Towards Compliance of Cross-Organizational Processes and their Changes*

Research Challenges and State of Research

David Knuplesch[1], Manfred Reichert[1], Jürgen Mangler[2], Stefanie Rinderle-Ma[2], and Walid Fdhila[2]

[1] Institute of Databases and Information Systems, Ulm University, Germany
[2] Faculty of Computer Science, University of Vienna, Austria
{david.knuplesch,manfred.reichert}@uni-ulm.de,
{juergen.mangler,stefanie.rinderle-ma,walid.fdhila}@univie.ac.at

Abstract. Businesses require the ability to rapidly implement new processes and to quickly adapt existing ones to environmental changes including the optimization of their interactions with partners and customers. However, changes of either intra- or cross-organizational processes must not be done in an uncontrolled manner. In particular, processes are increasingly subject to compliance rules that usually stem from security constraints, corporate guidelines, standards, and laws. These compliance rules have to be considered when modeling business processes and changing existing ones. While change and compliance have been extensively discussed for intra-organizational business processes, albeit only in an isolated manner, their combination in the context of cross-organizational processes remains an open issue. In this paper, we discuss requirements and challenges to be tackled in order to ensure that changes of cross-organizational business processes preserve compliance with imposed regulations, standards and laws.

1 Introduction

Improving the efficiency and quality of their *business processes* and optimizing their interactions with partners, suppliers and customers have become significant success factors for any enterprise. Hence, enterprises increasingly adopt emerging technologies and standards for business process automation [1]. These enable the definition, execution, and monitoring of the operational processes of an enterprise [2, 3]. In connection with Web service technology, the benefits of business process automation and optimization from within a single enterprise can be transferred to cross-organizational business processes as well [4, 5]. The next step in this evolution will be the emergence of the agile enterprise being able to rapidly implement new processes and to quickly adapt existing ones to

* This work was done within the research project C^{3}Pro funded by the German Research Foundation (DFG), Project number: RE 1402/2-1, and the Austrian Science Fund (FWF), Project number: I743.

environmental changes [3]. While flexibility issues for internal business processes and their implementation (i.e., *process* and *service orchestrations*) are well understood [3, 2], the controlled change of the interactions between IT-supported partner processes in a cross-organizational setting (i.e., *process and service choreographies*) has not been adequately addressed so far. If one partner changes its process in an uncontrolled manner, inconsistencies or errors (e.g., deadlocks) regarding these interactions might occur. This is particularly challenging if there exist running instances (i.e., cases) of these process choreographies. As a consequence, adaptations of cross-organizational business processes turn out to be costly and error-prone.
Generally, business processes cannot be defined or changed without considering *business process compliance* with imposed compliance rules (e.g., security guidlines). Due to the increasing importance of regulations like SOX and BASEL, compliance has emerged as one of the most urgent challenges for process-aware information systems. So far, it has been addressed by many approaches, which mostly deal with the automation of audits for verifying compliance rules imposed on internal business processes [6, 7, 8, 9].
Compliance of cross-organizational business processes, however, has not been investigated in connection with process changes or with respect to privacy constraints of partner processes. *Flexibility* on one hand and *compliance* on the other hand are crucial challenges for collaborative settings. However, the picture will be not complete if we do not consider both in interplay as well. Even though a compliance rule might be fulfilled for a collaborative process before a change, it does not automatically remain satisfied afterwards. Thus, it is indispensable to provide adequate mechanisms to control change effects on the compliance of collaborative business processes in a transparent way.

This paper first provides an overview of basic notations related to compliance and (cross-organizational) business processes in Section 2. Section 3 then discusses different layers of correctness that must be considered when modeling and changing business process models. Section 4 discusses the state of the art in related research areas; i.e., cross-organizational processes, process flexibility, and business process compliance. Section 5 elicits novel requirements and challenges for enabling compliance of cross-organizational business processes and their change. Finally, Section 6 closes the paper with summary and conclusion.

