UNIVERSITÄT AUGSBURG
Universitätsbibliothek

OPEN ACCESS
OPUS AUGSBURG

A decentralized broker architecture for collaborative business process modelling and enactment

Bernhard Bauer, Jörg Müller, Stephan Roser

Angaben zur Veröffentlichung / Publication details:

Bauer, Bernhard, Jörg Müller, and Stephan Roser. 2007. "A decentralized broker architecture for collaborative business process modelling and enactment." Edited by Guy Doumeingts, Jörg Müller, Gérard Morel, and Bruno Vallespir. *Enterprise Interoperability : new challenges and approaches*, 115–26.

Nutzungsbedingungen / Terms of use:

licgercopyright

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:
Deutsches Urheberrecht
Weitere Informationen finden Sie unter: / For more information see:
https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/

A Decentralized Broker Architecture for Collaborative Business Process Modelling and Enactment

Bernhard Bauer[1], Jörg P. Müller[2], Stephan Roser[1,2]

[1] Programming of Distributed Systems, Institute of Computer Science, University of Augsburg, D-86135 Augsburg, [bauer|roser]@ds-lab.org
[2] Siemens AG Corporate Technology, CT IC 6, D-81730 München, joerg.p.mueller@siemens.com

Keywords: Business process modelling, model-driven architecture, cross-organizational business processes, service-oriented architecture.

1 Introduction

To enable companies to keep up with constantly evolving business relationships and cross-organizational value chains, business systems and value chains need to become more adaptive. This requires methodologies, methods, and infrastructures to support changes to business processes to be defined at the business level and propagated down to the level of information and communication technology (ICT) systems via well-defined, largely automated model transformations and refinements.

The main objective of our research is to develop architectures and tools to improve business interoperability by providing end-to-end support for the design of business processes, from the business level (users' view) down to deployed applications (ICT view) on specific platforms. The architectural approach we follow is model-driven software development (MDSD) [2], a generalization of OMG's Model-Driven Architecture paradigm (MDA™) [11].

Within the context of the ATHENA IP [1], we have extended the MDA paradigm to fit the needs of modelling Cross-Enterprise Business Processes (CBPs). The goal here is to develop executable models of cross-enterprise collaborations by applying MDSD techniques. In previous work [3], we have described a conceptual MDSD-based architecture for modelling collaborative business processes. A set of model transformations enabling semi-automated mapping of a computation-independent (CIM, business-level) description (in our case starting from a CBP model expressed using ARIS) to a platform-independent model (PIM) representation based on the PIM4SOA metamodel developed in ATHENA.

The architecture described in [3] proposed a *centralized* broker architecture for realizing business level processes into an ICT architecture at the platform-independent level. This centralized architecture was found to be useful in a sce-

nario where the collaborative process was largely designed and its execution controlled by one partner in a cross-enterprise relationship, i.e., corresponding to a star topology of a business relationship including one large and powerful player and multiple smaller players, as we can observe it today e.g. in the automotive and aerospace domains. However, it reveals limitations in flexibly supporting more symmetric business relationships, focusing on scenarios supporting collaboration between Small and Medium Enterprises, as well as Virtual Enterprise scenarios, where partners that compete in other sectors join together temporarily to provide a product or service. These scenarios will benefit from a less centralized architecture, enabling looser coupling, more modular and flexible process modification strategies, and a higher degree of autonomy of the individual partners (including better support for encapsulating enterprise-internal information).

The main contributions of this paper are twofold: firstly, we propose a decentralized broker-based architecture for controlling and enforcing CBPs in a model-driven development context; secondly, we describe a two-step transformation procedure for deriving a platform-independent model of a CBP based on the decentralized broker architecture, starting from a business-level model of the CBP using ARIS event-driven process chains (EPCs).

The paper is structured as follows: In Section 2 we summarize the technological context of this work. Section 3 then describes the decentralized broker architecture. In Section 4, we describe an instance of a model transformation process from ARIS to PIM4SOA. Section 5 discusses the contribution of our approach related to interoperability, its relationship to related work, and areas of future research.

