions are atomic and are the lowest level of decomposition of the partners' processes. They are ordered by the business transaction choreography. The choreography is described through activity diagram concepts. Each business transaction is realized through business document flows between the trading parties.
The BPS of a trading partner is the basis for the creation of a CPP and respectively of a CPA. It is incorporated in or referenced by them. The XML version of the ebXML BPS is stored in the ebXML repository.
In ebXML, no explicit distinction is made between partners' process on the external and conceptual level. Therefore, mappings between the external and conceptual processes are not discussed.

8.1.2 Web services

Another standardization process, partially running parallel to the ebXML standardization effort, is the so-called Service Web approach. This approach resulted from the necessity, web applications to interact with each other in a dynamic web environment. Because of this necessity, several standards were developed, which taken together should provide the required application integration.

8.1.2.1 WSDL

Simple Object Access Protocol (SOAP) is a protocol specification for exchanging messages across Internet and for invoking methods, resident on arbitrary servers. However, in order to invoke remote services, a standard description for these services is required. Web Services Description Language (WSDL) is a standard for description of web services [W301]. WSDL defines the services that can be invoked and the requirements for their invocation, thus defining the content of a SOAP message. UDDI (Universal Description, Discovery, and Integration) addresses the location of the web services. It defines a registry, where service providers can register their services and service consumers can look up for them. SOAP is the communication channel to the UDDI registry. In general, any service can be exploited as a web service, provided that there are appropriate mappings to its internal organization process specification. The approach allows integration of the processes of organizations at the external level of the proposed framework. However, it is not devoted to the modeling of the exposed services, neither to their mapping to the internal organization processes specifications.

8.1.2.2 WSFL

The Web Services Flow Language is a specification language for cross-organizational processes whose steps are implemented as Web Services [WS01]. It allows the definition of steps based on their respective WSDL description and the control flow between these steps. The control flow is represented as control flow connectors, which can have a transition condition associated to them referring to process-relevant data The process-relevant data is represented in the form of data flows between steps. Data flows can only follow the control flow, directly or indirectly. Both control flow and data flow may contain parallel branches.

WSFL also provides a means for process abstraction. Process definitions themselves can be defined and addressed as Web Services. These WSDL-encapsulated processes can be again part of a higher-level process definition.

WSFL is IBM’s proposal of a process specification language for Web Services.

8.1.2.3 XLANG

XLANG is a proposal by Microsoft to extend WSDL by control flow elements [XL01]. WSDL definitions can be amended by control flow expressions that define the sequencing, repetition and other interdependencies of operations defined in ports of a WSDL file. This is referred to as behavior in XLANG. To identify operations of the same interaction, correlation sets-data items of messages-can be defined. Also, the transactional behavior of a service can be expressed. A part of the control flow of a service can be declared a transaction. A transaction can be associated with a compensating control flow that undoes its effects in case of failure. Transaction contexts can be nested for fine-grained compensation specification.

Multiple XLANG-extended WSDL specifications, potentially from different organizations, can be stitched together in a contract. While the purpose of XLANG’s WSDL extensions is to describe the behavior of a single service, a contract describe how multiple services relate together to form a, potentially, cross-organizational process. The main mechanism used in a contract is to relate incoming operations of one service with outgoing operations of another service, thus defining a connection between these services.

8.1.3 RosettaNet

RosettaNet [RN01] is a standardization effort, governed by a consortium of Information Technology, Electronic Components and Semiconductor Manufacturing companies. RosettaNet provides standards that facilitate the electronic exchange of business documents between partners. Fundamental for the RosettaNet standard are the Partner Interface Processes (PIPs). External processes of a company are realized through the implementation of PIPs specifications. A PIP is a specification that aligns the business processes between the trading companies. It specifies the business roles for a given business process, the business activities between these roles, and the type, content and sequence of the business documents exchanged by the partners. A PIP specification is described by means of three views, i.e., Business Operational View (BOV), Functional

