Taxonomy on Consistency Requirements in the Business Process Integration Context

Andreas Schönberger and Guido Wirtz

Distributed and Mobile Systems Group, University of Bamberg
Feldkirchenstr. 21, 96052 Bamberg, Germany
E-mail: {andreas.schoenberger | guido.wirtz}@uni-bamberg.de

Abstract

Consistency is a general goal in software development. The development of B2B integration (B2Bi) software has extended consistency requirements (CR) because B2Bi projects require personnel from different organizations to agree about the what and how of integrations and these projects require distributed computing as central IT infrastructure frequently is not available. The rise of numerous approaches targeting at consistency in the B2B area are evidence of these enhanced CRs. This paper refines the Open-edi business transaction model into a B2Bi schema by analyzing abstraction levels, development phases and distribution aspects of B2Bis. A taxonomy of CRs is then derived accordingly. Thus, this paper underlines the extended importance of consistency in B2Bis, furnishes a criterion for choosing B2Bi methods, helps classifying consistency support in B2B approaches and gives a starting point for finding consistency checking approaches.

Keywords: consistency, compatibility, business process integration, business collaboration, SOA

1. Introduction

Consistency is a predominant requirement in software development. This holds true for the early design phases of software development where methods like model checking (cf. [1]) are applied, for the transition between development phases (cf. [2]) and particularly for ensuring operational consistency using transactions in databases or transaction monitors.

The B2Bi domain even has extended CRs. First, this becomes clear by looking at the special organizational and technical conditions in the B2Bi domain. Organizationally speaking, people from different organizations with possibly different background and vocabulary have to agree about what goals to achieve in an integration and have to develop the necessary interaction protocols therefore. Technically speaking, central technical infrastructure is frequently not available or prohibited by business politics so that truly distributed computing is needed. Second, in order to provide consistent change of the common business state of integration partners, standards for realizing transactions in the Web Service domain are being developed, e.g.,Web Services Atomic Transaction (WS-AtomicTransaction, [3]) and Web Services Business Activity (WS-BusinessActivity, [4]) which are both constituent parts of the OASIS Open Web Services Transaction specification[1]. Taking into account that Web Services are an important technique for implementing B2Bis the development of these standards also bears witness to the special consistency needs of B2Bi.

Therefore, in this paper, we develop a B2Bi schema by analyzing the abstraction levels of B2Bi projects, its development phases and the purpose of relevant B2Bi standards. We then identify CRs between the components of this schema and thus derive a taxonomy of CRs. Note, that this paper does not define a new notion of consistency. For the purpose of identifying CRs the following, rather general, standard definition is used instead so as to capture a large amount of CRs: "**consistency.** *The degree of uniformity, standardization, and freedom from contradiction among the documents or parts of a system or component.*" [5].

Finally, after having discussed related work, we conclude the paper and point out directions for future work.

2. A B2B integration schema

The analysis of CRs in B2Bis needs a conceptualization of the domain in order to define between which concepts consistency is required. The development of our B2Bi schema therefore considers three important aspects which lay behind the need for consistency.

- *Abstraction levels.* B2Bis can be viewed on several abstraction levels where these abstraction levels should be consistent with each other.
- *Development phases.* B2Bis are developed according to some Software Engineering Process (SWE) of choice which consists of several phases. The artifacts produced during these phases should be consistent with each other.

[1] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-tx

- *Organizational distribution.* Multiple parties are interacting in B2Bis. These parties usually have their own IT infrastructure and IT policies. So the techniques and tools used for implementing B2Bis have to consider interoperability between the partners and transactions crossing enterprise boundaries.

Clearly, these three aspects overlap, but this distinction is useful for identifying gaps in the derivation of the schema according to only one aspect.

The derivation of our B2Bi schema starts out by considering the Open-edi reference model [6] which looks at business transactions on two abstraction levels. The so-called *Business Operational View* (BOV) covers the business aspects of business transactions while the so-called *Functional Service View* (FSV) covers the information technology aspects of business transactions [6]. This model, used as a B2Bi schema for identifying CRs, already furnishes a basic CR of the FSV being consistent with the BOV but this obviously is far too general for a CRs taxonomy of B2Bis.

