Detecting the Effects of Changes on the Compliance of Cross-organizational Business Processes *

David Knuplesch[1], Walid Fdhila[2], Manfred Reichert[1], and Stefanie Rinderle-Ma[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,
{walid.fdhila,stefanie.rinderle-ma}@univie.ac.at

Abstract. An emerging challenge for collaborating business partners is to properly define and evolve their cross-organizational processes with respect to imposed global compliance rules. Since compliance verification is known to be very costly, reducing the number of compliance rules to be rechecked in the context of process changes will be crucial. Opposed to intra-organizational processes, however, change effects cannot be easily assessed in such distributed scenarios, where partners only provide restricted public views and assertions on their private processes. Even if local process changes are invisible to partners, they might affect the compliance of the cross-organizational process with the mentioned rules. This paper provides an approach for ensuring compliance when evolving a cross-organizational process. For this purpose, we construct qualified dependency graphs expressing relationships between process activities, process assertions, and compliance rules. Based on such graphs, we are able to determine the subset of compliance rules that might be affected by a particular change. Altogether, our approach increases the efficiency of compliance checking in cross-organizational settings.

Keywords: Process Compliance, Process Change, Cross-organizational Process

1 Introduction

Ensuring the compliance of their business processes is crucial for enterprises [1]. To cope with this challenge, a variety of approaches are proposed that allow verifying the compliance of business processes with semantic compliance rules (e.g., domain-specific standards, guidelines, and regulations) [2–5]. In this context, only few approaches consider *cross-organizational business processes* (CBP); i.e., processes involving multiple partners. Ensuring the compliance of a CBP with

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

global compliance rules (GCR), however, raises additional challenges. In particular, compliance checking must cope with the fact that the partners do not know all parts of the CBP relevant for a GCR, e.g., due to privacy reasons [6, 7]. In this context, we developed techniques to *a priori* ensure compliance of a CBP with global rules [8]. For this purpose, we utilize the *public views* on the processes of the involved partners (i.e. *public process models*) as well as declarative *assertion rules* (AR), provided by each partner on the behavior of its private process. Note that respective views and assertions allow us to approximate the behavior of the partner processes and, hence, the CBP, while satisfying privacy issues. As further shown in [9], CBPs may be subject to change (e.g., a partner may want to change his process or partner interactions shall be changed). Existing work has already addressed issues related to the behavioral correctness (i.e, soundness) of CBPs in the context of such changes [9–12]. In turn, only few approaches address the issue whether or not a changed CBP still complies with a given set of compliance rules [6, 13, 7]. Unfortunately, these approaches do not provide a solution that considers privacy issues of the involved partners.

An obvious approach to ensure compliance of a changed CBP would be to recheck the former with all imposed compliance rules, e.g., utilizing the approach presented in [8]. However, compliance checking is known to be time-consuming and costly [2, 14]. In particular, this applies to *privacy-aware compliance checking* of a CBP, which not only needs to explore the state space of the specified models, but additionally must estimate the effects of activities not visible to all partners due to privacy constraints [8]. Consequently, being able to detect the possible effects, a CBP change may have on the compliance of the CBP with defined rules, would contribute to limit the number of compliance checks to be repeated after having changed the CBP. More precisely, only those compliance rules, which may be affected by the change, should be rechecked (cf. Fig. 1).

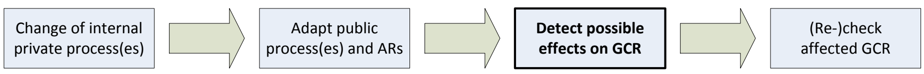

Fig. 1. Ensuring compliance of cross-organizational processes after a change

Another *naive approach* would be to solely recheck compliance of those rules referring to activities that are directly affected by the change. However, this approach is not sufficient. On one hand, as known from intra-organizational scenarios [3], it would also recheck compliance rules that become overfulfilled due to a change (*false positive*); e.g., adding a second safety check, although the required one has already been ensured. On the other hand, as opposed to intra-organizational scenarios, the naive approach is unable to identify all compliance rules to be rechecked when the CBP is being changed (*false negative*). This effect occurs when changes not only affect public, but also private elements of partner processes. Although the latter kind of changes are not visible, they could partially be assessed based on direct as well as transitive correlations between changes and assertions. Note that *false positive* cases result in superfluous checks, whereas *false negative* ones give a false sense of security, since they might prevent the detection of compliance violations.

This paper provides a sophisticated approach that enables us to detect and qualify all possible effects any CBP change may have on CBP compliance. To deal with both *false positive* and *false negative* cases, first of all, the dependencies between activities, assertions and compliance rules are analyzed before representing them as *qualified dependency graphs*. Based on the latter, two algorithms are introduced that allow assessing the possible effects a CBP change has on the compliance of the CBP with imposed rules. These algorithms not only enable us to detect possible new compliance violations introduced by the CBP change, but additionally allow determining whether a change has the potential to heal already present violations of a particular rule. The approach is illustrated along a realistic use case and evaluated by a proof-of-concept implementation that was applied to different scenarios.