Service View (FSV), and the Implementation Framework View (IFV). The BOV captures the business semantics of the performed activities and of the information flow. The FSV is derived from BOV and specifies the design of the network components that execute the PIPs. IFV specifies communication requirements.
In RosettaNet the e-business supply chain domain is divided into clusters. Each cluster is divided into segments. In each segment exist one or more PIPs.
RosettaNet defines standards for the alignment of the external processes of companies. From the perspective of the three-level framework that we have described, this approach requires specification of internal process to be in conformance with the external processes standard. This leads to the possibility the company to be forced to change existing internal processes or to implement new ones in order to be able to use the standard. Relations between the external processes and the internal processes of a company are out of the scope of RosettaNet.
RosettaNet is a representative of a large number of standardization fora that provide a platform to define very detailed, industry-specific business-to-business interaction specifications. This extends to both a representation for describing interaction as well as an organizational context in terms of procedures and rules. Another representative is, for example, BizTalk [Biz01].

8.2 Proprietary systems for cross-organizational process integration

There are a number of vendors of proprietary systems for cross-organizational process integration. A typical representative is Peregrine's Alliance Manager, formerly knows as Extricity Alliance [Per01]. Alliance Manager comprises a tool to define an external-level process and a runtime environment that facilitates the process management interaction among the individual parties and maps this interaction onto actions on a local process management system. Interfaces to many popular systems are available. The definition format and the runtime system are proprietary. All parties engaged in the interaction are supposed to run Alliance Manager. In addition to its proprietary format, RosettaNet PIPs and the associated XML-based runtime interaction are supported.

8.3 Research projects

In this section, we discuss two relevant research projects. We start with a discussion of the CrossFlow project that has partly inspired the work described in this report. Next, we pay attention to the WISE project. We see that neither of these projects clearly distinguishes three process specification levels as proposed in this report: CrossFlow merges conceptual and internal level, WISE merges conceptual and external level.

8.3.1 CrossFlow

In the CrossFlow project, concepts and technology for workflow support in dynamic virtual enterprises have been developed [Gre00, XF01]. In the context of this project, the formation of virtual enterprises is based on dynamic service outsourcing, as advocated in this paper as well.
Service offerings and service requests are specified in electronic contract templates [Koe00], which are matched by a service matchmaker. An established electronic contract is the basis for the dynamic generation of a service enactment infrastructure [Hof00], based on workflow management technology.

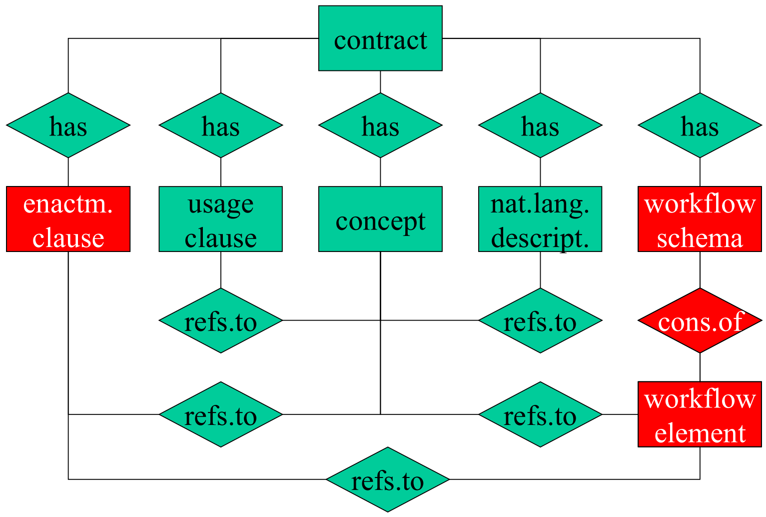

Figure 24: top level of CrossFlow contract model

8.3.1.1 External level

In the CrossFlow approach, external process specification is part of contract specification. For this purpose, the contract model contains a process specification submodel, based on the standard of the Workflow Management Coalition. The process specification submodel is augmented with an extensible set of enactment clauses, that specify service enactment characteristics, as discussed in Section 4.1. Apart from this, the contract model includes usage clauses, describing aspects of establishing contracts – this is a first step towards specification of establishment processes as discussed in Section 4.7. The contract model is reflected in a dedicated, XML-based contract specification language [Koe00].
In Figure 24, the top level of the CrossFlow contract model [Koe00] is shown. This model consists of five main entity clusters: a concept space that defines concepts relevant in the contract, a workflow schema that describes a service process, enactment clauses that specify how to execute the workflow schema, usage clauses that specify how to establish a contract, and a natural language description of the contract for human interpretation. The various clusters are interlinked to describe relations between them. The workflow schema and enactment clauses coincide with the external process specification as proposed in this paper. The concept space is essential for specifying the context of the service outsourcing, e.g., to identify the parties involved in service outsourcing.