2 Basic Notion and Example

We introduce basic notions related to business compliance in the context of cross-organizational processes and discuss their inter-relations. Fig. 1 gives an overview of the terminology we use. Basic to each cross-organizational process is a *collaboration*. In turn, the latter contains abstract *roles* that may be filled by concrete *business partners*. For each role, the set of related *task definitions* describes which tasks a role may perform when a cross-organizational process is executed. The model of a cross-organizational process is denoted as *choreography model* and consists of *public process models* connected through *message exchanges*. In turn, a public process model consist of tasks and refers to a role, whereas a private process model implements a public process model of a business partner enriched with internal (i.e., private) tasks. Both public and private process models may only comprise task definitions related to the role the public process model refers to.

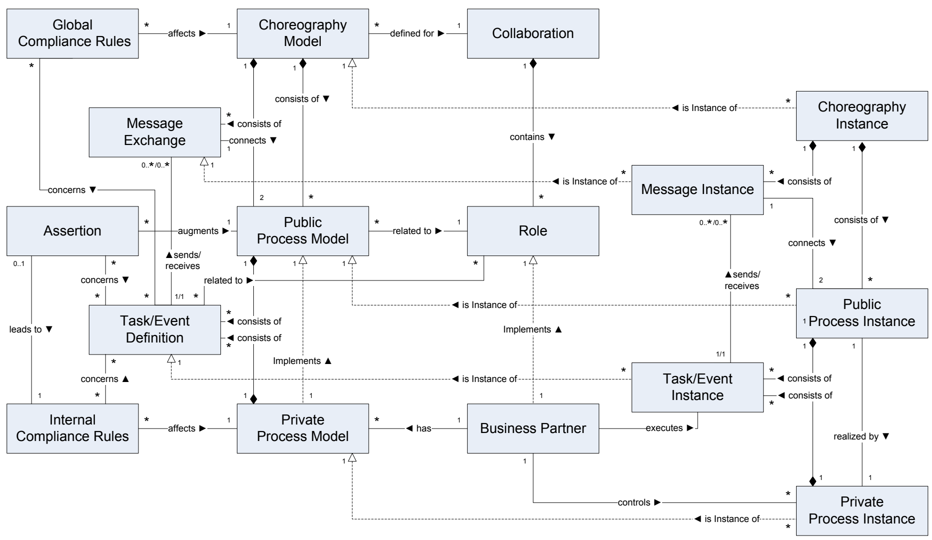

Fig. 1. Terminology Overview

A *choreography instance* reflects the execution of a cross-organizational process defined by a choreography model. Similar to a choreography model, the choreography instance consists of *public process instances*, which are based on the *private process instances* of the related business partner. Such a private process instance is controlled by the respective business partner through executing its *task instances*. The latter send and receive *message instances* connecting the public process instance within the choreography instance.

Both the choreography model and the private process models may be subject to *compliance rules* restricting allowed execution sequences of tasks. In this context, we distinguish between *global compliance rules* referring to a cross-organizational process (i.e., to a choreography model) and *internal compliance rules* restricting private process models. Furthermore, *assertions* can augment public process models. More precisely, they enrich a choreography's specification of a public process model with additional information about its executon behavior. In turn, an assertion constitutes an internal compliance rule restricting the private process model that implements the augmented public process model.

