

A Model-Driven Approach to Designing Cross-Enterprise Business Processes

Bernhard Bauer¹, Jörg P. Müller², and Stephan Roser¹

¹ Programming of Distributed Systems,
Institute of Computer Science, University of Augsburg, D-86135 Augsburg, Germany
{bauer | roser}@informatik.uni-augsburg.de

² Siemens AG Corporate Technology, Intelligent Autonomous Systems,
Otto-Hahn-Ring 6, D-81739 München, Germany
joerg.p.mueller@siemens.com

Abstract. Modeling and managing business processes that span multiple organizations involves new challenges, mainly regarding the ability to cope with change, decentralization, and the required support for interoperability. In this paper, we present an approach to modeling cross-enterprise business processes based on the model-driven architecture (MDA). Starting from the ATHENA interoperability architecture, we propose a conceptual architecture for cross-enterprise business processes. Then, we present a methodical approach towards designing cross-enterprise business processes based on a model-driven architecture. The core contribution of the paper is a set of original mappings at and across different layers of the model-driven architecture, including a mapping from ARIS to UML and the Business Process Definition Metamodel (BPDM).

1 Introduction

A key to maintain competitiveness is the ability of an enterprise to describe, standardize, and adapt the way it interacts with suppliers, partners, competitors, and customers. In the context of process orientation, enterprises today describe these procedures and interactions in terms of business processes, and invest huge efforts to describe and standardize these processes. The near future will bring an extension of these efforts towards cross-enterprise business processes. Modeling and managing business processes that span multiple organizations involves new challenges, mainly the ability to cope with change, decentralization, and the support for interoperability. Parts of the work reported on in this paper are motivated from the European Integrated Project ATHENA [1]. ATHENA addresses the vision of seamless interoperation of distributed enterprises across and beyond Europe, focusing on interoperability¹, but also covering aspects such as cross-enterprise business process modeling and architectures and platforms for business process management and enactment (see also Section 3 for more details).

¹ In the ATHENA context, interoperability is “the ability of two or more systems or components to exchange information and to use the information that has been exchanged” [4].

The focus of this paper is on business process modeling, as opposed to run-time business process management or enactment. However, given the ultimate goal to support end-to-end business processes, we need to provide a process of gradual transition from abstract conceptual descriptions of business process to concrete, executable business processes. Our approach towards this end is *model-driven development and architecture* (MDA) as promoted by the Object Management Group (OMG). Within MDA the software development process is driven by the activity of modeling the business software system. One of the major differences between MDA and traditional development processes lies in the nature of the artifacts that are created during the development process. These artifacts are formal models, i.e. models that can be understood by computers and finally be transformed into a representation that lends itself to execution which can effectively supported, e.g. by a web services ICT infrastructure.

In this paper, starting from the ATHENA interoperability architecture, we propose a conceptual architecture for cross-enterprise business processes. Then, we present a methodical approach towards designing cross-enterprise business processes based on a model-driven architecture. The core contribution of the paper is a set of original mappings at and across different layers of the model-driven architecture. In particular, the model transformations we describe are: (1) Mapping from ARIS to an UML2 business process representation (adherent to the Business Process Definition Meta-model specification [5]) at the computation-independent model (CIM); (2) mapping from an UML2 representation at the CIM level to an UML2 representation at the platform-independent model (PIM) (also adhering to BPDM).

2 Background

The next two sections outline the state-of-the-art in business-process related software architecture and IT infrastructure. They present related work and standards in business process modeling, and outline the ATHENA interoperability definition and architecture underlying our work.

The Model Driven Architecture (MDA) is a framework for software development driven by the Object Management Group (OMG). The following three models are at the core of the MDA: (1) *Computation Independent Model (CIM)*: This is the most abstract model within MDA independent of computational technology; (2) *Platform Independent Model (PIM)*: This model is defined at a high level of abstraction; it is independent of any implementation technology. It describes a software system that supports some business. Within a PIM, the system is modeled from the viewpoint of how it best supports the business; (3) *Platform Specific Model (PSM)*: In the next bin terms of the implementation constructs available in one specific implementation technology. A PIM is transformed into a PSM for each specific technology platform. Most systems today span several technologies; therefore it is common to have many PSMs with one PIM. The final step in the development is the transformation of each PSM to code. Because a PSM fits its technology rather closely, this transformation is relatively straightforward.