8.3.1.2 Conceptual and internal level

The internal process specification in CrossFlow is formed by the workflow specification language of the workflow management system, in the case of the CrossFlow prototype IBM MQSeries Workflow [IBM01]. The mapping between external and internal specification levels is described in an Internal Enactment Specification [Lud00, Lud01].
In the CrossFlow approach, no separate conceptual level for process specification is used. This means that conceptual process design and analysis are not integrated into the CrossFlow approach. An example is the support for advanced cross-organizational transaction management in CrossFlow [Von00]. This is supported both at the external and internal level, but without conceptual analysis facilities integrated in the model. In complex situations, however, this analysis is desirable to determine the precise semantic effects of transaction management.

An internal enactment specification (IES) is used for a direct mapping of external onto internal level [Hof00]. As such, the IES is a 'combination' of the E2C and C2I specifications as discussed in Section 6.

8.3.2 WISE

The WISE project (Workflow based Internet SErvices) at ETH Zürich aims at providing a software platform for process based business-to-business electronic commerce [Alo99, Laz01]. In doing so, the project focuses on support for networks of small and medium enterprises. The software platform used in WISE is based on the OPERA kernel [Alo97].
The approach followed is different from that of CrossFlow: CrossFlow relies on cooperating pairs of autonomous workflow systems with a peer-to-peer relation (conforming to the reference architecture of Section 6), whereas WISE relies on a central workflow engine to control cross-organizational processes (called virtual business processes). A virtual business process in the WISE approach consists of a number of black-box services linked in a workflow process [Alo99]. A service is offered by an involved organization and can be a business process controlled by a workflow management system local to that organization – but this is completely orthogonal to the virtual business process.

8.3.2.1 Conceptual and external layer

Specification of virtual business processes in WISE is performed using the Structware/IvyFrame tool [Lie98], which is internally based on Petri Nets. This tool and its specification technique are used to construct both the conceptual structure of cross-organizational processes and the specifications of services exchanged between organizations in a virtual enterprise. Hence, it can be placed both at the conceptual and at the external levels of our framework.
The Structware/IvyFrame tool has, however, also characteristics related to the internal level, as it not only supports process creation, but also configuration management of underlying enactment platforms [Laz01].

8.3.2.2 Internal layer

The graphical representation produced by the Structware/IvyFrame process definition tool is compiled into a language called Opera Canonical Representation (OCR) [Hag99]. This language is used internally by WISE to create process templates. As OCR is focused towards process enactment in the context of a specific platform, we place it at the internal level of our framework. Note, however, that the internal level of WISE is a cross-organizational internal level.

9 CONCLUSIONS AND OUTLOOK

The use of contract-based dynamic service outsourcing opens ways to efficient handling of fine-grained cross-organizational processes. Efficient means that contractual outsourcing can take place in a fast and cheap way – quite different from the traditional paper-based situation. This paradigm allows the creation and dismantling of short-term virtual enterprises, thereby supporting highly dynamic cooperation in fast-changing markets. In some situations, this can even enable completely new business models.

In this report, we propose a three-level framework for the support of e-services in a dynamic service outsourcing context. The three-level approach to business process specification provides a clear separation of concerns in business process design, thereby increasing quality, flexibility and reusability of process specifications in cross-organizational settings. This separation of concerns is becoming increasingly important, as the complexity of automated cross-organizational processes grows through the advent of e-business. The three-level approach to service specification is matched by a three level reference architecture for service enactment, that forms a starting point for the analysis and design of e-service support systems.

A spectrum of work is being performed at the University of Twente that is related to the approach presented in this paper, partly inspired by the results of the CrossFlow project [Gre00, Hof00, Koe00, Von00, Hof01]. Support for e-contract handling is being researched [Ang02, Gre02]. This research is focused on service contracting in B2B environments. Application of the three-level framework for contracting processes is investigated in this context. Research into flexible architectures for cross-organizational process support is being performed. Models and architecture for cross-organizational transactional workflow processes are a further topic of research.