The Open-edi reference model of business transactions is further refined by [7] for the purpose of classifying current B2B methodologies and technologies. This refinement splits up the BOV into *business models* {A} and *business process models* {B} while the FSV is split up into *deployment artifacts* {C} and *software environments* {D} (see figure 1 taken from [7]).

According to [7], business models describe the exchange of values between partners on an abstract level whereas business process models detail the relationships between the partners by specifying the flow of information and type of interaction. We agree with the authors in this point and also keep these two levels for our B2Bi schema. Further, [7] paraphrase deployment artifacts as machine-processable descriptions of business transactions and software environments as concrete implementations of information systems. While this distinction may be sufficient for surveying B2B methodologies and technologies we refine the schema of [7] in two points.

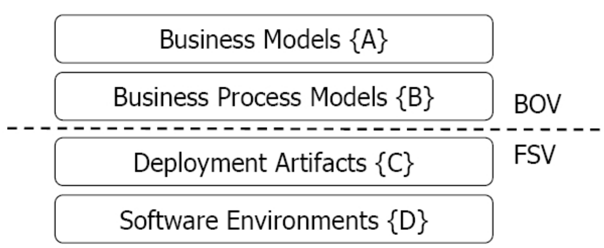

Figure 1. Classification schema based on refinements of the Open-edi reference model [7]

First we claim that, for a CRs taxonomy, the deployment artifacts level should be separated in a so-called *choreography* and a so-called *orchestration* layer. This finer distinction is necessary for respecting the distribution of the collaborating partners adequately, so that the overall message exchanges can be specified on the choreography level while the message exchanges of a single partner can be specified on the orchestration level. Apart from this argument, the development of choreography standards like WS-CDL [8] and ebxml BP (formerly known as BPSS, [9]) as well as orchestration standards like WS-BPEL [10] evidence the need for this distinction. Moreover, the orchestration level should be split up into so-called *public processes* and so-called *private processes*. This separation pays tribute to the obligation of an integration partner to obey a particular externally observable message sequence (public process) and, at the same time, integrate this public process with its (preexisting) backend systems (private process). One could argue, that if the participation of multiple integration partners leads to the dichotomy of choreography and orchestration, then the business process models would also have to be divided into *global* and *local* business process models. We claim that, from a B2Bi point of view, the core task of business process models is providing a means of communication for agreement of how to achieve business goals and we also developed a modeling approach for this task [11, 12, 13]. Nonetheless, B2Bi process models may be enhanced by local process models when doing local optimizations which then would introduce new CRs not discussed here.

Second, we refine the concept of software environments in [7] as we do not simply consider these to be implementations of information systems but to be the source for tracing consistency between actual process executions and process specifications. Hence we rename the software environments level as *runtime systems* furnishing the raw data for checking conformance of process executions with process specifications.

Although the B2Bi schema developed so far already lays the foundation for finding very important CRs, we enhance it by the following findings.

Looking at software development processes in general and in particular at the system/software requirements engineering phase, the lack of the *real world* in the schema is apparent.

Keeping the distribution of integration partners in mind, it is also clear that not only the message exchanges of the integration partners matter but also a way for synchronizing their local views on the global state of the B2Bi is essential. Thus, a way for implementing distributed transactions has to be found. A proof for this necessity is the development of standards like WS-AtomicTransaction [3] and WS-BusinessActivity [4]. Apart from distributed transactions, the integration partners have to agree about and provide for Quality-of-Service (QoS) aspects of their collaboration. Regarding the provision of QoS aspects, the integration partners either have the option to use the same integration frameworks or to use interoperability standards for ensuring QoS aspects like WS-Reliability [14] or WS-Security [15]. Thus, the application and the interoperability of such standards is another source for the identification of CRs.

The results of this discussion are summarized by figure 2 which does not only visualize the B2Bi schema developed but also identifies relevant CRs which are detailed in the next section.