Today, end-to-end business processes are the focus of internal and cross-company integration. To support this diagramming approaches that can adequately represent the process modeling requirements are needed. The following are main standards addressing business process modeling:

- **ARIS** (Architecture of integrated Information Systems) (see [8]) is a framework for development and optimization of integrated information systems. In this context the ARIS concept serves as model for creating, analyzing, and evaluating business management process chains. Thus ARIS allows the description of business processes and the complexity is reduced by decomposing them into different views. ARIS is commonly used by in specifying the business view of processes.
- **EDOC** – Enterprise Distributed Object Computing (EDOC) is an OMG-supported effort to simplify the development of component based systems by means of a modelling framework, based on UML 1.4 and conforming to the MDA. The business process modelling features of EDOC are now addressed by UML2 diagrams and therefore EDOC is not considered in this paper.
- **BPMN** – The Business Process Modelling Notation (BPMN) specification, produced by BPMI provides a graphical notation for expressing business processes in a Business Process Diagram. The objective is to support process management by both technical users and business users by providing a notation that is intuitive to business users yet able to represent complex process semantics. UML2 added many of the diagramming elements from BPMN to the UML2 diagram family.
- **UML2.0** – The Unified Modeling Language is a language for visualizing, specifying, constructing and documenting software artifacts. It is a general-purpose modeling language that can be used with all major object and component methods and applied to all application domains. Its extensive use has raised numerous application and implementation issues by modelers and vendors. UML2 was produced to address many of these issues — including business process modeling [10].
- **BPDM** – Resulting from an OMG Request for Submission, the primary objective of the Business Process Definition Metamodel (BPDM) initiative is to provide an abstract model for the definition of business processes (see [5]). BPDM is specified as a UML2 profile enabling generic UML tools to both author or consume business models. As BPDM provides basic concepts from business process modeling, such as processes, tasks, rules, transactions, workers, and organizations, as first-class citizens, and additionally provides support for the modelling of collaboration, it appears a promising approach to combine the openness and genericity of UML with the expressiveness and vocabulary required for business process modeling. Mappings from a business-level model directly to runtime model like J2EE or BPEL4WS need to be defined and supported by tools. There are numerous activities towards this end, some of them carried through within the ATHENA project.

3 The ATHENA Interoperability Architecture

The ATHENA project [1] attempts to contribute towards the vision of seamless interoperation of distributed enterprises across and beyond Europe, focusing on the prob-