Contract-related projects in many forms are carried out at IBM Research. The work relates in particular to formats and architectures for specific agreement types such as service level agreements [Kel02, Lud02] and collaboration protocols, e.g., in the context of WSFL [WS01] and the trading partner agreement project [Dan01] that formed the basis of the ebXML CPPA standard [eb01c].

Acknowledgments

All former CrossFlow project members are acknowledged for their role in the CrossFlow project, on which part of the ideas presented in this paper are based. Special thanks go to Jochem Vonk for his assistance with WFMS software.

Bibliography

[Aal98] W. van der Aalst; *The Application of Petri Nets to Workflow Management*; Journal of Circuits, Systems and Computers, Vol. 8, No. 1, 1998.

[Aal02] W. van der Aalst, A. Kumar; *XML Based Schema Definition for Support of Interorganizational Workflow*; to appear in Information Systems Research.

[Ada98] N. Adam, V. Atluri, W. Huang; *Modeling and Analysis of Workflows using Petri Nets*; Journal of Intelligent Information Systems, Vol. 10. No. 2, 1998.

[Alo96] G. Alonso et al.; *Advanced Transaction Models in Workflow Contexts*; Procs. Int. Conf. on Data Engineering; New Orleans, Louisiana, USA, 1996; pp. 574-581.

[Alo97] G. Alonso, C. Hagen, H.J. Schek, M. Tresch; *Distributed Processing over Standalone Systems and Applications*; Procs. 23rd Int. Conf. on Very Large Databases; Athens, Greece; 1997.

[Alo99] G. Alonso, U. Fiedler, C. Hagen, A. Lazcano, H. Schuldt, N. Weiler; *WISE: Business to Business E-Commerce*; Procs. 9th Int. Workshop on Research Issues on Data Engineering; Sydney, Australia, 1999; pp. 132-139.

[An01a] S. Angelov, P. Grefen; *B2B eContract Handling - A Survey of Projects, Papers and Standards*; CTIT Technical Report 01-21; University of Twente, 2001.

[An01b] S. Angelov, P. Grefen; *A Framework for the Analysis of B2B Electronic Contracting Support*; Proceedings 4th Edispuut Conference - Multidisciplinary Perspectives on Electronic Commerce; Amsterdam, Netherlands, 2001.

[Ang02] S. Angelov, P. Grefen; *A Conceptual Framework for B2B Electronic Contracting*; Proceeedings 3rd IFIP Working Conference on Infrastructures for Virtual Enterprises; Sesimbra, Portugal, 2002.

[Bas98] L. Bass, P. Clements, R. Kazman; *Software Architecture in Practice*; Addison-Wesley, 1998.

[Biz01] *BizTalk Framework*; http://www.biztalk.org.

[Boe98] E. Boertjes, P. Grefen, J. Vonk, P. Apers; *An Architecture for Nested Transaction Support on Standard Database Systems*; Proceedings 9th International Conference on Database and Expert System Applications; Vienna, Austria, 1998; pp. 448-459.

[Bom97] M. Boman, J. Bubenko, P. Johannesson, B. Wangler; *Conceptual Modelling*; Prentice Hall, 1997.

[Dan01] A. Dan, D. Dias, R. Kearney, T. Lau, T. Nguyen, F. Parr, M. Sachs, H. Shaikh; *Business-to-Business Integration with tpaML and B2B Protocol Framework.* IBM Systems Journal, Vol. 40 No. 1, February 2001.

[Deh01] J. Dehnert, M. Gajewski, S. Lembke, H. Weber; *The Petri Net Baukasten: 2nd Installment*; Procs. 2nd Int. Coll. on Petri Net Technologies for Modelling Communication Based Systems; Berlin, Germany, 2001; pp. 43-54.

[Der01] W. Derks, J. Dehnert, P. Grefen, W. Jonker; *Customized Atomicity Specification for Transactional Workflows*; Procs. 3rd Int. Symp. on Cooperative Database Systems for Advanced Applications; Beijing, China, 2001.

[Dui00] M. Duitshof; *Logistics Prototype Deployment Report*; CrossFlow Project Deliverable D13; KPN Research, The Netherlands, 2000 (available via http://www.crossflow.org).