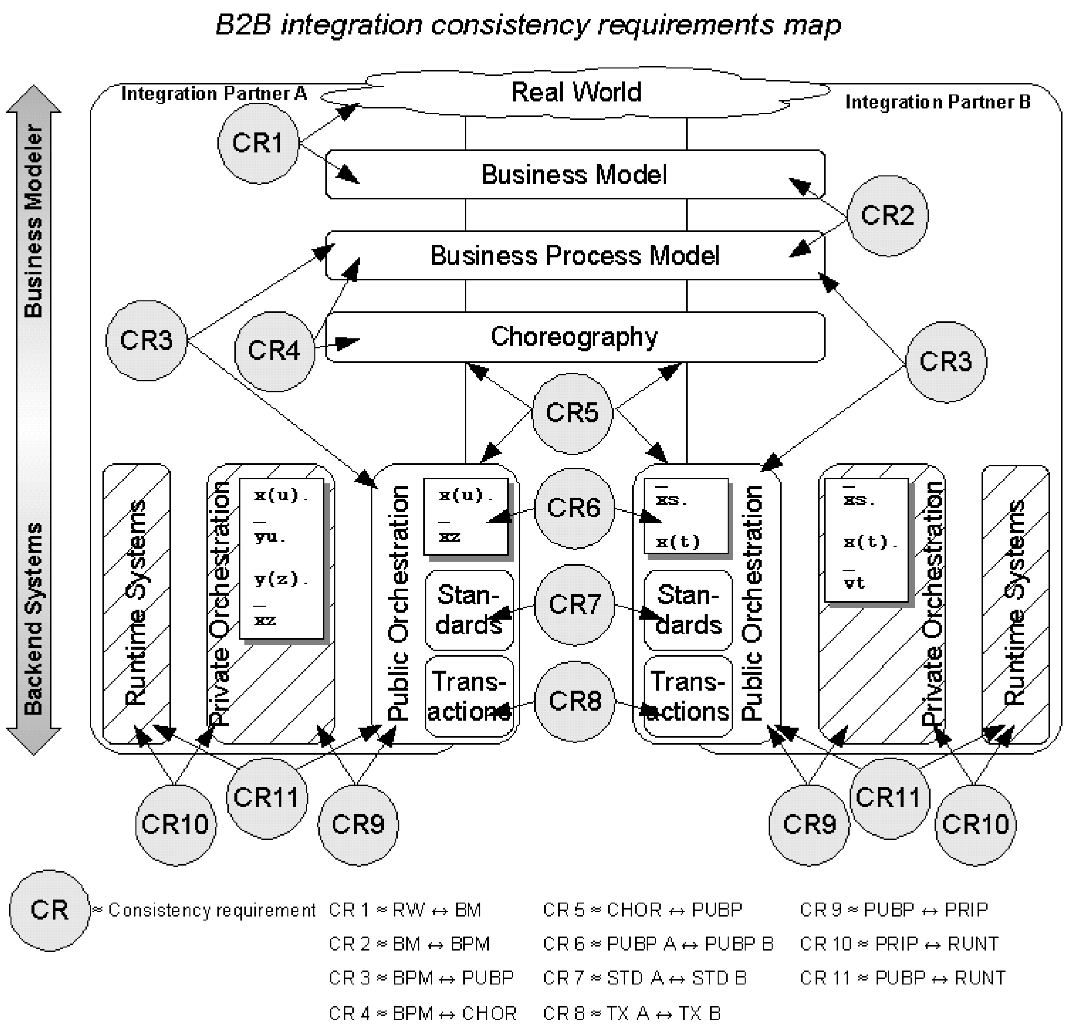

Figure 2. Consistency requirements

3. Consistency requirements

The identification of CRs using the given B2Bi schema is achieved by analyzing the relations between the components of the schema where each component contains a set of models for representing the B2Bi on a particular abstraction level. CRs between two schema components are actually CRs between the models of different components and usually emerge if these models are produced in subsequent phases of a particular software engineering process (SWE) used for implementing the B2Bi. Clearly using this way of identifying CRs would be heavily dependent on particular SWEs but it helps at least eliminating obscure CRs that would emerge from no reasonable SWE, e.g., no reasonable SWE would propose the definition of WS-BPEL [10] processes as the first step in implementing a B2Bi. Further, following the phases of a SWE, the models become more and more detailed. These additional informations cannot be completely derived from more abstract models because the more concrete models wouldn't present any further information otherwise. Theoretically this would lead to a further CR between any schema component and the real world but this theoretical requirement is neglected for practicability reasons. Finally, there are also CRs between the models of a single level, intra-model CRs (e.g. syntactic conformance to modeling languages used) and *evolution consistency* [26] requirements but these requirements are neglected here because they are a different area of concern. Applying this approach for identifying CRs leads to the set of CRs depicted in figure 2. The naming of the CRs is derived from the schema components that give rise to those CRs. This naming convention pays tribute to the fact that we are talking about consistency *requirements* and not about specialized definitions of consistency like *process inheritance* or *compatibility* [2] although these notions clearly may be related to our CRs. For each of these CRs we were able to find at least one approach that supports it which constitutes empirical evidence for the existence of the respective CR. Moreover, the solutions of the approaches under study could all be mapped to our CRs which may be a hint that there are not too many CRs missing in our taxonomy. As a detailed discussion of the approaches found must be omitted due to space limitations we provide a classification in table 1 that relates the approaches (Appr.) to the CRs of our taxonomy. Note that CR 10 and CR 11 are merged to one column and that we have included