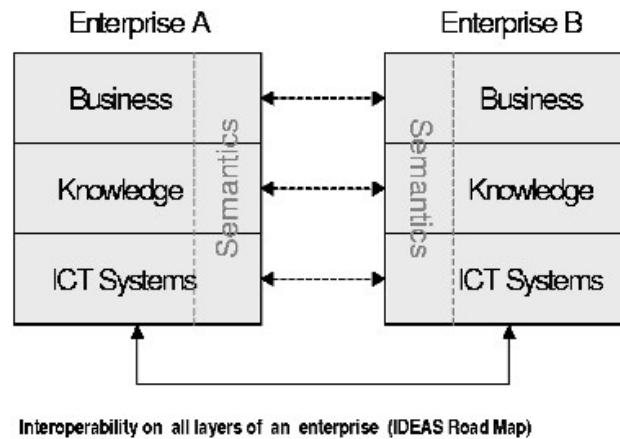


Fig. 1. The ATHENA business interoperability architecture

lem of interoperability, but also addressing aspects such as cross-enterprise business process modeling as well as architectures and platforms for business process management. Fig. 1 illustrates the conceptual architecture of ATHENA. The architecture addresses cross-organizational interoperability at three related levels: The *business level*, the *knowledge level*, and the *information and communication technology (ICT) level*. At the business level, all issues related to the organization and the operations of an enterprise are addressed, i.e. the way an enterprise is organized, how it operates to produce value, how it manages its relationships, etc.. Interoperability at business level is the organizational and operational ability of an enterprise to cooperate with other organizations. The knowledge level deals with acquiring a deep and wide knowledge of the enterprise. This includes knowledge of internal aspects such as products or the way the administration operates and controls as well as knowledge of external aspects such as partners and suppliers or laws and regulations. Furthermore, speed of changes tends to increase and the knowledge of the environment, in its widest accepted meaning, becomes more important, and sometimes even vital for the success of the business. Finally, the ICT Systems level focuses on the ICT solutions that allow an enterprise to operate, make decisions, and exchange information within and outside its boundaries. Interoperability at ICT Systems level should be seen as the ability of an enterprise's ICT systems to cooperate with those of other, external organisations.

In this paper, we focus on the business and ICT systems level. We regard the knowledge level as somewhat orthogonal to business and ICT systems level, in that different abstractions and representations of knowledge will be required at business and ICT level respectively.

In the following section, we discuss options for a conceptual modeling architecture before we describe our methodical approach and the MDA mappings in Section 5.

4 Modeling Architecture for Cross-Enterprise Business Processes

In order to enable business processes to collaborate with partners and to facilitate the composition of business processes, we apply the paradigm of service-oriented archi-

ture to business process modeling [2]. Business processes and activities are treated as components which provide services to and consume services from other business process components. Interacting business processes form a network of interconnected processes where conversations are conducted. The same process can appear in multiple solutions and can be connected to different partners in each case. The service-oriented approach to business process modeling provides a natural view to distributed business systems and lends itself to a mapping into an ICT perspective.

In describing cross-enterprise business processes, [2] suggests distinguishing between an *internal* and an *external view*.

- The *internal view* models the ‘how’ of a business process. In [3] processes, which model process flow as a set of partially ordered *tasks*, are called *executable processes*. There the flow of the processes’ interactions is described from the point of view of a single process, which coordinates participating sub-process. This kind of process composition is referred to as process *orchestration*.
- The *external view* models the ‘what’ of a business process. Each process specifies its roles but does not give any indication about its own realization. Beside the definition of interfaces it includes a description which process types can invoke the interfaces’ operations as well as which process types are invoked through the service itself. This *external view* on such business process components is described in *abstract processes*. Since *abstract processes* only provide interfaces for business process invocation, the interactions between processes are modeled by *business protocols*, specifying conversations of business processes as interaction patterns between their roles from the viewpoint of an external observer. Thereby only interactions visible to an external observer, i.e. the cross-enterprise process, are modeled. [2] refers to this as *collaborative processes*.

Business protocols can be realized in multiple ways, which differ in how the business protocols’ conversation flow is coordinated. Our architecture for modeling cross-enterprise business processes relies on the broker approach. An intermediary acts as a global observer process coordinating the partners, which take part in the cross-enterprise business process. There are several reasons for applying a broker pattern:

- Without a broker one executable process per partner has to be implemented, which coordinates the partners’ activity in the business protocol. This is problematic since control flow logic of one cross-enterprise business process is divided into different executable process. Due to the mutual exchange of messages these processes depend on one another. Changing the business protocol would result in changing multiple executable processes.
- This disadvantage doesn’t appear by implementing a broker coordinating the conversation between the communication partners. Each business protocol is realized through one of the broker’s executable processes. Moreover this approach is more convincing, since cross-enterprise business process, i.e. *collaborative processes*, are modeled as separate processes from the viewpoint of an external observer.

5 Model-Driven Business Process Design

In this section, we describe the core contributions of this paper: A methodical approach to designing business processes via model-driven architecture and two mappings that implement essential parts of the model-driven approach.