[eb01a] *ebXML Technical Architecture Specification v1.0.4*, February 16, 2001; http://www.ebxml.org.

[eb01b] *ebXML Business Process Specification Schema (BPSS 1.01)*; ebXML Business Process Project Team , May 2001, http://www.ebxml.org/specs/index.htm#technical_specifications (as of Dec 2001).

[eb01c] ebXML Collaboration-Protocol Profile and Agreement Specification v1.0, May 2001, http://www.ebxml.org.

[Elm94] R. Elmasri, S.B. Navathe; *Fundamentals of Database Systems*; Benjamin/Cummings, 1994.

[Es01a] R. Eshuis, R. Wieringa; *A Formal Semantics for UML Activity Diagrams - Formalising Workflow Models*; CTIT Technical Report 01-04; University of Twente, 2001.

[Es01b] R. Eshuis, R. Wieringa; *A Comparison of Petri Net and Activity Diagram Variants*; Procs. 2nd Int. Coll. on Petri Net Technologies for Modelling Communication Based Systems; Berlin, 2001.

[Flo99] D. Florescu, D. Kossmann; *Storing and Querying XML Data using an RDMBS*; IEEE Data Engineering Bulletin, Vol. 22, No. 3, 1999; pp. 27-34.

[Gre93] P.W.P.J. Grefen, P.M.G. Apers; *Integrity Control in Relational Database Systems - An Overview*; Journal of Data & Knowledge Engineering, Vol. 10, No. 2, 1993, North Holland - Elsevier; pp. 187-223.

[Gre97] P. Grefen, J. Vonk, E. Boertjes, P. Apers; *Two-Layer Transaction Management for Workflow Management Applications*; Proceedings 8th International Conference on Database and Expert System Applications; Toulouse, France, 1997; pp. 430-439.

[Gre98] P.W.P.J. Grefen, R.N. Remmerts de Vries; *A Reference Architecture for Workflow Management Systems*; Journal of Data & Knowledge Engineering, Vol. 27, No. 1; North Holland - Elsevier, 1998; pp. 31-57.

[Gre99] P. Grefen, B. Pernici, G. Sánchez (Eds.); *Database Support for Workflow Management: The WIDE Project*; Kluwer Academic Publishers, 1999.

[Gre00] P. Grefen, K. Aberer, Y. Hoffner, H. Ludwig; *CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises*; Int. Journ. of Computer Systems Science & Engineering, Vol. 15, No. 5, 2000; pp. 277-290.

[Gre01] P. Grefen, J. Vonk, P. Apers; *Global Transaction Support for Workflow Management Systems: from Formal Specification to Practical Implementation*; VLDB Journal, Vol. 10, No. 4; Springer, 2001; pp. 316-333.

[Gre02] P. Grefen, S. Angelov; *On Deep e-Contracting*; CTIT Technical Report 02-06; University of Twente, 2002.

[Hag99] C. Hagen; *A Generic Kernel for Reliable Process Support*; Ph.D. Dissertation, ETH Nr. 13114, 1999.

[Hof00] Y. Hoffner, H. Ludwig, C. Gülcü, P. Grefen; *Architecture for Cross-Organisational Business Processes*; Procs. 2nd Int. Worksh. on Advanced Issues of E-Commerce and Web-Based Information Systems; Milpitas, CA, USA, 2000; pp. 2-11.

[Hof01] Y. Hoffner, S. Field, P. Grefen, H. Ludwig; *Contract Driven Creation and Operation of Virtual Enterprises*; Computer Networks, Vol. 37, No. 2; Elsevier, 2001; pp. 111-136.

[Kel02] A. Keller, G. Kar, H. Ludwig, A. Dan, J. Hellerstein; Managing Dynamic Services: Approach to a Conceptual Architecture. To appear in Proceedings of IFIP/IEEE NOMS 2002.

[McH97] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, J. Widom; *Lore: A Database Management System for Semistructured Data*; SIGMOD Record, Vol. 26, No. 3, 1997; pp. 54-66.

[IBM00] *IBM MQSeries Workflow - Getting Started with Buildtime - Version 3.2.2*; IBM, 2000.