The remainder of this paper is structured as follows: Section 2 provides a running example. Fundamentals are introduced in Sect. 3, whereas Sect. 4 presents the approach. In detail, Sect. 4.1 investigates the dependencies between activities, compliance rules, and assertions. It further introduces the qualified dependency graphs. Sect. 4.2 then presents algorithms that analyze the dependency graph to detect the effects of CBP changes. A proof-of-concept is provided in Sect. 5. Sect. 6 discusses related work and Sect. 7 summarizes the paper.

2 Running Example

This section introduces a cross-organizational supply chain scenario as well as a related change (cf. Fig. 2). The scenario highlights the effects of changes on the compliance of a CBP [7]. A CBP involving 6 partners is introduced. It describes a supply chain from the *bulk buyer*'s order of a product batch via the order and provisions of two *intermediate products a* and *b* by two *suppliers A* and *B* to production and, finally, delivery by the *manufacturer*. Furthermore, the order and delivery of *intermediate product a* not only involve *supplier A* and *manufacturer*, but also a *middleman* and a *special carrier*.

The depicted supply chain process complies with the following 5 global compliance rules (GCR), which reflect regulations and standards to be obeyed:

C1: After production the final test must be performed.
C2: A full test of intermediates is required before starting the production.
C3: Each transport of the intermediate product *a* requires permission of authority. Furthermore, the special transporter must pass a safety check before.
C4: After a quick test, the parameters of the tests must be compared to ensure validity.
C5: If an intermediate is transported after its full test, a quick test is required after arrival and before production.

The partners only share public views on their processes in order to ensure privacy; e.g., *special carrier* abstracts from activity *safety check* by hiding the latter, whereas *middleman* hides activity *get permission from authority* (cf. Fig. 2). To enable the verification of the GCRs C1-C5, the partners provide the following assertion rules (AR) on the hidden behavior of their private processes:

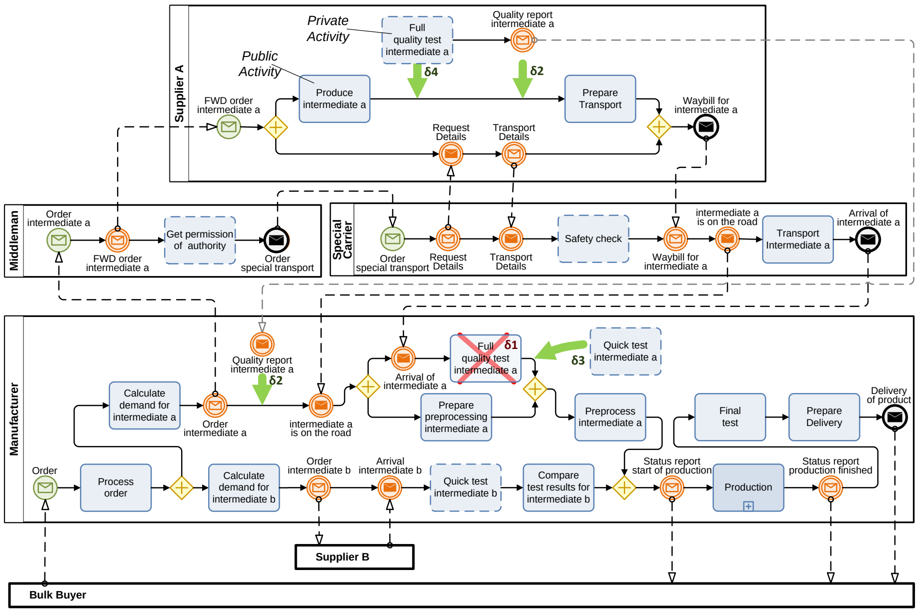

Fig. 2. Example: Cross-organizational supply chain process

A1: *Manufacturer* assures that a quick test is performed after the arrival of an intermediate and before processing it, if the manufacturer does not perform a full quality test in this period (A1.1). In turn, if a full quality test is performed after arrival and before production, the manufacturer does not perform a quick test (A1.2).
A2: *Middleman* assures that it gets permission from authority for the special transport before ordering the latter.
A3: *Special carrier* assures to perform a safety check before starting the transport of intermediate a.
A4: *Supplier B* assures that a full quality test of intermediate b is performed before the latter arrives at the *manufacturer* side.

When utilizing assertions A1-A4, one can successfully verify compliance of the CBP with C1-C5 based on the public process views (cf. [8]).

Change scenario. To decrease costs as well as to optimize the processing of intermediate *a*, *manufacturer* skips the full quality test of intermediate *a*. Instead, the full test shall now be performed by *supplier A* and a quality report be sent to *manufacturer*. In particular the following changes occur (cf. Fig. 2):

δ1: *Manufacturer* skips the *full quality test for intermediate a*.
δ2: Message *quality report for intermediate a* from *supplier A* to *manufacturer* is added.
(δ3: *Manufacturer* adds private activity *quick test for intermediate a*.)
(δ4: *Supplier A* adds private activity *full quality test for intermediate a*.)
δ5: *Supplier A* publishes new *assertion A5*. The latter shall guarantee that supplier A performs a *full quality test for intermediate a* before sending the corresponding *quality report*.