5.1 Methodical Approach

Due to the fact that different kinds of experts, like economists and computer scientists, did business process modeling from different angles, the existence of different approaches is not surprising. In addition to the differences in the models used, we can distinguish between a top-down and a bottom-up approach in the context of MDA:

- In the top-down approach business processes are first modeled in a computational independent fashion. Corporate as well as cross-enterprise business processes are modeled by process consultants with a business background. A very common modeling methodology for this kind of business process modeling is ARIS.
- Computer scientists tend to chose a more bottom-up approach to business process modeling, starting from platform specific models, which allow automated process execution (e.g. BPEL4WS, BPML, etc.). Hence models for higher level descriptions of business processes, which can be mapped to those process execution languages, like the Business Process Definition Metamodel (BPDM) are emerging. Since for most IT systems object-oriented methods have become accepted as a standard, it is obvious that these systems are modeled in UML. In addition, UML offers the advantage of openness and (to a large extent) vendor independence.

In such an environment it is important for developers of IT systems to ensure that the ARIS models of economists are consistent with the BPDM models of computer scientists. The change of modeling methods is a very crucial point in the development of an IT system, since mistakes in process modeling made here are rarely found before the system is deployed.

In this paper we introduce a mapping between ARIS and BPDM. This mapping makes it possible to deduce BPDM-Models at PIM level from ARIS-Models at CIM-level. Processes at PIM level shall be described in such a way, that they can be transformed to process execution languages on PSM level. The choice of an UML-based business process representation at the CIM and PIM levels addresses the tendency towards improving interoperability through choosing open standards for business process representation. At the same time, supporting a mapping from ARIS into BPDM takes the fact into account that ARIS is the leading modelling business process methodology in industry and tries to improve the impact of the work done in ATHENA by widening its scope, hence making it easier for enterprises to migrate their business process models into a model-driven framework. Finally, choosing BPDM as a vendor-independent representation will allow to leverage the support for modelling collaboration which will be provided in BPDM and makes our approach particularly suitable for the modelling of decentralized, cross-enterprise processes.

The mapping is performed in two stages:

- Transformation of ARIS models to object-oriented BPDM models, which are as far as possible equivalent, whereas static and dynamic structures have to be considered. The focus is on the transformation of the process-oriented event-driven process chains, which are used for the representation of dynamic aspects in ARIS.
- Generation of BPDM models at PIM level out of BPDM models at CIM level. Tools which implement the MDA approach will provide implementations to perform this task automatically.

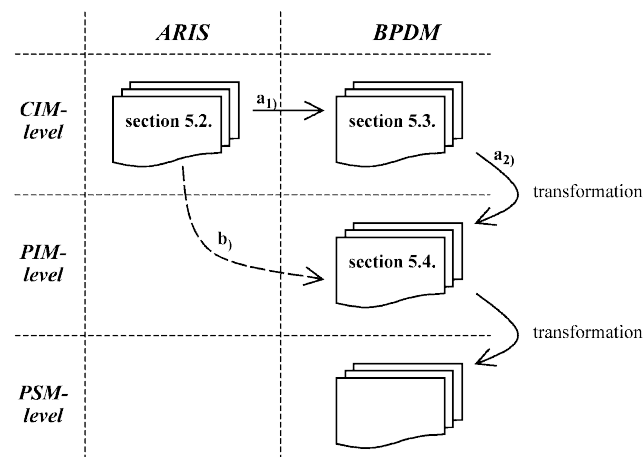


Fig. 2. Model-driven business process design approach

As illustrated in Fig. 2, there are two alternatives in developing the ARIS-BPDM mapping. In alternative a) each of the two identified tasks is carried out in a separate step. Alternative b) suggest performing the mapping, i.e. both the transformation from ARIS to BPDM and generation of PIM models from CIM models, in a single step. In the following, we shall briefly discuss advantages and disadvantages of these alternatives.

