

The CHORCH B2Bi Approach: Performing ebBP Choreographies as Distributed BPEL Orchestrations

Andreas Schönberger
Distributed and Mobile Systems Group
Otto-Friedrich-University of Bamberg, Germany
andreas.schoenberger@uni-bamberg.de

Abstract

Applying choreography and orchestration technology has become a popular method of attacking Business-2-Business integration (B2Bi) challenges like agreement and communication among integration partners, compatibility of interacting processes and distributed computing. ebXML BPSS (ebBP) as dedicated B2Bi choreography standard and WS-BPEL as number one Web service orchestration language are particularly promising technologies. While ebBP can be used as means for agreement and communication among integration partners WS-BPEL and Web services can be used to solve distributed computing issues. The CHORCH approach applies model driven development to the ebBP-BPEL tool chain in order to further foster conformance of WS-BPEL orchestrations to ebBP choreographies, compatibility of interacting WS-BPEL processes and efficient software development cycles. This paper introduces 10 requirements for applying choreography and orchestration technology to B2Bi and shows how these are reflected in the CHORCH approach by applying three different types of ebBP modeling flavors.

Keywords: B2Bi, choreography, orchestration, ebXML BPSS, WS-BPEL

1. Introduction

Business-2-Business integration (B2Bi) is an area of enormous economic importance. Recent figures of RosettaNet, a major B2Bi community defining business document formats and exchange procedures, show that B2Bi transactions implemented using RosettaNet's Partner Interface Processes (PIPs) alone are worth billions of dollars (RosettaNet Standards Assessment 2008¹). The implementation of B2Bi processes calls for adequate engineering support. In the last few years, the application of choreography and orchestration technology emerged as important technique for tackling system integration problems. For the B2Bi domain,

1. <http://www.rosettanet.org.my/Download/2009%20ImplementationStatistics%2005.26.09.pdf>

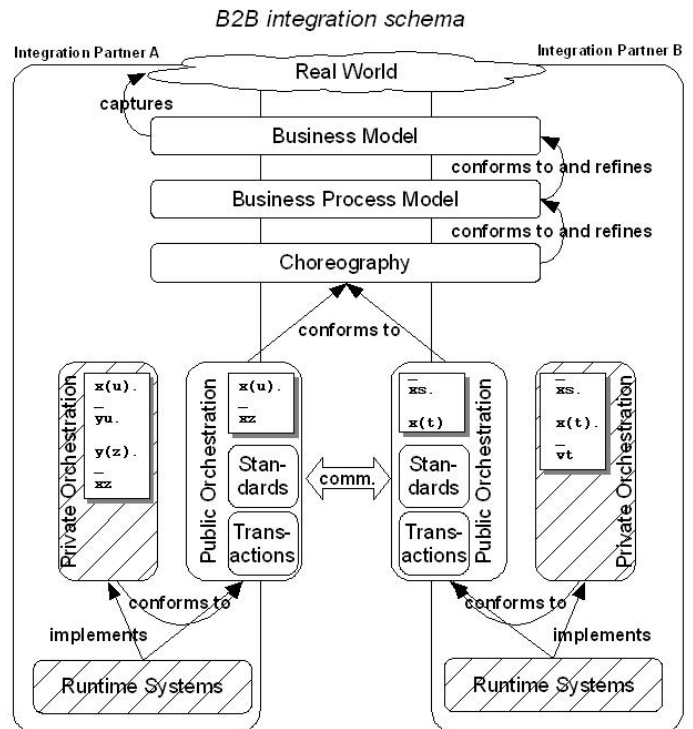


Figure 1. B2Bi schema (taken from [1])

choreography and orchestration can be put into context as defined in the B2Bi schema depicted in figure 1.