Only δ1, δ2 and δ5 are visible, whereas δ3 and δ4 as well as their effects (i.e., the insertion of private activities) remain hidden from the partners.

Note that it is evident that C2 should be rechecked when considering the public changes, because activity *full quality test for intermediate a* is affected by public change $\delta 1$. By contrast, the public changes do not directly imply the need to recheck C4. However, C4 becomes violated, since a *quick test for intermediate a* occurs, but the parameters of the tests are not compared. Hence, C4 constitutes a *false negative* case as described in the context of the *naive approach.*

3 Fundamentals

This paper aims to optimize compliance checking for evolving (i.e., changing) cross-organizational business processes (CBP) with a fixed set of partners $\mathcal{P}$. As opposed to an intra-organizational process, a CBP is not only based on a set of activities $\mathcal{A}$, but also comprises a set of interactions $\mathcal{I}$. The latter correspond to the messages exchanged between the partners. Note that different, but partially overlapping viewpoints (i.e., process models) on a CBP exist [9, 8]:

- A *private process model* describes the internal business logic of a partner and defines the execution constraints for its activities and interactions.
- A *public process model*, in turn, provides a restricted view on a private process model. In particular, it only contains *public (i.e., visible) activities* and *interactions*, but hides *private activities* and internal details of the private process. In this context, we refer to $\mathcal{A}_v$ as the set of all public activities and $\mathcal{A}_h$ as the set of all private activities of a CBP. [3]

We focus on compliance checking and, hence, presume structural and behavioral correctness of both public and private models (see [15–17] for respective approaches). Furthermore, the different viewpoints on cross-organizational processes need to be reflected by corresponding viewpoints on compliance rules [6, 18, 8]. In the context of this work,

- *asserted compliance* means that a particular partner assures to the other partners that all traces producible on its private process comply with its *assertion rules* (AR), and
- *global compliance* means that all traces virtually producible on the cross-organizational business process (i.e., the concurrent execution of all private processes) comply with all *global compliance rules* (GCR).

We developed the *extended Compliance Rule Graph* (eCRG) language to specify ARs and GCRs [19, 20]. The eCRG language is a visual language for modeling compliance rules. It not only focuses on the control flow perspective, but enables integrated support for interactions with business partners as well.[4] The elements of an eCRG may be partitioned into an *antecedence pattern* and a related *consequence pattern*. Both patterns are modeled using *occurrence* and *absence nodes*, which either express the occurrence or absence of certain events related to the execution of a particular activity or the exchange of a particular

[3] We assume $\mathcal{A} = \mathcal{A}_v \cup \mathcal{A}_h$, but do not require $\mathcal{A}_v \cap \mathcal{A}_h = \varnothing$.

[4] Note that the eCRG language also addresses the resources, data and time perspectives, but these are not relevant in the context of this paper.

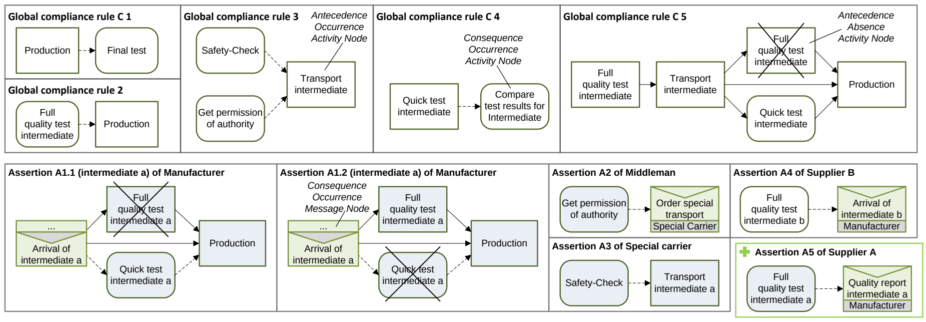

Fig. 3. Global compliance rules C1-C5 and assertions A1-A5 of running example

message (i.e., a particular interaction). In turn, eCRG edges are used to specify control flow dependencies (cf. Def. 1). An eCRG considers a process trace as *compliant*, if for each match of the *antecedence pattern* (i.e., *activation* of the eCRG), there exists at least one corresponding match of the *consequence pattern*. *Trivial compliance* refers to the absence of any activation. Fig. 3 shows the eCRGs that refer to the ARs and GCRs of the running example.

Note that our approach not depends on eCRG, but can be easily applied to other compliance rule languages as well (e.g. FCL [21]).