An advantage of alternative b) is that the mapping is conducted in one single step. It avoids the development and therefore the existence of a (redundant) second model at CIM level. Though it would be possible to realize the mapping in one single step, there are a number of aspects we consider problematic. A parallel and nested execution of the mapping, in reference to the mapping of ARIS to BPDM and the transformation of CIM level to PIM level, leads to a large, monolithic and poorly comprehensible mapping. Moreover, two different and insufficiently integrated modelling languages are used for describing the development of one ICT system, though integrating different models is one of the main points required by the MDA.

These disadvantages do not occur in alternative a), where the mapping is separated into two tasks as described above. Hence, the mapping becomes more transparent and easier to understand for the user. It is easier to track which model elements have their origin in ARIS diagrams and which model elements were generated through the transformation to PIM level. Furthermore, it is possible to use completely integrated models for system development. Even the existence of two models for one system at CIM level need not be a disadvantage since different models may appeal to different types of users (i.e. ARIS to business users and BPDM to users with an ICT perspective).

For the development of an ARIS-BPDM mapping we choose alternative a) and sum up the correlations and differences between various model types. ARIS models are common for describing corporations and their business processes from an economical point of view at CIM level. With BPDM models system developers describe the business processes and corporation's aspects from an object-oriented point of view at CIM level. At PIM level BPDM models describe business processes and aspects of the corporation, which are relevant for the ICT system in development, in a more detailed way. Finally on PSM level runtime models, like for example automatable business processes models, are described.

In the remainder of this section we present a mapping from ARIS models at CIM level to BPDM models at PIM level. The mapping is showed by an example. The example follows the architecture for modeling cross-enterprise business processes in Section 4. Since the focus of the mapping is the transformation from process-oriented to object-oriented descriptions of the control flow, the examples cover only models which are relevant for the description of control flows.

5.2 ARIS Model at CIM Level

Process structures can be modeled as value-added chain diagrams (VACDs) in ARIS. The control flow is modeled by event-driven process chains (EPCs) attached to the respective processes. EPCs consist of *functions* representing tasks which contribute to the corporation's objectives; *events* representing states, which are conditions under which a function can be executed or hold when a function was terminated; *process interfaces* indicating the passing of the control flow from one process to another one. Therefore a process interface at the beginning of an EPC specifies the process types by which the process can be invoked. Process interfaces at the end of an EPC specify which process types the process intends to invoke.

In the example (see figure 3) the processes of a buyer and a seller conducting cross-enterprise business processes are modeled. The *buy* process is partitioned into an *identify demand* and a *verify offer* process. The *sell* process consists of a *calculate offer* and a *coordinate offerings* process.

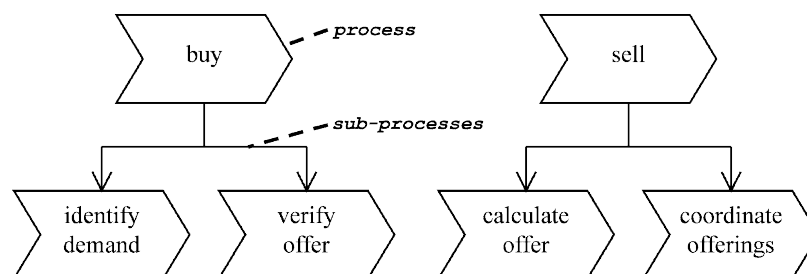


Fig. 3. Example of an ARIS value-added chain diagram

Figure 4 shows the refined EPCs for the example processes. When demand is identified by the buyer, the *identify demand* process finally invokes the seller's *calculate offer* process. This process calculates an offer and returns the result to the buyer.