Appr.	Consistency requirement										A/C
	RW $\leftrightarrow$ BM	BM $\leftrightarrow$ BPM	BPM $\leftrightarrow$ PUBP	BPM $\leftrightarrow$ CHOR	CHOR $\leftrightarrow$ PUBP	PUBP A $\leftrightarrow$ PUBP B	STD A $\leftrightarrow$ STD B	TX A $\leftrightarrow$ TX B	PUBP $\leftrightarrow$ PRIP	SPEC $\leftrightarrow$ RUNT	
[16]	+	-	-	-	-	-	-	-	-	-	A
[17]	-	+	-	-	-	-	-	-	-	-	C
[11, 13]	-	+	+	-	-	-	-	-	-	-	A/C
[18]	-	-	+	-	+	+	+	-	+	+	A
[19]	-	-	-	+	-	-	-	-	-	-	C
[20]	-	-	-	+	+	-	-	-	-	-	A/C
[21]	-	-	-	-	+	-	-	-	-	-	A
[22]	-	-	-	-	-	+	-	-	+	-	A
[23]	-	-	-	-	-	-	+	-	-	-	A
[3, 4]	-	-	-	-	-	-	-	+	-	-	C
[24]	-	-	-	-	-	-	-	-	+	-	A
[25]	-	-	-	-	-	-	-	-	-	+	A

Table 1. Survey of approaches targeting at consistency and CRs supported

an extra column that describes whether a particular approach supports consistency by means of analyzing models (A) or by constructing models (C) or both (A/C). In the following, we discuss the CRs of our taxonomy in more detail:

CR 1: Real World (RW) $\leftrightarrow$ Business Model (BM). This requirement is sometimes overlooked because the BM may not be part of the regular SWE artifacts produced during a B2Bi or the BM is quite nontechnical. If modeled, the BM cannot be checked for consistency with the RW automatically. However the BM itself is at least amenable to automatic analysis and thus inconsistencies within the BM may reveal inconsistencies between the RW and the BM.

CR 2: Business Model (BM) $\leftrightarrow$ Business Process Model (BPM). The task of the BPM is to specify how to achieve the business goals defined in the BM by defining the types of information to be exchanged, the flow of information and organizational aspects. CR 2 demands, that these specifications ensure the business goals or at least don't contradict them. One could argue that there should also be a CR between RW and BPM in case there's no BM. We do not reject this argument but leave out this CR for practical reasons.