Definition 1 (Assertions and Global Compliance Rules).
Let $\mathcal{A}$ be the set of activities and let $\mathcal{I}$ be the set of interactions. Then: a global compliance rule or an assertion rule r is a tuple $r = (A^+, A^-, C^+, C^-, type, \mu^{A^+}, \mu^{A^-}, \mu^{C^+}, \mu^{C^-})$ with

- A^+ (A^-) *being the set of* **antecedence occurrence (absence) nodes**,
- C^+ (C^-) *being the set of* **consequence occurrence (absence) nodes**,
- $type : A^+ \cup A^- \cup C^+ \cup C^- \longrightarrow \mathcal{A} \cup \mathcal{I}$ *mapping each node to its* **activity** *or* **message (type)**,
- μ^{A^+} (μ^{A^-}) *being the* **antecedence occurrence (absence) sequence flow condition**, *and*
- μ^{C^+} (μ^{C^-}) *being the* **consequence occurrence (absence) sequence flow condition**.

Further, we define $\mathcal{R}_c$ as the **set of global compliance rules** *and $\mathcal{R}_a$ as the* **set of assertions**.

To verify compliance of an intra-organizational process, model checking can be applied, since the complete state space of a process can be determined. However, in a CBP the partners usually do not publish their *private models*. Hence, the state spaces cannot be determined and global compliance cannot be directly verified. In turn, in [8] we showed that the state space and compliance can be (over-)approximated based on the available information (i.e., the public models, the activities, and the ARs). Fig. 4 sketches this approximation using the set of (virtual) traces producible on the CBP in order to characterize the process state space: First, the set of visible traces (i.e. state space) are determined based on the available public models (a). Second, these traces are enriched by including private activities in order to (over-)approximate the behavior of the private processes (b). Third, AR violating traces are filtered out (c). Finally, global compliance is (over-)approximated based on the remaining traces (d).

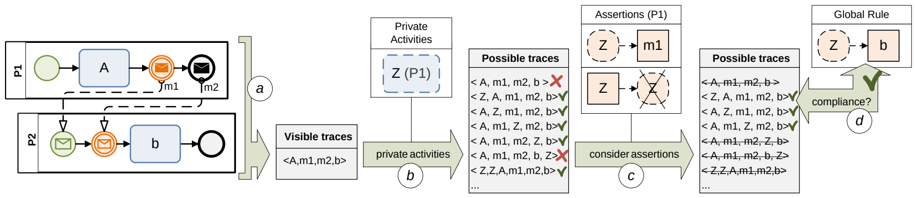

Fig. 4. Approximated global compliance of cross-organizational processes

4 Detecting Effects of Changes on Compliance

This section presents our approach for detecting those compliance rules that need to be rechecked after a change. In order to reduce *false positives*, first of all, different kinds of dependencies are analyzed between activities on one hand and compliance rules and assertions on the other. In particular, these dependencies are represented as qualified dependency graph (cf. Sect. 4.1). Finally, we ensure that there are no *false negative* results by calculating the possible transitive effects of a particular change (cf. Sect. 4.2).

4.1 Qualified Dependency Graph

In order to reduce *false positive* rechecks of global compliance rules (GCR), the elements of the latter are analyzed. Note that the elements of a GCR (AR) either express the occurrence or absence of activities. In turn, these activities either activate GCRs (ARs) or fulfill (violate) the activated GCRs (ARs). Depending on this semantics, additions (deletions) of corresponding activities have positive or negative effects on the compliance of the CBP with GCRs. By contrast, compliance with ARs is always ensured. Positive (negative) effects on assertions therefore indicate the addition (deletion) of private activities. Hence, they are relevant as well. Note that these effects can be both positive and negative (cf. GCR C5 and activity *full quality test intermediate a* (*b*)). Def. 2 formally introduces different kinds of dependencies, which are then used to express the dependencies between activities on one hand and ARs and GCRs on the other.

Definition 2 (Dependency Qualifications).
Let $\mathcal{Q} := \{\varnothing, +, -, \pm\}$ *denote the* **set of dependency qualifications**. *Thereby, a dependency can be either* $\varnothing$ **independent** *(i.e., there is no dependency),* $+$ **positive**, $-$ **negative**, *or* $\pm$ **positive&negative**. *Together with the below operations* **addition** $(+)$ *and* **multiplication** $(\cdot)$, $\mathcal{Q}$ *constructs an idempotent semiring or dioid* $(\mathcal{Q}, (+), \varnothing, (\cdot), +)$, *whereas operations* $(+)$ *and* $(\cdot)$ *are defined on* $\mathcal{Q}$ *as follows:*

(+)	Ø	-	+	±
Ø	Ø	-	+	±
-	-	-	±	±
+	+	±	+	±
±	±	±	±	±

(·)	Ø	-	+	±
Ø	Ø	Ø	Ø	Ø
-	Ø	+	-	±
+	Ø	-	+	±
±	Ø	±	±	±

Based on Def. 2, we can interpret and express the dependencies between activities and rules (i.e., ARs or GCRs) as *qualified dependency graph* (QDG).