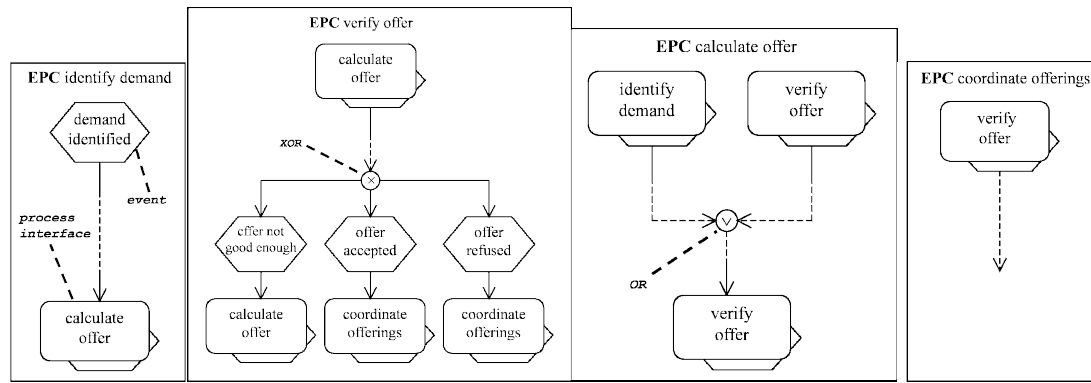


Fig. 4. Example of ARIS event-driven process chains

There the *verify offer* process decides whether the offer can be accepted or not. When the offer is accepted the process informs the seller's *coordinate offerings* process about this. In the case the offer cannot be accepted the *verify offer* process can invoke the *calculate offer* process for a new offer or inform the *coordinate offerings* process that the offer is refused.

5.3 Mapping to BPDM Model at CIM Level

Since BPDM is an extension of UML2 it is allowed to use UML2 concepts in BPDM models. In BPDM we model static aspects in class diagrams. Dynamic aspects of the collaboration between the processes of the buyer and the seller are modeled as *business protocols*. Therefore we use sequence diagrams.

Mapping rules for modeling static aspects are as follows:

- Each process from the ARIS VACD is mapped to a class with the stereotype `<<process>>` defined by BPDM (see figure 5).
- The process hierarchy from the ARIS model is realized through the *sub-process concept* specified in BPDM [5]. For modeling we use directed associations.

While information about the cross-enterprise processes in the ARIS model is contained in the EPC, i.e. through process interfaces, we use sequence diagrams to represent the business protocol interactions in UML. In figure 6 the corresponding business protocol is modeled as the *MyContractNet Protocol*.

- The processes (agents) representing buyer and seller respectively conduct a negotiation. They are modeled as *lifelines*.
- In the sequence diagram internal behavior of e.g. the seller process is not visible.