2 Context

2.1 Business Process Modelling

It can distinguished between an internal and an external view of business processes. Depending on the viewpoint, a process is described as executable, abstract, or collaborative: An **executable process** provides a detailed *internal view* ('how') of a business process. In [6], processes, which model process flows as a set of partially ordered tasks, are called executable processes. An **abstract process** describes the external view ('what') of a business process. Each process specifies its roles, which it takes up in the collaboration with other processes. Abstract processes describe public interactions process components perform in relation to their roles in collaborations. **Collaborative processes** describe the collaboration between abstract processes. Collaborative processes use abstract processes to model the sequence of the message exchange from the viewpoint of an external observer. The collaborations between the involved parties are modelled as interaction patterns.

In order to coordinate inter-organizational workflow, Liu and Shen introduced the concept of creating views to provide abstract information about internal processes [10], derived from views as used in database systems. They extend their work for the purpose of CBPs. Chiu et al. introduce workflow views to control visibility of internal processes and to enable interoperability of e-services [5], focusing on

combining views of different partners to composite e-services (CBPs). Schulz et al. take up the concept of views, and formalize the dependencies between private processes, process views and CBPs [13].

From their work, we adapt the distinction between private processes, view process and CBPs (according to [9]): **Private Processes** are internal to a specific organization and are the types of processes that have been generally called workflow or BPM processes. They contain data that must not be revealed by default, whereas process views provide an abstraction of the private process that is sufficient to coordinate internal actions with activities of business partners [13]. **View Processes** combine private processes to an abstract level that enables companies to hide critical information from unauthorized partners. The view process connects the private process with the abstract process an organization provides to a CBP. Different views of a private process can be generated, reflecting the requirements of multiple interactions. **CBPs** define the interactions between two or more business entities. These interactions take place between the defined abstract processes and are defined as a sequence of message and/or other material input/output exchange. Using different views of the same internal processes, organisations are able to interact in a different context without changing the internal process.

2.2 PIM4SOA

A major result of the ATHENA IP [1] is a set of metamodels and tools called PIM4SOA (Platform-Independent Model for Service-Oriented Architecture, see [4]), supporting the smooth integration into Web Service Composition standards, in particular WS-BPEL [6]. The PIM4SOA metamodel has essentially two main concepts for describing services and their collaboration at a platform-independent level. **Collaborations** specify patterns of interaction between participating roles. They specify the involved roles and their responsibilities within the collaboration. S**ervice providers** provide services; they take on roles through which they participate in collaborations and realize roles in collaborations. The participation of service providers in a collaboration is modelled via *Collaboration Uses*. The bindings of a collaboration use specify which roles of the collaboration are realized by the roles of the service provider.

The communication behaviour as well as the activities, that together realize the provided services of service component, can be described by the service provider's behaviour, i.e. process. The process of a service component specifies the externally observable activities independent of the realization. The process flow defines sequencing constraints and data flow on the related activities. A task is either an internal task that is not further specified or an interaction task through a service. In the latter case the service is referenced by specifying a collaboration use path.

3 Modelling architecture for CBPs

In [3] we presented a methodical approach deriving CBPs from EPC process descriptions based on a model-driven architecture. We chose a centralized broker architecture (see Figure 1) acting as a global observer process and coordinating the partners taking part in the cross-enterprise business process. From a design

perspective the broker could be regarded as a single intermediary concept, coordinating the message exchange between the various participants as well as making decisions on the basis of data used in the cross-enterprise business process.

The application of a broker pattern has several advantages. When changing the cross-enterprise business process, i.e. its protocol description, only the broker process needs to be changed, not the multiple private processes of the participating organizations. Organizations can hide their internal processes from the other participating parties. Nevertheless they had to reveal their internal processes to the broker. We observed two interoperability-related challenges for an architecture for CBPs:

Flexibility and **modularity**: The granularity of the architectural concepts should be sufficiently fine that changes to a participant's private implementation do not necessarily result in changes of the broker. The architecture should be modular, so that participants to a CBP can change without affecting other participants or the broker.