The latter constitutes a colored and bipartite graph, whose nodes correspond to activities and rules (i.e., GCRs or ARs). Thereby, positive (solid) and negative (dashed) edges express the dependencies between activities and rules. To construct the QDG, Def. 3 utilizes the partitioning of eCRGs into occurrence and absence nodes of the antecedence and consequence pattern respectively.

Definition 3 (Qualified Dependency Graph).

A qualified dependency graph Φ *is a tuple* $\Phi = (\mathcal{A}, \mathcal{R}, \overline{d})$*, with*
- $\mathcal{A}$ *being a set of activities and* $\mathcal{R}$ *being a set of rules, and*
- $\overline{d} \subseteq (\mathcal{A} \times \mathcal{R}) \times \{-,+\}$ *being the qualified dependency relation, which is defined as follows:*

$$\overline{d} := \{(a,r,-) | \exists r \in \mathcal{R}, n \in A_r^+ : type(n) = a\} \cup \{(a,r,+) | \exists r \in \mathcal{R}, n \in A_r^- : type(n) = a\}$$
$$\cup \{(a,r,+) | \exists r \in \mathcal{R}, n \in C_r^+ : type(n) = a\} \cup \{(a,r,-) | \exists r \in \mathcal{R}, n \in C_r^- : type(n) = a\}$$

Further,
- $\Phi_a := (\mathcal{A}, \mathcal{R}_a, \overrightarrow{d})$ *is the qualified dependency graph between activities and ARs and*
- $\Phi_c := (\mathcal{A}, \mathcal{R}_c, \overrightarrow{d})$ *is the qualified dependency graph between activities and GCRs.*

Activities of the antecedence occurrence (A^+) pattern have negative effects ($-$) on the compliance of the respective rule, since additional activities of the antecedence occurrence (A^+) pattern might trigger additional activations that might be violated. In turn, additional activities of the antecedence absence (A^-) pattern might deactivate existing rule activations and, therefore, increase compliance (+). Further, additional activities of the consequence occurrence (C^+) pattern might fulfill additional activations, and, hence, increase compliance with the respective rule (+). In turn, additional activities of the consequence absence (C^-) pattern might violate of present activations ($-$). Fig. 5 combines the QDGs related to the running example. For the sake of readability, we omitted rules and activities not relevant in the given change scenario.

Note that the partitioning of the eCRG, which is utilized in Def. 3, is not a unique characteristic of the eCRG, but is enabled by other compliance rule languages in a similar way; e.g, FCL [21] distinguishes between *premises* (→antecedence) and *conclusions* (→consequence), which both can be *negated* (→absence).

4.2 Algorithms

Based on the QDGs, we introduce algorithms to determine the direct and transitive effects of CBP changes on private activities (Alg. 1) as well as their possible influence on the compliance of the CBP with GCRs (Alg. 2). These algorithms utilize the *public properties of changes*; i.e., the additions and deletions of public activities and assertions (cf. Def. 4). Fig. 6 shows the public properties of the changes applied in the context of the running example.

Definition 4 (Public Properties of Changes).
The **public properties of a CBP change** *correspond to a function* $chg : \mathcal{A}_v \cup \mathcal{R}_a \to \{\varnothing, -, +, \pm\}$ *that states whether activities of a particular type are not affected* ($\varnothing$)*, removed* ($-$)*, added* (+)*, or both (i.e. moved* ($\pm$)*).*

We distinguish between two kinds of effects, a CBP change may have on the compliance of the CBP with a particular GCR. On one hand, a GCR may