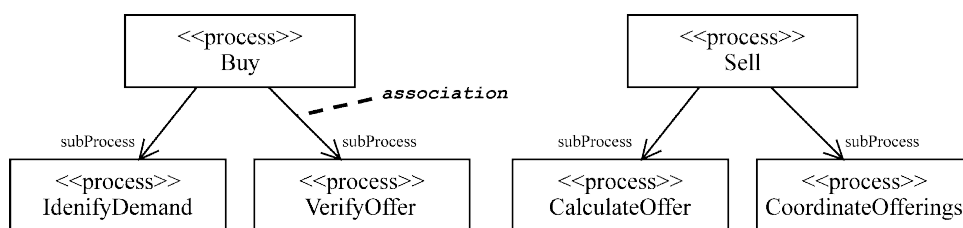


Fig. 5. BPDM – class diagram at CIM level

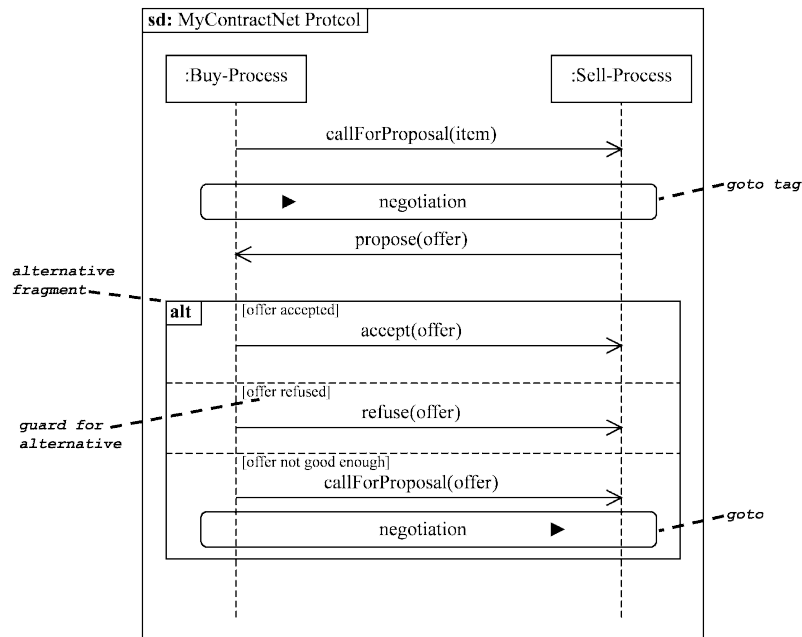


Fig. 6. BPDM – business protocol modeled in sequence diagram

- *Messages* are derived from the *process interfaces* of the EPCs in ARIS. Parameters of the messages, which are not visible in the discussed ARIS diagrams, are added to obtain a more complete model.
- Decisions in the control flow are mapped to *combined fragments*, e.g. the *alternative fragment* which models whether an offer is accepted or not.
- *Guards* are derived from the *events* in the EPCs and specify conditions for the alternatives to be taken.

5.4 BPDM Model at PIM Level

At PIM level business processes und business protocols are modeled from the IT implementation point of view. According to our modeling architecture of cross-enterprise business processes above, a broker (see figure 7) is introduced as a collaborative process coordinating the control flow of the business protocols.

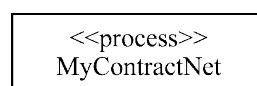


Fig. 7. BPDM – process class of the broker

For modeling the broker process's internal behavior as an executable process we use an activity diagram. Following constructs can be found in the activity diagram:

- The *control flow* identifies sequencing of activities. *Data flow* identifies the flow of data objects between activities. Data objects being in- and outputs for actions are modeled as *PINs*. The names of the PINs specify the type of the data objects exchanged.

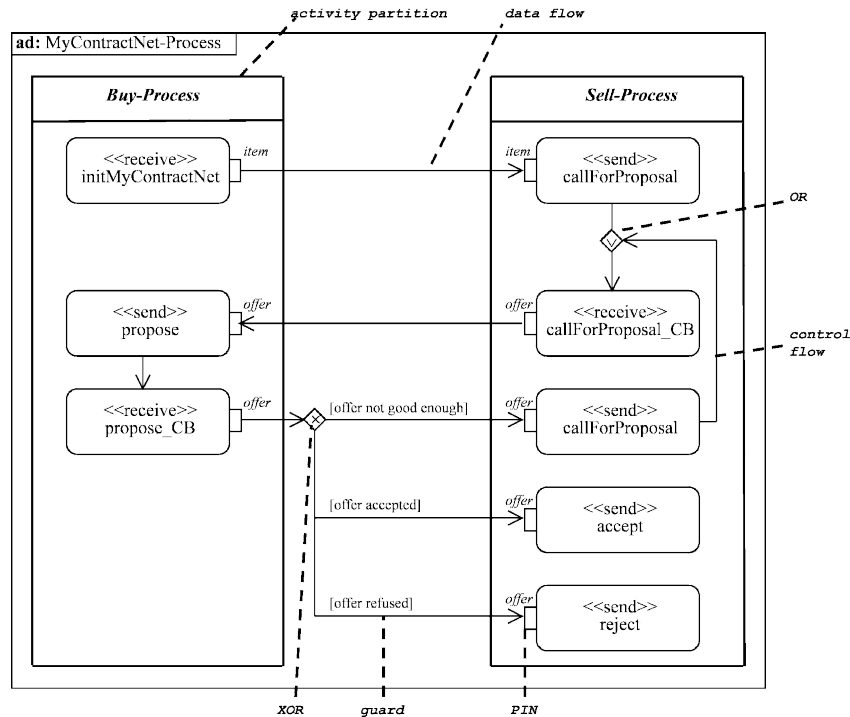


Fig. 8. BPDM – modeling cross-enterprise business process control flow in an activity diagram

- BPDM [2] describes actions stereotyped *<<send>>* or *<<receive>>* to model inter-process communication. The corresponding partner processes are identified by *activity partitions* in which these actions are modeled.
- *Decision nodes* and *merge nodes* are used to model alternatives of the control flow. Which alternatives have to be taken is specified with *guards*.