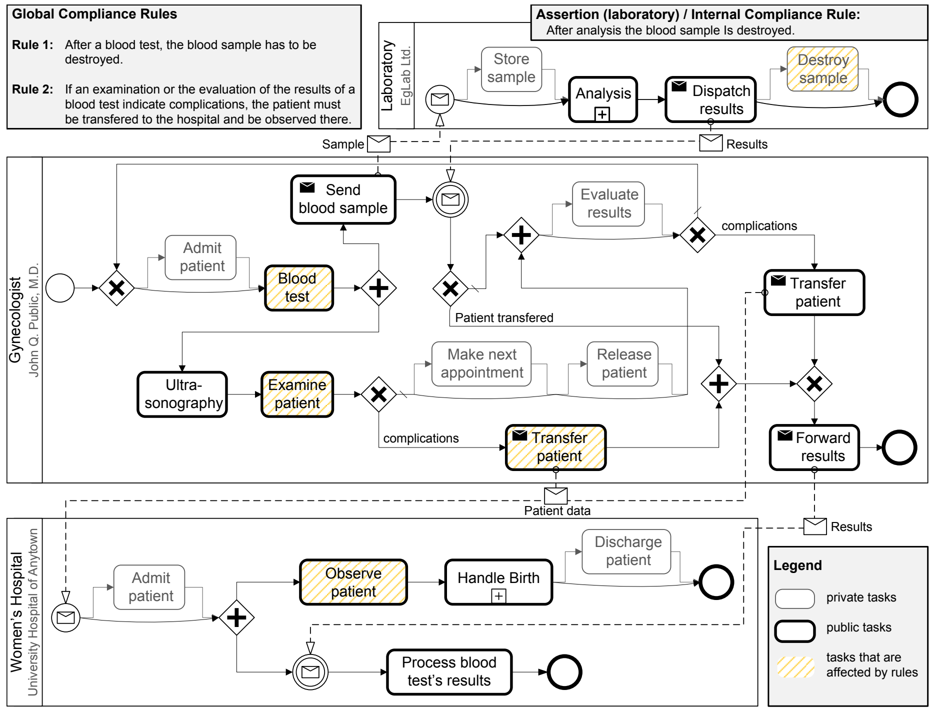

Fig. 2. Example of Cross-organizational Process

Example 1. Fig. 2 shows an example of a cross-organizational process from the healthcare domain. The fundamental collaboration comprises three *roles*: `gynecologist`, `laboratory`, and `women's hospital`. They are filled by the three business partners John Q. Public, M.D., EgLab Ltd., and University Hospital of Anytown. The *choreography model* of this cross-organizational process is based on the *public process models* of the three roles that communicate through *message exchanges*. For example, the *message exchanges* `sample` and `results` connect the public process of the `gynecologist` with the one of the `laboratory`. In turn, the public process of the `gynecologist` is connected with the one of the `women's hospital` based on the two message exchanges `patient data` and `results`. The *private process models* of the business partners, in turn, implement the related public processes and additionally enrich them with internal behavior, i.e. private tasks not contained in public processes models. (In Fig. 2 the private tasks are indented and greyed). The logical execution of this *choreography model* results in a *choreography instance* of the cross-organizational healthcare process; it contains related instances of the choreography, i.e. its public as well as private processes, and the messages exchanged. Finally, Fig. 2 shows one assertion and two global compliance rules. Through the `assertion`, the `laboratory` ensures the destruction of a blood sample after its analysis. `Rule 1` requires a blood sample to be destroyed. In turn, `Rule 2` requires the patient to be monitored in the hospital when complications occur.

3 Correctness Levels for Process Models

Business processes models cannot be defined or changed in an arbitrary way. Generally, three correctness levels build on each other and constitute a *pyramid of business process model correctness*. In particular, each level has to be considered and checked for each definition or change of a process model (cf. Fig. 3):

Syntactical Correctness refers to the correct use and composition of the elements of the underlying meta model. Examples of syntactical constraints include the existence of start and end events, as well as the correct use of different kinds of edges; e.g., control flow edges may only connect flow elements (i.e., tasks, gateways, events), while data flow edges connect tasks with data objects. *Syntactical correctness* is a prerequisite for *behavioral correctness*, since the behavior of a syntactical incorrect model is undefined.

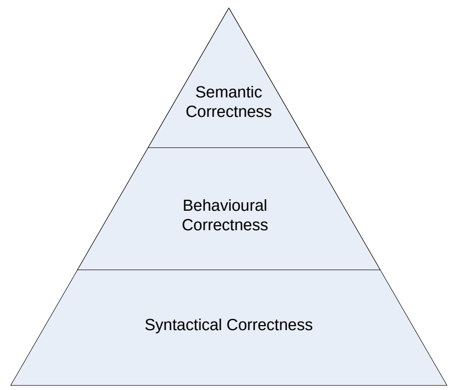

Fig. 3. Pyramid of Business Process Model Correctness

Behavioral Correctness requires from a process model to be executable and includes properties like the absence of deadlocks and lifelocks. Further, it requires a correct flow of data; e.g., data objects must be written before read the first time. In the context of (dynamic) process changes, *state consistency* and *data consistency* respectively, must be preserved; i.e., a running process instance must not face deadlocks, lifelocks, and data-flow errors, when dynamically migrating its execution to a new process model version. In the context of cross-organizational processes, *compatibility* between the public processes of the different partners requires their composition is a behaviorally correct process. *Conformance* requires the private process of a partner to implement the behavior of his public process. Finally, *behavioral correctness* is a prerequisite for *semantic correctness.*

Semantic Correctness (i.e. business process compliance) requires a process model to comply with imposed compliance rules stemming from regulations, standards, and laws. Hence, each possible execution of the process must not violate any compliance rule. For a set of compliance rules, *consistency* requires the conjunction of the rules to be satisfiable; i.e. consistent compliance rules do not conflict with each other.