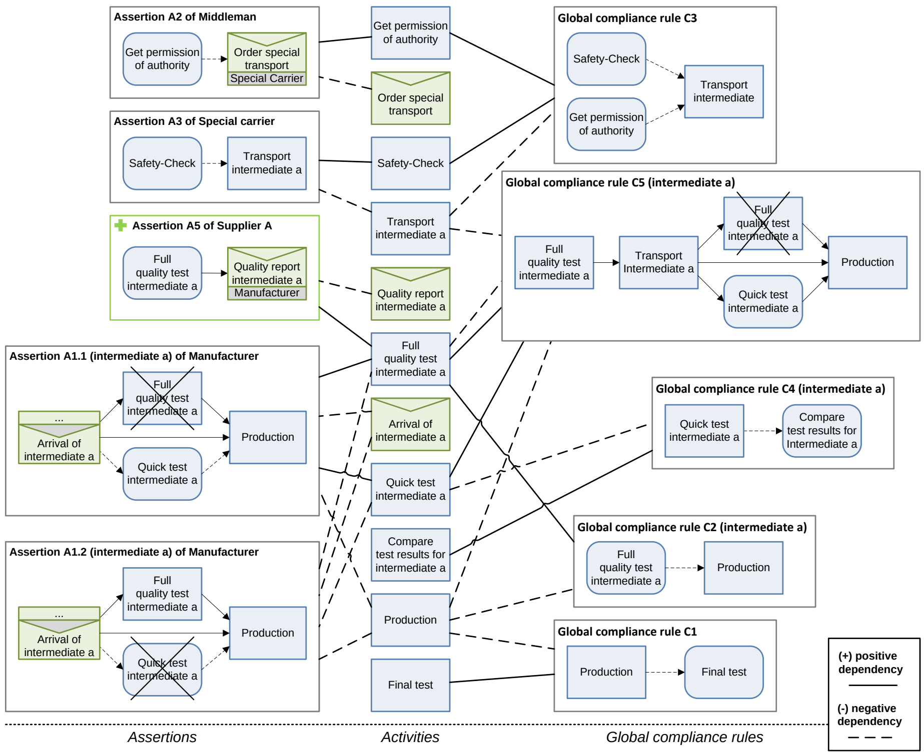

Fig. 5. Dependency graph

directly refer to activities affected by the change (i.e., added, deleted or moved). On the other, a CBP change can increase (decrease) the activations of assertions or even add (remove) assertions. In turn, the latter might then (no longer) filter out traces that violate the compliance of the CBP with GCRs. Fig. 7 illustrates how the the deletion of an activity or an assertion weakens the assertion-based filtering, so that traces can pass and violate compliance. Note that decreasing activations of assertions weakens the related filtering and, hence, always tends to decrease compliance of the CBP. In turn, increasing activations of assertions strengthens the filtering and, hence, always tends to increase compliance of the CBP. According to this, Algorithm 1 utilizes the given change and the qualified dependencies of the QDG to calculate the increasing (decreasing) effects of assertions on compliance (cf. `Lines 7-9` and `Lines 11-17`). Note that `Lines 8`

$$chg(x) := \begin{cases} -, & if\ x \text{ is activity } \textit{Full quality test of intermediate a} \quad (\delta 1) \\ +, & if\ x \text{ is activity } \textit{Quality report intermediate a} \quad (\delta 2) \\ +, & if\ x \text{ is assertion rule A5} \quad (\delta 5) \\ \varnothing, & else \end{cases}$$

Fig. 6. Public properties of the change in the running example

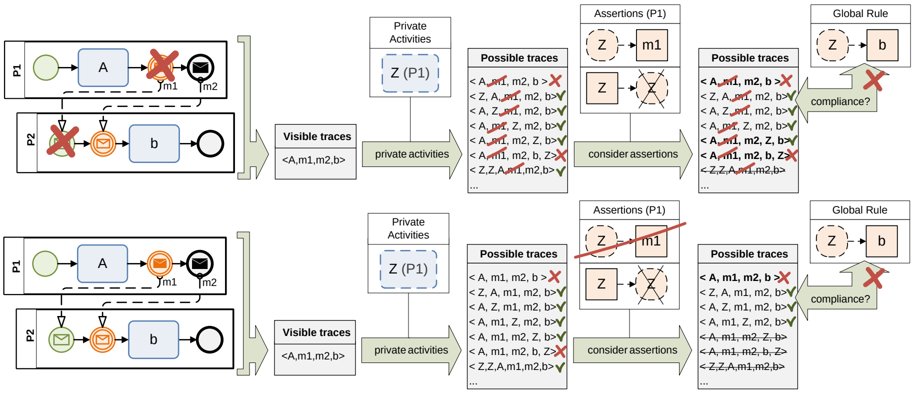

Fig. 7. Effects of local changes on the assertion-based filtering

and `16` change the semantics of qualifications (e.g., + or ±). The original qualifications expressed the additions or deletions of activities and ARs. Afterwards, in turn, the qualifications express whether ARs and activities have positive or negative effects on the global compliance of the CBP with GCRs. For public activities, however, the original semantics is preserved. Since these effects may transitively spread through private activities and ARs, `Lines 19-27` propagate them based on the QDG. As a result, Alg. 1 enriches the given change with its transitive effects on private activities (and ARs respectively).

In turn, Algorithm 2 aggregates the direct and transitive effects of CBP changes on GCRs based on the dependency relations of the QDG. However, the qualifications on dependency relations are only relevant in the context of direct effects; i.e., changed public activities (cf. `Line 6`). In turn, transitive effects are directly aggregated (cf. `Line 8`). Finally, Alg. 2 returns all possible effects of a given CBP change on compliance of the CBP with GCRs. Assuming the latter was ensured before the change, compliance of the CBP with the GCRs, which are annotated with − or ±, needs to be rechecked, e.g. by applying the approach presented in [8]. Note that our approach is not limited to this use case. Additionally, it can be applied to the opposite case; i.e., determining whether a change has the potential to heal current violations of a particular GCR.