The schema shows six abstraction layers for capturing B2Bi processes from different perspectives and at different levels of detail. In an ideal world, a business model of a B2Bi scenario is first created for capturing its value proposition. Such a model then is to be refined by a business process model that identifies the type of information to be exchanged, the tasks to be performed, and valid sequences of task executions from a global perspective. A so-called choreography model then should be used for refining a business process model by adding technical execution parameters, in particular message schema definitions and B2Bi relevant Quality-of-Service (QoS) parameters such as encryption and reliability. All three models should apply a global perspective in order to be a means for agreement and communication among

personnel of integration partners. B2Bi connects information systems of distributed partners and therefore execution of local logic is an inherent B2Bi characteristic. So-called orchestration models can be used for describing the local behavior of participants using an executable specification format. While public orchestration models allow for focusing on the publicly visible message exchanges of an integration partner private orchestration models allow for the specification of the full logic for integrating publicly visible message exchanges with internal business applications. Finally, the executable orchestration models have to be deployed to runtime systems.

Up to now, the potential of choreography and orchestration technology has not fully been leveraged for implementing B2Bi scenarios. In particular, important B2Bi requirements are not reflected in current approaches that are based on ad-hoc compositions, mix up business process models and choreography models, define choreography models in a technology specific manner, or provide one single solution for arbitrary B2Bi problems.

The CHORCH approach analyzes the relation of B2Bi choreographies and orchestrations using ebXML BPSS [2] (ebBP) as dedicated B2Bi choreography standard and WS-BPEL [3] as de facto Web service orchestration standard. The core goal of CHORCH is providing the technical foundations for automatically deriving WS-BPEL orchestrations from ebBP choreographies while respecting the most important B2Bi requirements. Such an approach is in line with current industry and research focus on business process management. In a recent Delphi study, Indulska et al. [4], who interviewed business process management (BPM) vendors, practitioners and academics, found that the top BPM priorities are standardization, model-driven process execution and the value proposition of business process modeling. ebBP and WS-BPEL both are standards and therefore the first two priorities clearly are addressed while the value proposition of BPM modeling rather is a question of higher abstraction layers as depicted in figure 1. In section 2, the requirements to support when applying model-driven development (MDD) to B2Bi choreographies and orchestrations are presented. Deficiencies of existing approaches in meeting these requirements are discussed in section 3 and section 4 describes how these are reflected in the CHORCH approach. Section 5 concludes and points out directions for future work.

2. Requirements

In [5], 78 B2Bi requirements have been identified by means of an extensive literature study based on B2Bi standards, reference models and related literature. For narrowing down these requirements to CHORCH's purpose, i.e., investigating the derivation of B2Bi orchestrations from choreographies, the following assumptions are made:

- Integration partners know each other and have agreed to do business with each other for an extended period of time. This corresponds to the extended enterprise integration type of B2Bi instead of market B2Bi (cf. [6]).
- Choreography definitions are defined collaboratively and are agreed upon by integration partners before orchestration processes are used to implement the agreed-upon choreographies in a distributed manner. In [7], this type of integration is denoted *hybrid* integration and is shown to perform well.
- B2Bi processes are intended to be reused and the process structure itself does not have to be manually modified for every process instance. This corresponds to the *Straight-Through-Processing* style instead of *Case Handling* ([8]).

Moreover, requirements that target at concrete implementation projects, that are very general in nature, or that go beyond investigating the technical conditions for deriving B2Bi orchestrations from choreographies are not presented. In summary, the following list of requirements can be derived (see [5] for more detailed information):

- 1) **Usage of standards.** Choreography and orchestration standard languages are to be used.
- 2) **Language technical actor appropriateness.** Choreography and orchestration models shall be amenable to automatic processing which has several implications:
 - a) Machine-processable format. The syntax of models is to be precisely defined, e.g., using XML Schema technology.
 - b) Clear semantics. The meaning of language constructs must be precisely defined.
 - c) No deviations from standards. The use of standard tools should not be hindered by deviations from choreography or orchestration standards.
- 3) **Support for business documents.** The import for existing business document definitions as defined by business document libraries like RosettaNet or Odette² is to be supported.
- 4) **Language domain appropriateness.** Choreography and orchestration languages shall reflect the characteristics of the B2Bi domain. This includes:
 - a) Support for business transactions. The concept of business transaction should be used to abstractly define alignment of the integration partners' IT systems at the choreography level. At the orchestration level, the details of interaction are to be specified.
 - b) B2Bi Quality-of-Service. B2Bi-related Quality-of-Service (QoS) features like encryption or reliable messaging are to be supported.
 - c) Data oriented process definition. It should be possible to describe the sequence of admissible business transactions in terms of the data that has been exchanged.