4 State of the Art

In this section, we outline the state of research. First, we discuss existing approaches in the area of *cross-organizational processes*. Then, we deal with *process flexibility*. Finally, we discuss *business process compliance*.

4.1 Cross-organizational Processes

Modeling *cross-organizational* processes has been discussed for many years. There are widespread standards such as BPEL4WS, WS-CDL, and RosettaNet, as well as powerful modeling frameworks and notations (e.g. "Let's dance" [10], iBPMN [11], and BPMN 2.0). [12] further identified *interaction patterns*, which describe well-defined patterns for message exchanges between partner processes. For privacy reasons, a choreography definition is usually restricted to those activities relevant for the message exchanges between the partners involved. More precisely, partners publish restricted views on their private processes [13, 14]. Several top-down-approaches exist, which, starting from a choreography of public processes, determine whether or not private processes comply with the corresponding public ones [15, 16, 17]. Furthermore, [18] introduces a set of transformation rules that allow for the inheritance-preserving stepwise enrichment of a public process to obtain the corresponding private one. These rules are also applicable to evolve private processes without changing their public view. To support the opposite direction (i.e., to interconnect existing partner processes), [19] provides a bottom-up-approach for checking whether or not processes can interact with each other successfully. Further, they propose a central as well as distributed architecture, which both allow for the dynamic matching and execution of cross-organizational processes based on a shared registry for public processes. Finally, the scenario of modeling process choreographies and private processes independently is addressed by [20], using the *Formal Contract Language* to check conformance between choreographies and processes.

4.2 Process Flexibility

Issues related to process flexibility have been discussed for more than a decade [21, 22, 23, 5]. However, existing approaches consider flexibility mainly for processes orchestrations; i.e., workflows implementing a private process and being executed by a single process engine. In approaches like Pockets of Flexibility [24] or Worklets [25], for example, processes are executed on the basis of a loosely or partially specified model, which is then fully specified at runtime. Relevant techniques include late modeling and late composition of process fragments as well as declarative modeling styles [26].

Dynamic process adaptations, in turn, represent the ability of the implemented processes to cope with exceptional situations. On one hand, this includes the handling of expected exceptions, which can be anticipated and thus be captured in the process model [27]. On the other hand, it covers non-anticipated exceptions, which are usually handled through structural adaptations of single process instances based on well-defined change patterns (e.g., to add, delete or move activities) [28]. A particular challenge is to ensure the behavioral correctness (i.e., state and data consistency) of a process instance in this context [21]. Approaches like ADEPT [29] guarantee for the behavioral correctness of the modified process model.

Besides this, there exists support for assisting end-users in reusing ad-hoc changes [30] and for restricting changes to authorized users [31]. Another fundamental aspect concerns process schema evolution [32, 33]; i.e., the ability of the implemented process to change when the business process evolves. Relevant problems in this context concern the handling of running process instances, which were initiated based on the old model version, but are required to use the new specification from now on [33, 32]. Since thousands of active instances may be affected by a given process change the issue of behavioral correctness is rather critical. Traceability of changes and mining of dynamic processes are other relevant issues, closely related to process evolution, which are considered in existing frameworks [34, 35].

Only few approaches exist that address changes of distributed processes and choreographies. [36] shows how partitioned workflows can be changed in a controlled way. [37] distinguishes between shallow and deep service changes in the context of a choreography. While the effects of shallow changes (e.g., changes of service versions, interfaces, and operations) are restricted to a service, deep changes have cascading and disseminating effects on the whole choreography. [38] describes how the version of stateful service instances can be changed efficiently, if the behavior of the new service version covers the behavior of the replaced one. Constructing such new service versions is addressed by [39]. However, no comprehensive solution approach is provided.