5 Evaluation

In order to evaluate and demonstrate the technical feasibility of the approach, we implemented a proof-of-concept prototype. The latter is not only able to construct the QDG and to calculate the results of the presented algorithms, but also allows visualizing the QDG as well as listing intermediate results of the algorithms (cf. Fig. 8). Based on these intermediate outputs, we were able to enhance and optimize the algorithms. In Fig. 8, the prototype applies the approach to the presented example. In particular, it recommends rechecking 3

Algorithm 1: Transitive effects of CBP changes

Input:
- Function $chg() : \mathcal{A}_v \cup \mathcal{R}_a \rightarrow \{\varnothing, -, +, \pm\}$ specifying the initial CBP change,
- Qualified dependency graph $\Phi_a := (\mathcal{A}, \mathcal{R}_a, \overline{d})$ between activities and ARs.

```
begin
Q := ∅; //A queue for unhandled, but affected elements.
//Initialize the change effects on hidden activities with ∅:
foreach a ∈ A_h do effects(a) := ∅;
//Initialize the change effects on assertions depending on chg
//Append them to Q to handle their transitive effects:
foreach r ∈ R_a do
    effects(r) := chg(r);
    if chg(r) ≠ ∅ and r ∉ Q then Q := Q + r;
//Initialize the change effects on visible activities depending on chg:
foreach a ∈ A_v do
    if chg(a) ≠ ∅ then
        //Calculate the transitive change effects on assertions that depend on activity 'a'
        //Append them to Q to handle them:
        foreach (a, r, σ) ∈ d̄ do
            effects(r) := effects(r) − σ · chg(a);
            if r ∉ Q then Q := Q + r;
//As long as there exist unhandled elements in Q, calculate their effects:
while Q ≠ ∅ do
    n :← Q; //Remove and store the head of queue Q in variable 'n'.
    //Recalculate the effects on assertions or hidden activities depending on 'n':
    foreach (n, m, σ) or (m, n, σ) ∈ d̄ do
        if m ∈ A_h ∪ R_a then
            oldEffects := effects(m);
            effects(m) := effects(m) + effects(n);
            //Append changed elements to Q to handle them:
            if oldEffects ≠ effects(m) and m ∉ Q then Q := Q + m;
end
```

Output: Function $effects()$ specifying the transitive effects of $chg()$.

Algorithm 2: Effects of CBP changes on global compliance

Input:
- Function $effects() : \mathcal{A}_h\ (\cup\mathcal{R}_a) \rightarrow \{\varnothing, -, +, \pm\}$ specifying the transitive effects of change $chg()$,
- Function $chg() : \mathcal{A}_v \cup \mathcal{R}_a \rightarrow \{\varnothing, -, +, \pm\}$ specifying the initial changes,
- Qualified dependency graph $\Phi_c := (\mathcal{A}, \mathcal{R}_c, \overline{d})$ between Activities and GCRs.

```
begin
//Initialize the change effects on GCRs with ∅:
foreach r ∈ R_c do effectsC(r) := ∅;
foreach (a, r, σ) ∈ d̄ do
    //Recalculate the direct effects on GCRs based on visual activities:
    if a ∈ A_v then effectsC(r) = effectsC(r) + σ · chg(a);
    //Recalculate the transitive effects on GCRs based on hidden activities:
    if a ∈ A_h then effectsC(r) = effectsC(r) + effects(a);
end
```

Output: Function $effectsC()$ specifying the effects of $chg()$ on compliance.

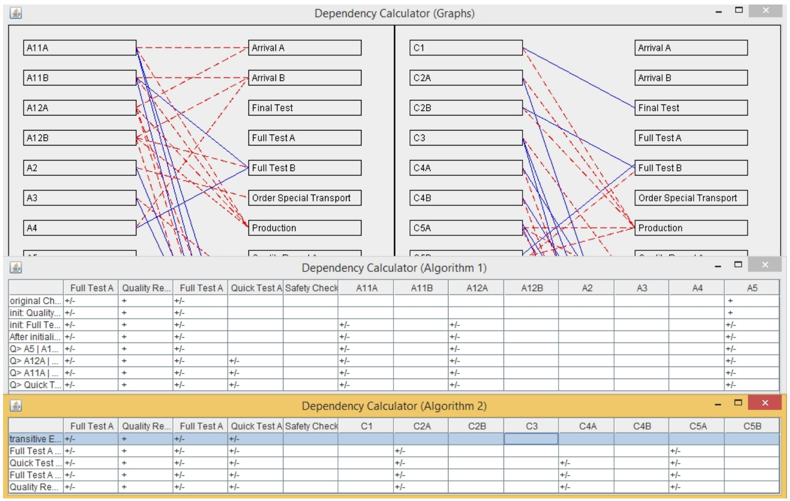

Fig. 8. Proof-of-concept prototype (Screenshots)

GCRs (i.e., C2A, C4A and C5A). Note that the the *obvious approach* would recheck all 8 GCRs, whereas the *naive* one (i.e., to recheck only directly affected GCR) fails. It rechecks C2A and C5A, but not C4A that is violated.