2. <http://www.odette.org>

- d) Support for roles. Roles should be supported to allow for the abstract definition of tasks of integration partners that then can be mapped to concrete partner instances.
- e) State-based modeling. Changing the state of the integration partners' IT systems is the goal of B2Bi processes and state typically influences the applicability of business transactions. State therefore should adequately be represented in choreography and orchestration models.
- f) Interfacing with business applications/communication interface. B2Bi projects have to consider the integration with business applications and therefore the according interfaces shall be defined.

5) Language comprehensibility appropriateness. Choreography and orchestration languages should be easy to understand.

6) Technology independence of process model. At the choreography level, B2Bi processes should be defined in a technology-agnostic way in order to allow for different messaging technologies like Web services, ebMS [9] and AS2 [10]. This comes in two flavors:

- a) Support for multiple communication technologies in new processes.
- b) Support for multiple communication technologies when reusing existing interactions.

7) Control flow definition. Reasonable control flow support is needed.

- a) Hierarchical decomposition. Composing/decomposing complex interactions should be possible.
- b) Support for multi-party collaborations. The definition of interactions between more than two integration partners should be supported.
- c) Control flow/interaction patterns. Control flow/interaction patterns like those defined in [11] or [12] shall be supported.

8) Error handling. The handling of errors in performing B2Bi orchestrations must be defined.

9) Extensibility. Business transactions should be extensible in order to allow for new types of interactions.

10) Formalization. Formalization of choreography and orchestration models provides the foundation for automated translation, validation, simulation or semantic constraint management.

- a) Formalization of input models. The classes of valid choreography models have to be defined formally.
- b) Clear execution semantics. The semantics of executing choreographies shall be defined formally.

3. Related Work

Deriving orchestrations from choreographies has frequently been proposed and some of these approaches are also relevant for B2Bi. Most of these approaches use WS-BPEL as orchestration language. Conversely, there is no

common B2Bi choreography language. Some approaches such as [13] do not even provide a choreography language for composing business transactions. Requirements 1, 2a/b, 7a/b/c and 10a/b then barely can be met and therefore these approaches are not an option. Other approaches propose to use BPEL4Chor [14] or WS-CDL [15] but these approaches do neither provide an adequate representation for technology-agnostic business transactions nor sufficient support for specifying B2Bi-related QoS attributes at the choreography level. Finally, there are several approaches that propose visual representations for B2Bi choreographies, most notably, UMM [16], BCL [17] and Let's Dance [18]. Let's Dance and BCL are not official standards and therefore violate requirement 1. Moreover, Let's Dance does not provide sufficient support for B2Bi-related QoS specification. UMM is UN/CEFACT's official standard for visually describing B2Bi choreographies. While this fosters comprehensibility an XML-based standard such as ebBP seems to be better suited as common interchange format (requirement 2a) that may be derived from various visual languages and seems to be more suitable for further handling by analysis, transformation and execution machinery. Finally, none of the above approaches provides sufficient support for multiple messaging technologies in the same collaboration (requirements 6a/b).

A different class of related work concerns the translation of graph-like languages such as ebBP into block-oriented languages like WS-BPEL. In this regard, [19] seems to be most important in describing a technique for mapping almost arbitrary (BPMN) graphs to WS-BPEL. Support for B2Bi concepts like business transactions, B2Bi QoS and machine-processable format is limited though.