CR 3: Business Process Model (BPM) $\leftrightarrow$ Public Processes (PUBP). While the BPM is a *common* model the integration partners have to agree upon, the PUBP is the definition of the communication tasks of each partner. If BPM is transformed into PUBP directly, CR 3 demands that the PUBP strictly conform to the BPM. This conformance is achievable with respect to control flow but it is also necessary to specify Quality-of-Service (QoS) attributes of the PUBP which frequently lack in BPM, e.g., *Reliable Communication* or *Security*. These QoS attributes can be more easily checked for conformance if BPMs are not directly transformed into PUBP but first in CHOR and afterwards in PUBP.

CR 4: Business Process Model (BPM) $\leftrightarrow$ Choreography (CHOR). The differences between BPMs and CHOR standards are sometimes fluent but it helps to think of BPM as a model that serves as communication means between business analysts while CHOR is a detailed technical communication specification intended to be processed by machines. The main claim of CR 4 is the conformance of information types and control flow in CHOR to the same aspects in BPM while the aforementioned QoS aspects frequently have to be introduced in CHOR.

CR 5: Choreography (CHOR) $\leftrightarrow$ Public Processes (PUBP). It is feasible to generate PUBP from a CHOR specification to a large extent. But taking into account that the results of this transformation are not unique, the demand of CR 5 for conformance of information types, control flow and QoS aspects should not be neglected, in particular for bottom-up approaches.

CR 6: Public Process of partner A (PUBP A) $\leftrightarrow$ Public Process of partner B (PUBP B). CR 6 defines the first of three CRs between arbitrary integration partners who are simply referred to as *partner A* and *partner B* for the sake of practicability. CR 6 particularly refers to the compatibility (as a form of consistency) between the observable communication of the integration partners. Note, that compatibility checks for particular properties by analyzing the PUBP may be replaced by checking the properties for the CHOR and then proving that the transformation of CHOR into PUBP is preserving these properties.

CR 7: Standards used by partner A (STD A) $\leftrightarrow$ Standards used by partner B (STD B). CR 7 takes into account that the partners of B2Bis frequently are independent from each other and thus may have heterogeneous IT systems. In order to provide for the correct implementation of QoS attributes like *Reliable Messaging* or *Security* the integration partners then either have to use the same integration frameworks with the same configurations or they have to agree on the application of QoS interoperability standards like [14] or WS-Security [15]. The former approach leads to tight coupling between IT systems and is more and more unacceptable nowadays.

CR 8: Transactional data concerned by partner A (TX A) $\leftrightarrow$ Transactional data concerned by partner B (TX B). CR 8 alludes to the common state of a business collaboration that has to be synchronized among the integration partners, e.g.,whether an order has been accepted or not or if a bill has been delivered or not. Standards like [3] and [4] have been developed to meet this requirement. In order to decide if an information item is to be synchronized using these standards it has to be decided if it belongs to the common business state.

CR 9: Public Processes (PUBP) ↔ Private Processes (PRIP). Frequently PRIP are just a refinement of PUBP that couple the observable communication of an integration partner to its backend systems. CR 9 claims that these refinements may not change the observable communication as otherwise the interaction protocol of the collaborating partners would be broken. The analysis of conformance of PRIP to PUBP is especially useful in bottom-up approaches where preexisting private processes may be wrapped to conform to predefined public processes.

CR 10: Private Processes (PRIP) ↔ Runtime Systems (RUNT). CR 10 claims that a process instance of an integration partner in execution must conform to its specification. This CR is usually met by installing monitoring systems for RUNT that furnish sufficient information. Note, that if an executed process conforms to its private process specification and CR 9 is sufficiently accounted for, then CR 11 is met as well.

CR 11: Public Processes (PUBP) ↔ Runtime Systems (RUNT). Although CR 11 may be substituted by CR 9 and CR 10, CR 11 is stipulated because it may be far easier to check because of the possibly reduced state space of public processes compared to private processes. From the B2Bi point of view both options ensure that a process in execution does not break the interaction protocol of the integration partners so just addressing CR 11 and ignoring CR 10 may be admissible as well.