Privacy of internal data: The modelling architecture should enable participants of CBP to preserve the privacy of their internal data, interfaces and processes. The information provided to participate in a CBP should not necessarily allow conclusion to the participant's internal realization of the functionality.

To meet these requirements we modified the conceptual architecture to a decentralized broker approach (see Figure 1). The single broker component is replaced by several view processes jointly providing the broker functionality (note the boundary in Figure 1). The view process behaviour, which is relevant to the CBP, is defined by public abstract processes. Each abstract process is realized by the executable process of the respective view process. A view process also provides internal abstract processes in order to use the private processes' functionality. The enterprise boundary in Figure 1 shows which private processes are used by a view process and vice versa.

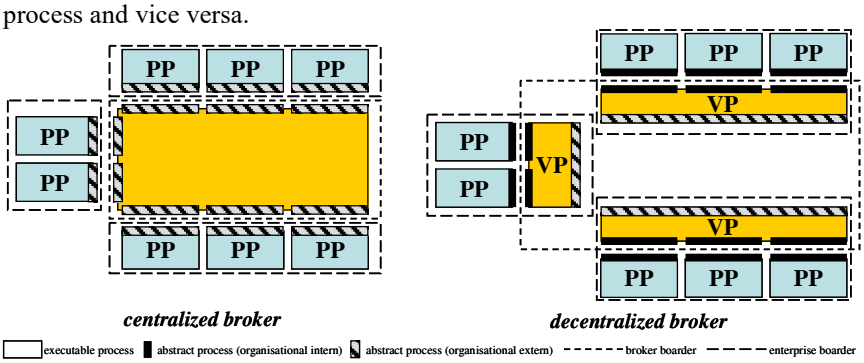

Fig. 1. Centralized vs. decentralized broker (PP=private process, VP=view process)

From a runtime view point there are two alternatives depending on whether the broker is hosted by a third party or not. In case the logical enterprise boundary in the architecture is also a physical boundary in the runtime architecture, a view process is realised and executed by the participating enterprise owing also the private processes.

Thus, the decentralized broker architecture satisfies the requirement of *flexibility and modularity* at the conceptual level. The view processes are preserved in any

runtime architecture derived from this conceptual architecture. The *privacy of internal data* depends on the realization of decentralized broker at runtime. This requirement can be met, if view processes are not hosted by a third party, but rather implemented and executed by the respective enterprises participating in the CBP.

4 A Model-Driven Design of Collaborative Business Processes

Business level models on a computational independent level of abstraction are not affected by the adoption of a decentralized broker in the conceptual architecture. In contrast to this ICT models at platform independent level of abstraction differ very well in describing centralized and decentralized brokers. Thus the transformation presented in [3] had to be adjusted to a decentralized broker architecture.

This chapter shows how from an ARIS business level description on a very high level of abstraction we derive ICT models for realizing collaborating components of a decentralized broker architecture. We are able to generate view processes, being an abstraction from more detailed private processes, and their links to the private process implementation by the means of the service-oriented PIM4SOA metamodel at platform independent level.

4.1 Starting Point: CBPs with ARIS

ARIS provides means to model private processes focusing on the isolated consideration of an organisation's (internal) processes [12]. EPC description of **private processes** can be used to model process orchestration and process choreography. For process orchestration, the concept of vertical hierarchy [8] can be applied to describe process steps and decomposition in detail. ARIS process modules represent **view processes** encapsulating internal process steps. ARIS process module chains [8] can be used for modelling c**ross-organisational business processes** describing the interactions between collaborating parties. Process module chains are EPCs extended with a swim lane concept. View processes which belong to the same role of the cross-organisational collaboration are allocated in the same swim lane. Organisational units responsible for the realization of roles in the CBP are depicted in the first column of the respective swim lane. A control flow from a process module enacted by one role, to a process module enacted by another role, represents an interaction that takes place between these roles and their realizing organizational units respectively.