4.3 Business Process Compliance

In many domains, process execution is subject to compliance rules and restrictions that stem from laws, regulations, and guidelines (e.g. Basel or Sarbanes-Oxley-Act) [40]. Existing approaches that allow ensuring compliance of business processes with imposed compliance rules differ with respect to the lifecycle phase in which compliance is verified as well as the strategy applied. Moreover, different paradigms and formalisms are used to define compliance rules and process models [41]. Compliance rules are often considered as restrictions to the order in which process activities may be executed. In this context, there exist approaches that formalize compliance rules in temporal logic (e.g., LTL [8] and CTL [6]). Other ones emphasize the modalities of compliance rules (e.g., obligations or permissions) by applying deontic logic [42, 43]. Since these approaches are not easy to comprehend, [44] suggests a pattern-based approach to encapsulate logic. There also exist graphical notations for modeling compliance rules [7, 8, 45]. The integration of compliance rules throughout the process lifecycle has been discussed in [46, 47, 48].

To verify whether compliance rules are fulfilled by process models at design time, many approaches apply model checking techniques [6, 7, 8, 9]. Since these depend on the exploration of the state space of process models, state space explosion constitutes a big obstacle for practical applications. Graph reduction and sequentialization of parallel flows as well as predicate abstraction are proposed to deal with this challenge [8, 7, 49]. Among those approaches, there exist few that do not only consider the control flow perspective. [50] introduces state-based data objects. [49] enables data-aware compliance checking for larger data-domains and [9] additionally considers temporal constraints. For cycle-free processes, there exist algorithms that allow for more efficient design time compliance verification than model checking [20, 51].

Runtime checking and monitoring (i.e. continous auditing [52]) of business process compliance are addressed by several approaches: [53] enriches a process models with a semantic layer of internal controls. Another compliance monitoring framework based on common event standards and middleware is a presented in [54]. [55] discusses the monitoring and enforcement of compliance within process collaborations. [56] uses *Compliance Rule Graphs* and [57] colored automata to enable fine-grain compliance diagnostics at runtime.

To complement design time and runtime compliance checking, backward compliance checking of process logs has been proposed by checking logs for compliance with LTL-formulas [58]. Finally, declarative approaches [26, 43, 42] ensure compliance in an elegant manner. Since processes are defined by means of a set of rules, imposed compliance rules only have to be added to the process definition to ensure business process compliance.

In summary, there exist many approaches considering aspects of compliance of cross-organizational processes and their changes. However, only few approaches discuss flexibility issues in the context of cross-organizational processes or in the context of business process compliance. Even fewer approches address business process compliance of cross-organizational processes. Finally, Fig. 4 shows that the interplay of change and compliance in the context of cross-organizational processes has not been addressed yet.

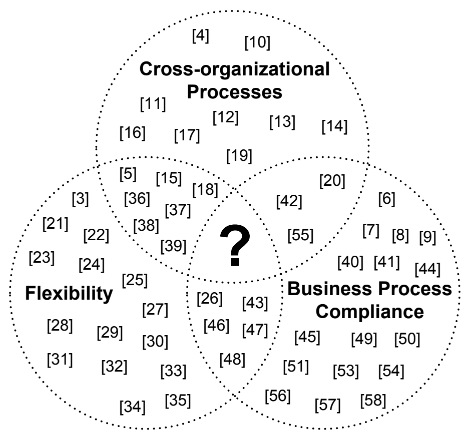

Fig. 4. State of the Art

5 Challenges

As mentioned in Section 4, the interplay of change and compliance in the context of cross-organizational processes has not been addressed yet. In this section we outline challenges arising in the context of this interplay:

Modeling Cross-organizational Compliance Rules. First of all, the (graphical) modeling of assertions and cross-organizational compliance requires proper support. In addition to intra-organizational compliance rules, messages as well as the role executing a task must be considered (cf. `Rule 2` of Example 1).