4. CHORCH

For the purpose of analyzing the derivation of B2Bi orchestrations from choreographies, ebBP is selected as choreography language and WS-BPEL as orchestration language. ebBP better suits the requirements defined in section 2 (index in parentheses) than the choreography languages considered in the last section. ebBP allows for the import of existing business document definitions (3), the messaging technology agnostic definition of business document exchanges using the concept of business transactions (4a, 6a/b), routing expressions defined on business documents (4c) as well as the specification of B2Bi relevant QoS parameters (4b). ebBP business collaborations can be used to choreograph ebBP business transactions (and other business collaborations) using control flow constructs like decisions, forks and joins (7a, partly 7c). Multiple roles (7b) can be defined on the level of business collaborations that then are mapped to the (exactly two) roles of business transactions (4d). As XML-based B2Bi standard, ebBP naturally supports requirements 1 and 2a.

At the orchestration level, the decision of using WS-BPEL as orchestration language is driven by the choice of Web services as most important messaging technology. While ebMS, AS2 or Web services are considered to be relevant for implementing business transactions (6a/b), interactions for implementing control flow between business transaction executions as well as integration with business applications is assumed to be implemented using Web services due to its interoperability benefits. WS-based business transaction implementations using Web services and WS-BPEL have been researched in [20], [21] and therefore the standards-based realization of B2Bi-relevant QoS attributes as well as sufficient means for error handling can be assumed to be realistic (requirements 1, 2a, 3, 4a/b/c/d, 8).

The discussion so far shows that important B2Bi requirements can be addressed by simply selecting the ebBP-BPEL tool chain. The real challenge rather is defining a precise semantics for ebBP which is neither formalized nor unambiguously described so far (2b, 10a/b), defining a suitable integration architecture for performing B2Bi orchestrations (4f), allowing for more than one messaging technology in the implementation of a single business collaboration (6a/b), and weighing up comprehensibility (5) and standard compliance (2c) against state-based modeling (4e), support for control flow features (7a/b/c) and extensibility (9).

The discussion of these requirements relies on the types of ebBP models to be translated into WS-BPEL. For example, multi-party collaborations are harder to design and understand than binary collaborations while explicitly representing state in B2Bi collaborations fosters comprehensibility and impairs ebBP standard compliance. The CHORCH approach therefore defines three different types of ebBP modeling for satisfying different integration scenarios.

Shared-state based ebBP modeling [22] (ebBP-ST) targets at comprehensibility and state-based modeling by explicitly modeling so-called *shared states* and limiting admissible ebBP models to business collaborations with exactly two roles and no support for parallel task executions, advanced interaction patterns or hierarchical decomposition. This leads to a state-machine like choreography definition as depicted in figure 2 which can smoothly be translated into WS-BPEL orchestrations ([22], note that visualization is not part of the work). Both partners of an ebBP-ST model concertedly leave and enter shared states (rectangles labeled $ST\langle X \rangle$ in figure 2) by performing business transactions (rounded rectangles labeled $BTA\langle X \rangle$ in figure 2) that consistently align state between integration partners. Every BTA is followed by a decision node (diamonds labeled $DEC\langle X \rangle$) that makes routing between BTAs explicit. In figure 2, arrows visualize transitions where transitions that emerge from a shared state either are triggered by the execution of a BTA or a distributed timeout (denoted [Timeout]). Transitions that emerge from BTAs are triggered upon completion of the BTA and either directly link back to the shared state the BTA