4.2 Refining CBPs for PIM4SoA

In this section, the metamodel for specifying services and collaborations presented in Section 2.2 is extended by concepts for describing private processes which participate in various collaborations through their view processes.

Private processes, view processes and collaboration processes are service providers. A private process is an executable service provider who references view processes that enact its participation in external collaborations. Its behaviour is modelled by an executable process. A view process is an executable service provider whose behaviour is a process flow model that may include view tasks. A

view task is an activity that abstracts a set of activities of the realizing private processes into a single task. A view process realizes roles in a single collaboration and view tasks are visible in the collaboration. A collaboration process is an abstract service provider whose behaviour is a process flow model. The collaboration process may specify the view processes that together enact the collaboration.

We regard a view process as an *executable process* that realizes several abstract processes - one for the collaborations it participates in and the others to participate in the implicit collaborations with the private processes it supports. A view process connects the abstract process an organization provides to a CBP with realizing private processes of the organization. Nevertheless a view process is an executable process.

4.3 Generation of decentralized PIM4SOA broker description from ARIS

Now we present transformations from an ARIS CBP-description to a metamodel / domain specific model for service-oriented (business) process modelling at a platform independent level, namely PIM4SOA.

4.3.1 Example CBP

The CBP example comprises the solicitation of quotations and the choice of component suppliers by an automotive manufacturer. Three roles are involved in the cross-organizational business process:

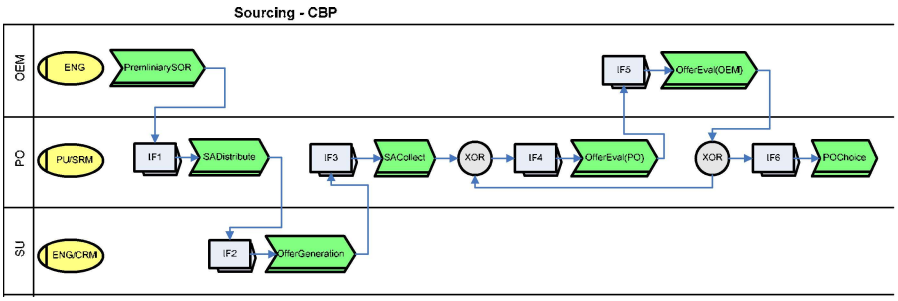

Fig. 2. Case study – process overview

OEM (Original Equipment Manufacturer): An OEM is the automotive manufacturer planning to produce a new automobile type.
PO (Purchasing Organization): Independent company or department of the OEM conduction the solicitation of quotations and the final selection of the suppliers.
SU (Supplier): The SU is a component supplier for the automotive industry aiming to place contracts with the OEM and PO.

4.3.2 Transforming the CBP

The ARIS to PIM4SOA transformation of CBPs is described by rules consisting of source and target patterns. Each rule is illustrated by the means of a sample transformation dealing with the CBP example presented in Section 4.3.

Transformation of CBP, derivation of View Processes and Collaborations
[Rule 1.1]

Src: A CBP is modelled in an EPC with row display containing a swimlane concept.
Trg: A collaboration process, which is an abstract service provider, is instantiated.

[Rule 1.2]

Src: The EPC describing the CBP is structure by swimlanes separating the process modules of the different participants.
Trg: For each swimlane a VP is instantiated and connected to the collaboration process it participates in. The name of the VP, i.e. of the service provider, is the name of the department participating in the CBP and realizing roles of the collaboration.