Change Propagation. Before applying a change, all three correctness levels have to be checked as we will illustrate by means of the scenario from Fig. 2: If the message-sending task `forward results` is deleted in the `gynecologist`'s private process, the corresponding public process will be affected. Furthermore, the message `results` as well as the public and private process of the `women's hospital` will be affected by this change. This indicates that a partner must not change his public process in an uncontrolled manner. Otherwise, behavioral correctness of the choreography his public process is involved in cannot be preserved. Thus, the `women's hospital` should be supported in adapting its public and private processes to the change. Hence, a major research challenge is how to *propagate* changes of a partner's private process to the other partners in order to keep the choreography behaviorally correct.

Efficient Cross-organizational Instance Migration. Changes are particularly challenging if there exist running choreography instances being in different states or partially differing from the original process model (e.g., due to adaptations of single choreographies in the context of exceptions). For any of these hundreds or even thousands of choreography instances it has to be determined whether or not the change or parts of it are applicable, while satisfying all three correctness levels at the same time. Since large numbers of cross-organizational process instances may be affected, efficient algorithms are required.

Ensuring Compliance with Regard to Privacy. Using Example 1 from Fig. 2, we can show that privacy constraints of partner processes aggravate compliance checking at design time. For example, the `gynecologist` cannot correctly verify `Rule 1`, since the `laboratory` hides the relevant task `destroy sample` from its public process. Nevertheless, when additionally considering the assertion of the laboraty, it becomes clear that `Rule 1` is satisfied. Similar difficulties occur when compliance is monitored at runtime. This requires a solution supporting compliance checking with respect to privacy issues. This may become more difficult or even impossible if partners do not publish any parts of their processes.

Efficiently Ensuring Compliance at Change Time. Ensuring compliance of cross-organizational process models and instances at change time is another challenging research issue. Particularly, in the context of compliance, effects of changes cannot be easily traced. For example, assume that task `destroy sample` is deleted in the `laboratory`'s private process (cf. Fig. 2). Although this task is not part of the `laboratory`'s public process, compliance of the `gynecologist`'s process models with `Rule 1` will be affected by this change. However, the task affects the `laboratory`'s `assertion` that must be withdrawn. Consequently, assertions, internal compliance rules, and global compliance rules should be re-evaluated. Due to the high complexity of current compliance checking approaches and due to the potentially large number of affected process instances, optimization strategies and efficient algorithms are required.

Adequate User Feedback. Users require intelligible feedback on compliance violations. Sources and effects of compliance violations have to be highlighted.

Furthermore, the available courses of action for healing compliance violations should be offered to selected user roles.

6 Summary and Outlook

This paper emphasized that ensuring compliance for cross-organizational processes and their changes raises several challenges. We first provided an overview about the relevant terms in this context, and introduced an example from the medical domain. Next, we introduced three correctness levels, i.e., syntactical correctness, behavioral correctness, and semantic correctness. Then we discussed the state of the art. We came to the viewpoint that the interplay of change and compliance in the context of cross-organizational processes has not been adressed yet. Finally, we provided unanswered challenges that raise from that interplay.

In our future work, we plan to adress the challenges denoted in Section 5, within our research project C^3Pro: First, we want to enable graphically modeling of assertions and cross-organizational compliance rules. Second, we will put emphasis on enabling cross-organizational compliance checking and monitoring with respect to privacy issues and assertions. As stated, this requires the checking of syntactical and behavioral correctness before. In addition, we plan to study the efficient application of changes to running choreographies instances first and then look and instances that partially differ from the original process models. Finally, we plan to combine our approaches for ensuring compliance and correct changes to cross-organizational processes and thus enable compliance of cross-organizational changes.

References

1. Mutschler, B., Reichert, M., Bumiller, J.: Unleashing the effectiveness of process-oriented information systems: Problem analysis, critical success factors, and implications. IEEE Trans on Sys, Man, and Cybernetics, Part C **38**(3) (2008) 280–291
2. Weske, M.: Workflow management systems: Formal foundation, conceptual design, implementation aspects. Springer (2007)
3. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems. Springer (2012)
4. Alonso, G., et al.: Web Services. Springer (2004)
5. Dustdar, S.: Caramba - a process-aware collaboration system supporting ad hoc and collaborative processes in virtual teams. Dist & Parall Datab **15**(1) (2004) 45–66
6. Ghose, A.K., Koliadis, G.: Auditing business process compliance. In: ICSOC'07. (2007) 169–180
7. Liu, Y., Müller, S., Xu, K.: A static compliance-checking framework for business process models. IBM Systems Journal **46**(2) (2007) 335–261
8. Awad, A., Decker, G., Weske, M.: Efficient compliance checking using BPMN-Q and temporal logic. In: BPM'08. (2008) 326–341
9. Kokash, N., Krause, C., de Vink, E.: Time and data aware analysis of graphical service models. In: SEFM'10. (2010)