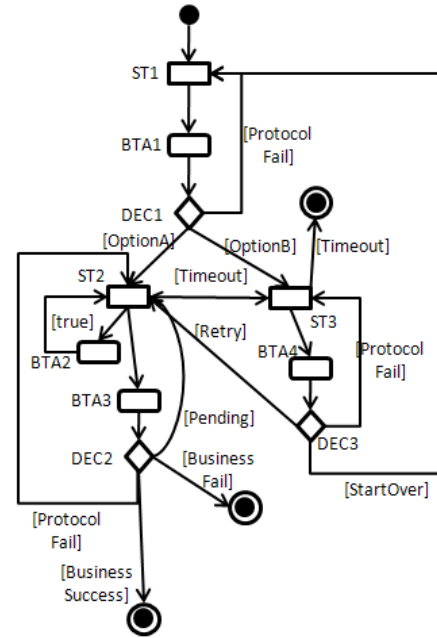


Figure 2. Valid ebBP-ST model

was triggered from or link to a decision that evaluates the BTA outcome. Finally, transitions that emerge from decision nodes immediately are triggered and represent the different outcomes of a BTA (captured as boolean guards in brackets). ebBP-ST fosters comprehensibility in a two-fold way: Shared states allow for explicitly reasoning about the applicability of performing business transactions and for reasoning about the results of business transaction executions while disallowing parallelism allows for almost arbitrary graph structures. Note that this class of ebBP models still is sufficient for capturing a large set of real-world B2Bi scenarios (cf. [23]). ebBP-ST modeling requires the introduction of shared states to the ebBP standard. Although shared states may be represented in an ebBP compliant way, modeling efficiency calls for an extension (cf. [22]). In so far, fulfillment of requirement 2c is limited to some extent.

Regular ebBP modeling (ebBP-Reg) is CHORCH's second proposed ebBP modeling flavor that is more expressive in terms of control flow features. Most notably, parallel task execution and hierarchical decomposition are allowed for. In such an environment, the concept of shared states cannot easily be supported without substantial modifications of the ebBP standard and therefore shared states are dropped. ebBP-Reg only imposes slight restrictions on standard ebBP for solving semantics issues and ensuring translatability into WS-BPEL. This concerns clarifications of how to model the necessary information for routing after having performed a business collaboration within a different collaboration and the modeling of parallel structures (and only of parallel structures) as defined in [24]. These restrictions do not contradict the explanations of the ebBP standard and therefore can be considered to be standards compliant.

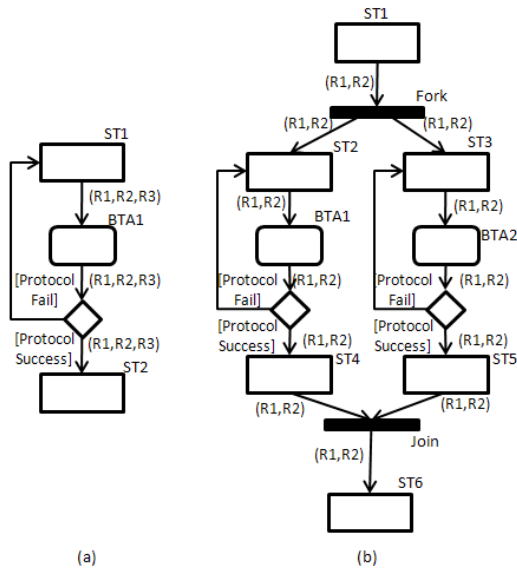


Figure 3. ebBP+ business transaction and state split
Extended ebBP modeling (ebBP+) is CHORCH’s third

proposed ebBP modeling flavor and combines advanced control flow features with the concept of shared states at the cost of dropping standards compliance. ebBP+ applies a Petri-net like modeling technique that allows multiple collaboration partners to be in different shared states (rectangles in figure 3). For controlling state alignment, the integration partners that must be in a shared state before performing a BTA (rounded rectangles in figure 3) are explicitly modeled by adding the according role names to the incoming transition of business transactions. For example, figure 3a uses the text label (R1,R2,R3) to require ebBP partner roles R1, R2 and R3 to be in shared state ST1 before BTA1 can be triggered. Similarly, the result of BTAs is also tied to participating integration partners by adding role names to outgoing transitions. By expressing preconditions and results of business transaction executions in terms of shared states, it is also possible to express interaction patterns as defined in [12] or new transaction types (requirements 7c, 9). Further, parallel structures are supported by using Fork/Join nodes (visualized as black horizontal bars labeled Fork/Join in figure 3b) to split up/combine shared states into/from sub-states that are modified by different BTAs (figure 3b). Incoming transitions of Fork/Join nodes immediately fire once the respective ebBP roles have entered the source states (denoted as text labels on the transitions). Outgoing transitions of Fork/Join nodes always immediately fire and associate the ebBP partner roles specified on the transitions with the target states.