The core contribution of this taxonomy is the identification of high level CRs that emerge during B2Bi projects. During such a project, a particular CR has to be further investigated with respect to an adequate definition of consistency, evaluation of suitable modeling methods and analysis tools as well as organizational implications. Finally, we claim that the choice of B2Bi frameworks should consider the support of the CRs identified.

4. Related Work

Clearly, consistency always played an important role in software development [2, 27, 28, 29] but we put our focus particularly on the B2Bi domain. In that domain, there is a lot of literature that discusses methods for checking some kind of consistency which frequently apply concepts like *process inheritance* or *process compatibility*. But work about CRs in the B2Bi domain is very scarce. Greenfield et al. [30] discuss *Consistency for Web Services Applications*. Their work is different from ours in so far as they discuss in detail for Web Services what we identified as CR 6 (PUBP A ↔ PUBP B) and CR 8 (TX A ↔ TX B). Decker et al. [31] describe compatibility and consistency notions in the B2Bi domain. They define consistency between public and private processes (CR 9) with respect to compatibility between public processes of interacting parties (CR 6, CR 5). In so far they also focus on parts of our taxonomy. To our knowledge we are the first to derive a detailed CRs taxonomy for the B2Bi domain regarding several abstraction levels.

Apart from that, there are several papers that discuss the comparison between various business process reference models or business process modeling methodologies like [32] or [33]. Our taxonomy is not suitable for performing such comparisons but we claim that consistency should be an important criterion in these comparisons and that our taxonomy is useful for classifying and comparing business process reference models and business process modeling methodologies with respect to support for CRs.

Finally, there is extensive work on *(in)consistency management* [27, 29] that describes how and when to enforce consistency and how to react to inconsistencies. Spanoudakis and Zisman [27] propose a process that consists of *detecting overlaps in models*, *detecting, diagnosing, handling and tracking inconsistencies*, as well as *specifiying and applying a management policy for inconsistencies*. Although consistency management is different from the work presented here the CRs identified can help in deciding where to apply processes like the one described by [27].

5. Conclusions and Future Work

In this paper we systematically derived a taxonomy of CRs using a B2B schema. The various areas that call for consistency checking methods underpin the importance of consistency in the B2Bi domain and especially necessitate the consideration of consistency in comparison frameworks for B2Bi methodologies. Our taxonomy is also useful for classifying approaches targeting at consistency checking and the survey we undertook not only proves empirical evidence for the existence of our CRs but also gives a starting point for finding relevant consistency checking methods.

In the future, each CR should be analyzed in more detail in order to develop differentiated criteria for evaluating support for a particular CR by a particular consistency checking method. Looking at the diversity of CRs identified, the need for integrated consistency support throughout the whole life cycle of B2Bis, i.e. the application of methods like (in)consistency management [27, 29], is striking. In particular, integrating consistency management practices into SWEs targeting at B2Bi is an interesting area of research. In this respect, the seamless application of existing consistency checking methods throughout several abstraction levels as well as enhancing the usability of rather scientific approaches is also an interesting area of research. Finally, special attention from the point of view of consistency is to be payed to the question whether BPMs should be directly mapped to public processes or indirectly via choreography specifications.

References

[1] E. M. Clarke Jr., O. Grumberg, and D. A. Peled, *Model checking*. Cambridge, MA, USA: MIT Press, 1999.

[2] C. Canal, E. Pimentel, and J. M. Troya, "Compatibility and inheritance in software architectures," *Sci. Comput. Program.*, vol. 41, no. 2, pp. 105–138, 2001.

[3] OASIS Open, "Web Services Atomic Transaction (WS-AtomicTransaction) version 1.1," July 2007.

[4] ——, "Web Services Business Activity (WS-BusinessActivity) version 1.1," July 2007.

[5] IEEE, *IEEE Standard Glossary of Software Engineering Terminology*, 1990.

[6] ISO/IEC, *Information technology - Open-edi reference model*, 2nd ed., ISO/IEC, May 2004.