10. Zaha, J., Barros, A., Dumas, M., ter Hofstede, A.: Let's dance: A language for service behavior modeling. In: CoopIS'06. (2006) 145–162
11. Decker, G., Weske, M.: Interaction-centric modeling of process choreographies. Inf Sys **35**(8) (2010)
12. Barros, A., Dumas, M., ter Hofstede, A.: Service interaction patterns. In: BPM'05. (2005) 302–318
13. Liu, D.R., Shen, M.: Business-to-business workflow interoperation based on process-views. Decision Support Sys **38**(3) (2004) 399–419
14. Maamar, Z., Benslimane, D., Ghedira, C., Mrissa, M.: Views in composite web services. IEEE Internet Comp **9**(4) (2005) 52–57
15. van der Aalst, W.M.P.: Inheritance of interorganizational workflows to enable Business-to-Business E-Commerce. Elec Com Research **2**(3) (2002) 195–231
16. Martens, A.: Consistency between executable and abstract processes. In: EEE'05. (2005) 60–67
17. Decker, G., Weske, M.: Behavioral consistency for B2B process integration. In: CAiSE'07. (2007) 81–95
18. van der Aalst, W.M.P., et al.: Multiparty contracts: Agreeing and implementing interorganizational processes. The Comp Journal **53**(1) (2010) 90–106
19. Tata, S., et al.: CoopFlow: A Bottom-Up Approach to Workflow Cooperation for Short-Term Virtual Enterprises. IEEE Trans on Serv Comp **1**(4) (2008) 214–228
20. Governatori, G., Milosevic, Z., Sadiq, S.: Compliance checking between business processes and business contracts. In: EDOC'06. (2006) 221–232
21. Rinderle, S., Reichert, M., Dadam, P.: Correctness criteria for dynamic changes in workflow systems – a survey. Data & Knowl Eng **50**(1) (2004) 9–34
22. Weber, B., Sadiq, S., Reichert, M.: Beyond rigidity–dynamic process lifecycle support. Comp Science-Research and Dev **23**(2) (2009) 47–65
23. Reichert, M., Dadam, P.: ADEPT$_{flex}$ – supporting dynamic changes of workflows without losing control. Intelligent Inf Sys **10**(2) (1998) 93–129
24. Sadiq, S., Sadiq, W., Orlowska, M.: A framework for constraint specification and validation in flexible workflows. Inf Sys **30**(5) (2005) 349–378
25. Adams, M., ter Hofstede, A., Edmond, D., van der Aalst, W.M.P.: Worklets: A service-oriented implementation of dynamic flexibility in workflows. In: CoopIS'06. (2006) 291–308
26. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: EDOC'07. (2007) 287–300
27. Reichert, M., Dadam, P., Bauer, T.: Dealing with forward and backward jumps in workflow management systems. Software and Systems Modeling **2**(1) (2003) 37–58
28. Weber, B., Reichert, M., Rinderle-Ma, S.: Change patterns and change support features - Enhancing flexibility in process-aware information systems. Data & Knowl Eng **66**(3) (2008) 438–466
29. Reichert, M., Rinderle, S., Kreher, U., Dadam, P.: Adaptive process management with ADEPT2. In: ICDE'05. (2005) 1113–1114
30. Weber, B., et al.: Providing integrated life cycle support in process-aware information systems. Int J Coop Inf Sys **18**(1) (2009) 115–165
31. Weber, B., Reichert, M., Wild, W., Rinderle, S.: Balancing flexibility and security in adaptive process management systems. In: CoopIS'05. (2005) 59–76
32. Casati, F., et al.: Workflow evolution. Data & Knowl Eng **24**(3) (1998) 211–238
33. Rinderle, S., Reichert, M., Dadam, P.: Flexible support of team processes by adaptive workflow systems. Dist & Parall Datab **16**(1) (2004) 91–116
34. Li, C., et al.: The MinAdept clustering approach for discovering reference process models out of process variants. Coop Inf Sys **19**(3) (2010)