The commonalities of CHORCH’s ebBP modeling flavors are determined by the inherent advantages of using ebBP and WS-BPEL as B2Bi choreography/orchestration language and by the need for a suitable integration architecture as

well as for support of multiple messaging technologies in a single business collaboration (addressed in prior work [25], [22]). Table 1 summarizes the support of B2Bi requirements by CHORCH’s ebBP modeling flavors.

Requirement	ebBP-ST	ebBP-Reg	ebBP+
1. Usage of standards	0	+	-
2a. Machine-processable	+	+	+
2b. Clear semantics	+	+	+
2c. No standards extensions	0	+	-
3. Business documents	+	+	+
4a. Business transactions	+	+	+
4b. B2Bi QoS	+	+	+
4c. Data orientation	+	+	+
4d. Roles	+	+	+
4e. State-based modeling	+	-	+
4f. Interfacing with back-end systems	+	+	+
5. Comprehensibility	+	-	0
6a/b. Technology independence	+	+	+
7a. Hierarchical decomposition	-	+	+
7b. Multi-party collaborations	-	-	+
7c. Control flow/interaction patterns	0	0	+
7d.* Parallelism	-	+	+
8. Error Handling	+	+	+
9. Extensibility	-	-	+
10a. Formalization of input models	+	+	+
10b. Formal execution semantics	+	+	+

Table 1. CHORCH’s ebBP modeling flavors

5. Results and Future Work

This paper defines requirements for analyzing the derivation of B2Bi orchestrations from B2Bi choreographies. Using ebBP and WS-BPEL is proposed for naturally supporting a considerable part of these requirements. Finally, the CHORCH approach is introduced that recommends different ebBP modeling flavors for accommodating different B2Bi scenarios and the commonalities and distinguishing characteristics of these modeling flavors are pointed out. Currently, CHORCH’s promise of realizing B2Bi requirements has not yet completely been fulfilled. A distributed integration architecture, support for different messaging technologies within the same business collaboration, support of B2Bi QoS properties during ebBP to WS-BPEL translations as well as a concise execution model for ebBP BusinessTransactions have been defined ([25], [20], [21]). ebBP-ST modeling has been described in [22] and the

formalization of valid ebBP-ST models and a formal operational ebBP-ST execution semantics have been defined. Immediate next steps include designing ebBP+ in detail and formally describing valid ebBP-Reg/ebBP+ models and according execution semantics. The detailed feature set of ebBP+ will be based on the analysis of control flow and service interaction patterns as well as use cases taken from B2Bi reference processes like the Northern European Subset processes³ and uses cases provided by the RosettaNet community. The validity of the formal execution semantics and WS-BPEL implementation of ebBP-Reg and ebBP+ will be based on mutually checking the formal execution semantics against the prototypical WS-BPEL implementation. Future work in the long term comprises providing visual representations of CHORCH's ebBP modeling flavors and improving analysis, simulation and semantic constraint management support.