[IBM01] *IBM MQSeries Workflow*; http://www-4.ibm.com/software/ts/mqseries/workflow/; IBM, 2001.

[Koe00] M. Koetsier, P. Grefen, J. Vonk; *Contracts for Cross-Organizational Workflow Management*; Procs. 1st Int. Conf. on Electronic Commerce and Web Technologies; London, UK, 2000; pp. 110-121.

[Laz01] A. Lazcano, H. Schuldt, G. Alonso, H. Schek; *WISE: Process Based E-Commerce*; IEEE Data Engineering Bulletin; Vol. 24, No. 1, 2001.

[Ley95] F. Leymann; *Supporting Business Transactions via Partial Backward Recovery in Workflow Management Systems*; Procs. Datenbanksysteme in Büro, Technik und Wissenschaft; Dresden, Germany, 1995; pp. 51-70.

[Lie98] H. Lienhard; *IvyBeans - Bridge to VSH and the project WISE*; Procs. Conf. of the Swiss Priority Programme Information and Communication Structures, Zürich, Switzerland, 1998.

[Lud00] H. Ludwig; *CrossFlow Deliverable 6.c; Internal Contract Representation*. La Gaude, 2000. (http://www.crossflow.org, as of Mar 2002).

[Lud01] H. Ludwig, Y. Hoffner; *The Role of Contract and Component Semantics in Dynamic E-Contract Enactment Configuration*. In Procs. of the 9th IFIP Workshop on Data Semantics (DS9), pp 26 - 40, Hong Kong, 2001.

[Lud02] H. Ludwig, A. Keller, A. Dan, R. King; *A Service Level Agreement Language for Dynamic Electronic Services.* To appear in Procs. of WECWIS 2002.

[Mec01] M. Mecella, B. Pernici; *Designing Wrapper Components for e-Services in Integrating Heterogeneous Systems*; VLDB Journal, Vol. 10, No. 1, 2001; pp. 2-15.

[Per01] Peregrine Alliance Manager, http://www.peregrine.com (as of Nov 2001).

[Pet81] J. Peterson; *Petri Net theory and Modeling of Systems*; Prentice Hall, 1981.

[PM01] *Petri Net Markup Language*; http://www.informatik.hu-berlin.de/top/pnml/.

[Rei92] W. Reisig; *A Primer in Petri Net Design*; Springer verlag, 1992.

[RN01] *RosettaNet Implementation Framework: Core Specification (RNIF 02)*; RosettaNet, July 2001, http://www.rosettanet.org.

[Sem01] *FORO Workflow Management System*; http://dis.sema.es/projects/FORO/foro.html; Sema Group, Spain, 2001.

[Sha99] J. Shanmugasundaram, K. Tufte, C. Zhang, G. He, D. DeWitt, J. Naughton; *Relational Databases for Querying XML Documents: Limitations and Opportunities*; Procs. Int. Conf. on Very Large Databases; Edinburgh, UK, 1999; pp. 302-314

[Tsi78] D. Tsichritzis, A. Klug; *The ANSI/X3/SPARC DBMS Framework*; AFIPS Press, 1978.

[UD00] *UDDI Technical White Paper*; http://www.uddi.org; 2000.

[Von00] J. Vonk, W. Derks, P. Grefen, M. Koetsier; *Cross-Organizational Transaction Support for Virtual Enterprises*; Procs. 5th Int. Conf. on Cooperative Information Systems; Eilat, Israel, 2000; pp. 323-334.

[WF01] Workflow Management Coalition; *Workflow Standard Interface 1: Process Definition Interchange Process Model*; Doc. nr WfMC TC-1016-P; 1998.

[WM01] Workflow Management Coalition; http://www.wfmc.org.

[WS01] F. Leymann; *Web Services Flow Language (WSFL 1.0)*; IBM, May 2001, http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf (as of Nov 2001).

[W301] W3C; *Web Services Description Language (WSDL) 1.1*; http://www.w3.org/TR/wsdl; 2001.

[XF01] *CrossFlow Project*; http://www.crossflow.org; CrossFlow Consortium, 2001.

[XL01] S. Thatte; *XLANG – Web Services for Business Process Design*; Microsoft, 2001; http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm (as of Nov 2001).