[Rule 1.3]

Src: In the case the source and target process module of a control flow edge lie in different swimlanes, there is a collaboration between the two roles represented by swimlanes.
Trg: a) For each pair of roles that collaborate according to the source pattern, one collaboration and the two collaborating roles are instantiated. The two roles are assigned to the collaboration. For each collaboration pair of role one and only one collaboration is instantiated.
b) Two VPs, which were derived from the swimlanes (Rule1.2), belong to the two roles participating in the collaboration. For each of the two VPs a collaboration use is instantiated referencing the collaboration.
c) Bindings are instantiated specifying for the collaboration uses, which roles of VPs (boundRole) realize are which roles of the collaboration (role).

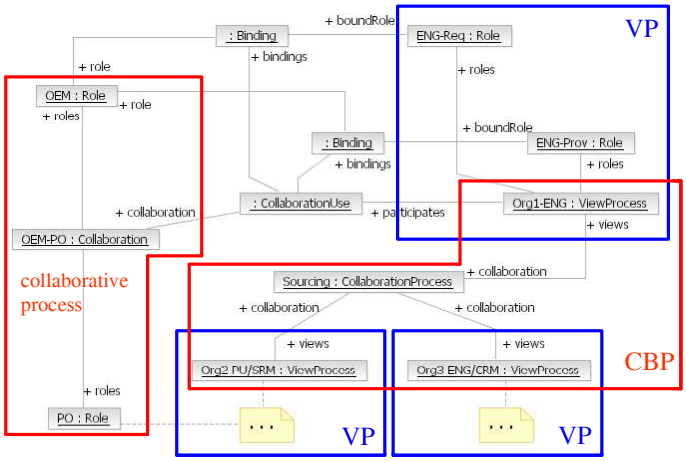

Fig. 3. Application of transformation rules 1.1-1.3

In Figure 3 we can see the target PIM4SOA model which is generated by applying the transformation Rules 1.1-1.3 to the sample ARIS model introduced in Section 04.3.1. A collaboration process is instantiated for the CBP 'Sourcing' modelled in Figure 2. The collab. process is an abstract service provider. By applying Rule 1.2 we can generate three VPs, one for each organisation that takes part in the CBP, i.e. the engineering department of the first participating organisation (Org1-ENG) and so on. An association connects the VP (`+views`) to the collaboration process (`+collaboration`). For each pair of roles that collaborate in the 'Sourcing'-CBP, one collaboration process and one role for each of the collaborating roles are instantiated for the PIM4SOA model (Rule 1.3a); in Figure 3 these are the roles *OEM* and *PO*. The participation of the organisations' departments in the collaboration is represented by collaboration uses connecting the VPs with the

collaboration (Rule 1.3b). A binding (Rule 1.3c) is used to specify with which role (`+boundRole`) a service provider realizes a role (`+role`) (OEM) in collaboration.

Considering the architecture described in Figure 1 the collaborative processes (figure 3) represents the protocol description (message exchange) between the publicly visible abstract processes. These abstract processes are realized by the VPs which are executable service providers. The CBP is an abstract service provider and groups the VPs belonging to one cross-organizational business process.

Generation of view processes behaviour/processes

[Rule 2]

Src: A CBP is modelled in an EPC with row display based on a swimlane concept.

Trg: For each VP, which has been derived from the CBP, a process is instantiated describing the VPs behaviour. The control flow of the CBP description can be taken over with a few modifications to the VP's behaviour description:

1. Each process module, belonging to the swimlane of the VP, is replaced by a view task.
2. 'Send' and 'receive' tasks are added to the control flow of the VP at the points, where the VP interacts with other VPs (i.e. where the source and the target process module of a control flow edge lie in different swimlanes).
3. The 'send' and 'receive' tasks reference the appropriate collaboration use with their collaborationUsePath. With the interactions represented by the tasks the VP takes apart in the collaboration.
4. All process modules which do not belong to the swimlane of the VP are removed from the process flow.
5. All interfaces are removed from the process flow.