References

- [1] A. Schönberger and G. Wirtz, "Taxonomy on consistency requirements in the business process integration context," in *2008 Conf. on Software Engineering and Knowledge Engineering*.
- [2] OASIS, *ebXML Business Process Specification Schema Technical Specification*, 2nd ed., OASIS, December 2006.
- [3] —, *Web Services Business Process Execution Language*, 2nd ed., April 2007.
- [4] M. Indulska, J. Recker, M. Rosemann, and P. Green, "Business process modeling: Current issues and future challenges," in *Proc. of the 21st Int. Conf. on Advanced Information Systems Engineering 2009 (CAiSE)*.
- [5] A. Schönberger, C. Wilms, and G. Wirtz, "A Requirements Analysis of Business-To-Business Integration," Universität Bamberg, to appear in *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik 83*, 12 2009.
- [6] F. Goethals, J. Vandenbulcke, W. Lemahieu, M. Snoeck, and B. Cumps, "Two basic types of business-to-business integration," *International Journal of E-Business Research (IJEBR)*, vol. 1, no. 1, pp. 1–15, 2005.
- [7] C. Schroth, T. Janner, and V. Hoyer, "Strategies for cross-organizational service composition," in *Proc. of the 2008 Int. MCETECH Conf. on e-Technologies*.
- [8] W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske, "Business process management: A survey," in *Business Process Management*, ser. LNCS, vol. 2678. Springer, 2003.
- [9] OASIS, *ebXML Messaging Services Version 3.0: Part 1, Core Features*, OASIS, October 2007.
- [10] D. Moberg and R. Drummond, *MIME-Based Secure Peer-to-Peer Business Data Interchange Using HTTP, Applicability Statement 2 (AS2)*, IETF, July 2005.
- [11] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros, "Workflow patterns," *Distrib. Parallel Databases*, vol. 14, no. 1, pp. 5–51, 2003.
- [12] A. Barros, M. Dumas, and A. H. M. T. Hofstede, "Service interaction patterns," in *Proc. of the 3rd Int. Conf. on Business Process Management 2005 (BPM)*.
- [13] R. Khalaf, "From RosettaNet PIPs to BPEL processes: A three level approach for business protocols," *Data & Knowledge Engineering*, vol. 61, no. 1, pp. 23–38, 2007.
- [14] G. Decker, O. Kopp, F. Leymann, and M. Weske, "Interacting services: From specification to execution," *Data & Knowledge Engineering*, vol. 68, no. 10, pp. 946 – 972, 2009.
- [15] I. Weber, J. Haller, and J. A. Mülle, "Automated derivation of executable business processes from choreographies in virtual organizations," in *Multikonferenz Wirtschaftsinformatik 2006*.
- [16] B. Hofreiter and C. Huemer, "A model-driven top-down approach to inter-organizational systems: From global choreography models to executable BPEL," in *Joint IEEE Conf. on E-Commerce Technology (CEC'08) and Enterprise Computing, E-Commerce, and E-Services (EEE'08)*.
- [17] M. Zapletal, T. Motal, and H. Werthner, "The business choreography language (BCL) - a domain-specific language for global choreographies," in *Proc. of IEEE 2009 World Congress on Services (SERVICES PART II)*.
- [18] J. Zaha, M. Dumas, A. ter Hofstede, A. Barros, and G. Decker, "Bridging global and local models of service-oriented systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 38, no. 3, pp. 302 –318, 2008.
- [19] C. Ouyang, M. Dumas, A. H. M. ter Hofstede, and W. M. P. van der Aalst, "From BPMN process models to BPEL web services," in *Proc. 2006 IEEE Int. Conf. on Web Services*.
- [20] A. Schönberger et al., "QoS-enabled business-to-business integration using ebBP to WS-BPEL translations," in *Proc. of the IEEE SCC 2009 Int. Conf. on Services Computing*.
- [21] A. Schönberger, G. Wirtz, C. Huemer, and M. Zapletal, "A composable, QoS-aware and web services-based execution model for ebXML BPSS BusinessTransactions," in *Proc. of IEEE 2010 World Congress on Services (SERVICES 2010)*.
- [22] C. Pflügler, A. Schönberger, and G. Wirtz, "Introducing partner shared states into ebBP to WS-BPEL translations," in *Proc. of 11th Int. Conf. on Information Integration and Web-based Applications & Services 2009*.
- [23] H. A. Reijers and W. M. van der Aalst, "The effectiveness of workflow management systems: Predictions and lessons learned," *International Journal of Information Management*, vol. 25, no. 5, pp. 458 – 472, 2005.
- [24] B. Kiepuszewski, A. H. M. ter Hofstede, and C. Bussler, "On structured workflow modelling," in *Proc. of the 12th Int. Conf. on Advanced Information Systems Engineering (CAiSE 2000)*.
- [25] A. Schönberger and G. Wirtz, "Using variable communication technologies for realizing business collaborations," in *Proc. of IEEE 2009 World Congress on Services (SERVICES PART II)*.

3. <http://www.nesubl.eu>