[7] Dorn et al., "A survey of B2B methodologies and technologies: From business models towards deployment artifacts," *HICSS*, vol. 00, p. 143a, 2007.

[8] W3C, *Web Services Choreography Description Language*, 1st ed., W3C, November 2005.

[9] OASIS Open, *ebXML Business Process Specification Schema*, 2nd ed., OASIS Open, December 2006.

[10] ——, *Web Services Business Process Execution Language*, 2nd ed., April 2007.

[11] A. Schönberger and G. Wirtz, "Using Webservice Choreography and Orchestration Perspectives to Model and Evaluate B2B Interactions," in *SERP 2006*, June 26-29 2006.

[12] ——, "Realising RosettaNet PIP Compositions as Web Service Orchestrations - A Case Study," in *EEE 2006*, June 26-29 2006.

[13] A. Schönberger, "Modelling and Validating Business Collaborations: A Case Study on RosettaNet," University of Bamberg, Contributions to Applied and Business Informatics of University of Bamberg 65, Mar. 2006.

[14] OASIS Open, "WS-Reliability v1.1," November 2004.

[15] ——, "Web Services Security v1.1," February 2006.

[16] OMG, *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0*, OMG, January 2008.

[17] Koliadis et al., "Combining i* and BPMN for business process model lifecycle management," in *Business Process Management Workshops*, 2006, pp. 416–427.

[18] L. Baresi and S. Guinea, "Towards dynamic monitoring of WS-BPEL processes," in *ICSOC*, ser. LNCS, vol. 3826. Springer, 2005, pp. 269–282.

[19] M. Ilger and M. Zapletal, "An implementation to transform business collaboration models to executable process specifications," in *Service Oriented Electronic Commerce*, 2006, pp. 9–23.

[20] Zhao et al., "Towards the formal model and verification of web service choreography description language," in *WS-FM*, ser. LNCS, vol. 4184, 2006, pp. 273–287.

[21] W. Yeung, "Mapping WS-CDL and BPEL into CSP for behavioural specification and verification of web services," *ecows*, vol. 0, pp. 297–305, 2006.

[22] Weidlich et al., "Efficient analysis of BPEL 2.0 processes using p-calculus," *Asia-Pacific Service Computing Conference*, pp. 266–274, 11-14 Dec. 2007.

[23] Nezhad et al., "Web services interoperability specifications," *Computer*, vol. 39, no. 5, pp. 24–32, May 2006.

[24] A. Martens, "Consistency between executable and abstract processes," *e-Technology, e-Commerce and e-Service, 2005*, pp. 60–67, April 2005.

[25] W. van der Aalst, "Business alignment: using process mining as a tool for delta analysis and conformance testing," *Requirements Engineering Journal*, vol. 10, pp. 198–211, August 2005.

[26] Engels et al., "Towards consistency-preserving model evolution," in *IWPSE '02*. ACM, 2002, pp. 129–132.

[27] G. Spanoudakis and A. Zisman, *Handbook of Software Engineering and Knowledge Engineering*. World scientific, 2001, ch. Inconsistency management in software engineering, pp. 329–380.

[28] N. Medvidovic and R. N. Taylor, "A classification and comparison framework for software architecture description languages," *IEEE Transactions on Software Engineering*, vol. 26, no. 1, pp. 70–93, 2000.

[29] R. Van Der Straeten, "Inconsistency management in model-driven engineering," Ph.D. dissertation, Vrije Universiteit Brussel, Belgium, September 2005.

[30] Greenfield et al., "Consistency for web services applications," pp. 1199–1203, 2005.

[31] G. Decker and M. Weske, "Behavioral consistency for B2B process integration," in *CAiSE*, ser. LNCS, vol. 4495. Springer, 2007, pp. 81–95.

[32] Fettke et al., "Business process reference models: Survey and classification," in *Business Process Management Workshops*, vol. 3812, 2005, pp. 469–483.

[33] Kaschek et al., *Technologies for Business Information Systems*. Springer, 2007, ch. Characterization and Tool Supported Selection of Business Process Modeling Methodologies, pp. 25–37.