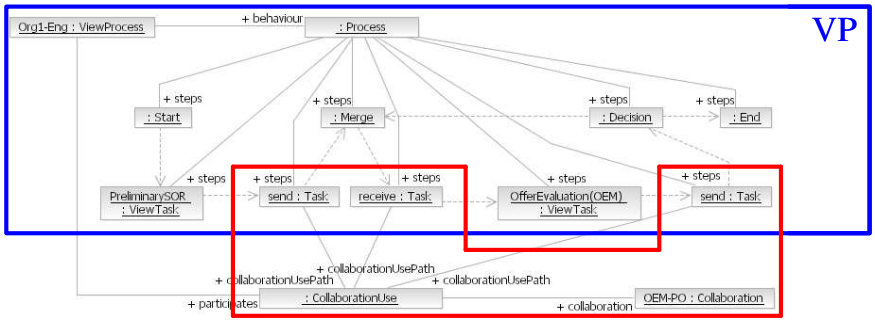

Fig. 4. Application of transformation rule 2

Figure 4 shows the generation of VPs' behaviour description at the example of the Org1-ENG VP. The behaviour of the VP, i.e. the service provider, is described by process. This process consists of steps which are derived from the ARIS-CBP according to the algorithm described in Rule 2. Two view tasks 'PreliminarySOR' and 'OfferEvaluation(OEM)' are instantiated for the corresponding process modules of the ARIS-CBP. Since the 'Org1-ENG' VP has three interactions with other VP, two times it invokes another process and one time it is invoked, two 'send' and one 'receive' tasks are added to the control flow of the VP's behaviour. Those tasks reference the respective collaboration uses over which the VP participates in collaborations. In case of the 'Org1-ENG' VP all tasks reference the same collaboration use, since the VP only participates in the 'OEM-PO'

collaboration. In Figure 4 the control flow is depicted in a simplified version for clarity reasons as arrows with dashed lines. It shows the complete description of the VP's exec. process. Those parts of the exec. process relevant to publicly visible abstract process, are bound to the respective collaborative process (see Figure 3).

Binding of private processes to the view processes

[Rule 3.1]

Src: VPs abstract from PPs and PP offer functionality to cross-organizational collaborations with the help of VPs.

Trg: A VP references all PPs it abstracts. A PP references all VP over which it offers functionality to cross-organizational collaborations.

[Rule 3.2]

Src: View tasks are used to describe the behaviour of a VP. They encapsulate tasks, which describe the more detailed behaviour of PPs.

Trg: For each view tasks all tasks are reference the view task encapsulates.

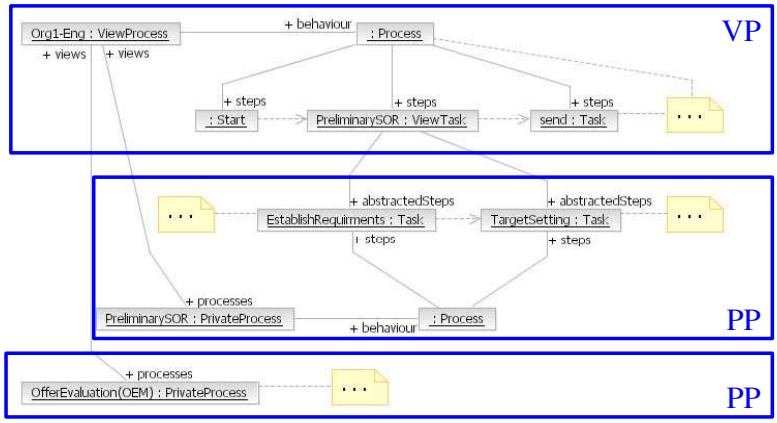

Fig. 5. Application of transformation rules 3.x

The 'Org1-ENG' VP is bound to two PPs, 'PreliminarySOR' and 'OfferEvaluation(OEM)', by applying Rule 3.1 (see Figure 5). In addition the view tasks of the VP's behaviour description reference the tasks of the PPs they abstract from. For example the 'PreliminarySOR' view tasks abstracts from the 'EstablishRequirements' and 'TargetSetting' tasks. The VP needs also connected to the PP realizing the functionality the VP provides to the CBP. Therefore the ViewTasks, which are part of the VP's internal abstract processes are connected to the PP description by the 'abstractedSteps' association.