Although the activity diagram models the internal view of the broker process, i.e. the executable process, elements used to model the external view of a process are also present, e.g. the stereotyped actions are also used to describe abstract processes.

In Fig. 8, the *MyContractNet* protocol is depicted in an activity diagram as an executable process. We give a brief description of main tasks performed by transforming the sequence to an activity diagram:

- Lifelines of a sequence diagram become activity partitions in an activity diagram.
- Sequence diagram messages lead to stereotyped actions in an activity diagram. In the example the initial receive action *initMyContractNet* starts the broker's process. Due to the *callForProposal*-message of the sequence diagram the broker process sends a *callForProposal* to the sell agent's process. Since the *callForProposal*-message contains the parameter *item*, a data flow between the two actions is modeled. As we use asynchronous messaging, the *callForProposal*-message's results, i.e. an *order*, are passed to the broker's process in the next receive action *callForProposal_CB*. The broker's process sends the *propose* to the sell agent's process.
- The *alternative fragment* of a sequence diagram, which models whether an order is accepted or not, is realized as a *decision node* in the activity diagram.

6 Conclusions and Outlook

The contribution of this paper is threefold: First, we present a conceptual architecture for modeling cross-enterprise business processes based on a model-driven architecture; second, we propose a design approach suitable to MDA, and third, we provide specifications of two model transformations (mappings) to implement the design approach, thus enabling the smooth transition from an ARIS model via a computation-independent BPDM model to a platform-independent BPDM model.

Future work will complete the approach by adding mappings to selected platform-specific models including the BPEL4WS language [3]. We will also explore richer models of cross-organizational business processes, such as that proposed by [9], to address a wider range of central and decentralized collaboration architectures. A third strand of work will be to enhance adaptability of runtime process infrastructures by a Robust Planning and Execution Layer that combines dynamic service discovery and composition as well as richer transaction and compensation in business process execution. Preliminary work in this direction has been published in [7].

Acknowledgements. Part of this work reported in this paper has been funded by the ATHENA IP under the European grant FP6-IST-507849. This paper is partly funded by the E.C. through the Athena IP. 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. Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications. Europ. IP FP6-IST-507849. <http://www.athena-ip.org>
2. Frank, Gardner, Johnston: Business Process Definition Metamodel – Concepts and Overview, IBM, 2004
3. IBM, Business Process Execution Language for Web Services Version 1.1, <http://www.ibm.com/developerworks/library/ws-bpel/>
4. IDEAS. Interoperability Developments for Enterprise Application and Software–Roadmaps. European Project IST-2001-37368. <http://www.ideas-roadmaps.net>
5. Iyengar et al. Business Process Definition Metamodel. Revised Submission to BEI RFP bei/2003-01-06. 2004
6. Kleppe, Warmer, Bast. MDA Explained – The Model Driven Architecture: Practice and Promise, Addison Wesley, 2003
7. Müller, Bauer, Friese. Programming software agents as designing executable business processes: a model-driven perspective. Vol. 3067 of Lecture Notes in AI, Springer, 2004
8. Scheer. ARIS: Modellierungsmethoden, Metamodelle, Anwendungen, Springer 1998
9. Schulz, Orlowska. Architectural Issues for Cross-Organisational B2B Interactions, International Workshop on Distributed Dynamic Multiservice Architectures at ICDCS-21, IEEE Computer Society Press, 2001
10. UML Homepage. The Object Management Group. <http://www.omg.org/uml/>
11. W3C Web Services glossary. <http://www.w3.org/TR/ws-gloss/>