35. Günther, C., Rinderle, S., Reichert, M., van der Aalst, W.M.P.: Change mining in adaptive process management systems. In: CoopIS'06. (2006) 309–326
36. Reichert, M., Bauer, T.: Supporting ad-hoc changes in distributed workflow management systems. In: CoopIS'07. (2007) 150–168
37. Papazoglou, M.: The challenges of service evolution. In: CAiSE'08. (2008) 1–15
38. Liske, N., Lohmann, N., Stahl, C., Wolf, K.: Another approach to service instance migration. Service-Oriented Computing (2009) 607–621
39. Mooij, A., et al.: Constructing replaceable services using operating guidelines and maximal controllers. Web Services and Formal Methods (2011) 116–130
40. Sadiq, S., Governatori, G., Naimiri, K.: Modeling control objectives for business process compliance. In: BPM'07. (2007)
41. El Kharbili, M., et al.: Business process compliance checking: Current state and future challenges. In: MobIS'08. (2008) 107–113
42. Alberti, M., et al.: Expressing and verifying business contracts with abductive logic programming. In: NorMAS'07. Dagstuhl Seminar Proceedings (2007)
43. Goedertier, S., Vanthienen, J.: Designing compliant business processes with obligations and permissions. In: BPM'06 Workshops. (2006) 5–14
44. Dwyer, M.B., Avrunin, G.S., Corbett, J.C.: Property specification patterns for finite-state verification. In: FMSP'98. (1998)
45. Ly, L.T., et al.: Design and verification of instantiable compliance rule graphs in process-aware information systems. In: CAiSE'10. (2010) 9–23
46. Ly, L.T., et al.: Integration and verification of semantic constraints in adaptive process management systems. Data & Knowl Eng **64**(1) (2008) 3–23
47. Ly, L.T., Rinderle-Ma, S., Göser, K., Dadam, P.: On enabling integrated process compliance with semantic constraints in process management systems - requirements, challenges, solutions. Inf Sys Frontiers (2009)
48. Knuplesch, D., Reichert, M.: Ensuring business process compliance along the process life cycle. Technical Report 2011-06, University of Ulm (2011)
49. Knuplesch, D., Ly, L.T., Rinderle-Ma, S., Pfeifer, H., Dadam, P.: On enabling data-aware compliance checking of business process models. In: ER'2010. (2010)
50. Awad, A., Weidlich, M., Weske, M.: Specification, verification and explanation of violation for data aware compliance rules. In: ICSOC'09. (2009) 500–515
51. Weber, I., Hoffmann, J., Mendling, J.: Semantic business process validation. In: SBPM'08. (2008)
52. Alles, M., Kogan, A., Vasarhelyi, M.: Putting continuous auditing theory into practice: Lessons from two pilot implementations. Inf Sys **22**(2) (2008) 195–214
53. Namiri, K., Stojanovic, N.: Pattern-Based design and validation of business process compliance. In: CoopIS'07. (2007) 59–76
54. Giblin, C., et al.: From regulatory policies to event monitoring rules: Towards model-driven compliance automation. Technical Report RZ-3662, IBM (2006)
55. Berry, A., Milosevic, Z.: Extending choreography with business contract constraints. Coop Inf Sys **14**(2-3) (2005) 131–179
56. Ly, L.T., Rinderle-Ma, S., Knuplesch, D., Dadam, P.: Monitoring business process compliance using compliance rule graphs. In: CoopIS'11. (2011) 82–99
57. Maggi, F., Montali, M., Westergaard, M., van der Aalst, W.M.P.: Monitoring business constraints with linear temporal logic: an approach based on colored automata. In: BPM'11. (2011) 132–147
58. van der Aalst, W.M.P., Beer, H.D., van Dongen, B.: Process mining and verification of properties: An approach based on temporal logic. In: CoopIS'05. (2005) 130–147