5 Discussion

In this paper we have introduced a decentralized broker architecture as the target platform of a model-driven transformation of cross-enterprise business processes from the CIM down to the PIM-level. Based on the PIM4SOA metamodel, we identified a number of modelling constructs that allow us to describe a platform-independent architecture that can later be mapped to different ICT architectures. We have argued that in loosely coupled virtual organizations a decentralized

architecture has benefits over centralized approaches, in terms of flexibility, robustness, and autonomy. Moreover we showed how we can derive from a CIM model the necessary information to automatically create PIM-level view processes as well as the links between public processes and view processes.

Having implemented both a central and a decentral approach, the major insights we gained are as follows: First, while both approaches can be derived from a CIM description without an explicit description of the CBP, we found it important that CBPs be explicitly and carefully modelled; otherwise, model transformation results are likely to be of poor quality. Second, using the decentral broker architecture relies on the existence of a CBP model to a higher degree than the central broker architecture does: the latter can be easier derived from the process flow, whereas in the former, the appropriate grouping of processes to view processes in a decentralized broker must be specified/modelled explicitly (unless hardcoded into the transformation algorithm – which should be avoided).

Area of future work is to investigate benefits and feasibility of a decentralized peer-to-peer ICT architecture at the PIM level. In such an environment, no brokering process will be necessary; rather, every participant will have their own process description and enactment mechanisms, from which the overall CBP will emerge.

Part of the work reported in this paper has been funded by the ATHENA IP under the European grant FP6-IST-507849. It does not represent the view of E.C. nor that of other consortium members, and authors are responsible for the paper's content.

References

1. ATHENA IP project web site: www.athena-ip.org
2. ATHENA A6. Specification of a basic architecture reference model. Deliverable D.A6.1, ATHENA IP, 2005. (downloadable from [1], Public Documents).
3. Bauer, Müller, Roser: Adaptive design of cross organizational business processes using a model-driven architecture, 7. International Conference on Business Information Systems (WI 2005), Physica-Verlag, pp. 103-121, 2005.
4. Benguria, Larrucea, Elvesæter, Neple, Beardsmore, Friess: A platform-independent model for service-oriented architectures. Submitted for I-ESA'2006, 2006.
5. Chiu, Karlapalem, Li, Kafeza: Workflow view based e-contracts in a cross-org. e-services environment. Distributed and Parallel Databases 12(2-3), Dordrecht, 2002.
6. Frankel: Model Driven Architecture - Applying MDA to Enterpr. Comp., Wiley, 2003.
7. IBM, Business Process Execution Language for Web Services Version 1.1, http://www.ibm.com/developerworks/library/ws-bpel/
8. Klein, Kupsch, Scheer: Modellierung inter-organis. Prozesse mit Ereignisgesteuerten Prozessketten. In: Scheer, August-Wilhelm (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Nr. 178, Universität des Saarlandes, 2004.
9. Lippe, Greiner, Barros: A Survey on State of the Art to Facilitate Modelling of Cross-Organisational Business Processes, In Proc. of XML4BPM 2005, Karlsruhe, 2005.
10. Liu, Shen: Modeling workflows with a process-view approach. Proc. 7th Internat'l Conf. on Database Systems for Advanced Applications, Hong Kong, 2001.
11. Model-Driven Architecture homep.: Object Management Group, www.omg.org/mda.
12. Scheer: ARIS – Business Process Modeling, 3rd edition, Springer-Verlag, 2000.
13. Schulz, Orlowska: Facilitating cross-organisational workflows with a workflow view approach. Data & Knowledge Engineering 51(1), 2004.