6 Related Work

In many domains, business processes are subject to laws, regulations, and guidelines [1]. Approaches, methods and techniques ensuring the compliance of a process with respective rules and constraints are covered under the term *business process compliance* [22]. In particular, the specification of compliance rules has been addressed by several approaches; e.g, [23, 4] provide sets of compliance patterns, whereas [2, 24, 19] introduce visual notations. Besides the formal specification of compliance rules, their integration along the process lifecycle has been discussed [3, 22, 25]. Different techniques are applied to *a priori* check the compliance of process models at design time; e.g., [2] applies *model checking*, whereas [26] relies on Mixed-Integer Programming. In turn, [27, 28, 5, 29] address *compliance monitoring* and *continous auditing* [30] at runtime. Finally, [4] discusses *a posteriori* compliance checking based on process event logs. However, so far, only little work exists addressing compliance issues in the context of cross-organizational processes (e.g., [27, 31]). Furthermore, [13] discusses compliance after changing the set of partners in such settings. However, these approaches have not taken privacy constraints into account yet [6, 7].

To remedy this drawback, we investigated *a priori* compliance checking of cross-organizational processes with respect to privacy constraints [18, 8]. This paper supplements our previous work by explicitly addressing global compliance in the context of CBP changes.

Note that few approaches deal with structural and behavioral effects of CBP changes and take privacy issues into account [9–12]. However, these approaches do not consider the effects of CBP changes on the compliance of the CBP with imposed global compliance rules.

7 Summary

Ensuring compliance with guidelines, standards and laws is crucial for both intra- and cross-organizational business processes (CBP). However, only few approaches consider compliance of CBPs taking into account that the partners do not know all parts of the CBP due to privacy reasons [6, 8]. In particular, compliance of evolving (i.e. changing) CBPs has not been sufficiently investigated yet.

To remedy this drawback, we developed algorithms that detect possible effects of CBP changes on global compliance rules. In particular, our approach limits the number of compliance checks to be repeated when changing a CBP. For this purpose, we utilized the dependencies between compliance rules, public views on partner processes, and declarative assertions provided by the partners on the behavior of their private processes. Based on these dependencies, which were represented as qualified dependency graph, we introduced two algorithms assessing the possible effects of CBP changes on the compliance of the CBP. Furthermore, we illustrated the approach along a running example and provided a proof-of-concept prototype. To the best of our knowledge, there exists no other approach ensuring semantic compliance of CBPs after changes, taking privacy constraints into account (i.e. the non-availability of information on the private elements of partner processes).

In future work, we will consider the effects of CBP changes on the compliance of running CBP instances. In particular, we will investigate, whether these instances can be migrated to new versions of the CBP, without violating compliance. Further, we will improve the approach by taking further information into account (e.g., positions of changed activities, control flow dependencies within compliance rules).

References

1. Governatori, G., Sadiq, S.: The journey to business process compliance. In: Handbook of Research on BPM. IGI Global (2009) 426–454
2. Awad, A., et al.: Efficient compliance checking using BPMN-Q and temporal logic. In: BPM'08. (2008) 326–341
3. Ly, L.T., et al.: Integration and verification of semantic constraints in adaptive process management systems. Data & Knowl Eng **64**(1) (2008) 3–23
4. Ramezani Taghiabadi, E., et al.: Diagnostic information for compliance checking of temporal compliance requirements. In: CAiSE'13, Springer (2013) 304–320
5. Maggi, F.M., Francescomarino, C.D., Dumas, M., Ghidini, C.: Predictive monitoring of business processes. In: CAiSE'14, Springer (2014) 457–472
6. Knuplesch, D., et al.: Towards compliance of cross-organizational processes and their changes. In: BPM'12 Workshops. (2013) 649–661

7. Fdhila, W., Knuplesch, D., Rinderle-Ma, S., Reichert, M.: Change and compliance in collaborative processes. In: SCC'15
8. Knuplesch, D., et al.: Ensuring compliance of distributed and collaborative workflows. In: CollaborateCom'13, IEEE (2013) 133–142
9. Fdhila, W., et al.: Dealing with change in process choreographies: Design and implementation of propagation algorithms. Inf Sys **49** (2015) 1 – 24
10. 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
11. Rinderle, S., Wombacher, A., Reichert, M.: Evolution of process choreographies in DYCHOR. In: CoopIS'06. (2006) 273–290
12. Mafazi, S., Grossmann, G., Mayer, W., Stumptner, M.: On-the-fly change propagation for the co-evolution of business processes. In: OTM'13. (2013) 75–93
13. Comuzzi, M.: Aligning monitoring and compliance requirements in evolving business networks. In: OTM'14. (2014) 166–183
14. Knuplesch, D., et al.: On enabling data-aware compliance checking of business process models. In: ER'10. (2010) 332–346
15. van der Aalst, W.M.P., et al.: Multiparty contracts: Agreeing and implementing interorganizational processes. Comp J **53**(1) (2010) 90–106
16. Decker, G., Weske, M.: Behavioral consistency for B2B process integration. In: CAiSE'07. (2007) 81–95
17. Rouached, M., et al.: Web services compositions modelling and choreographies analysis. Int. J. Web Service Res. **7**(2) (2010) 87–110
18. Knuplesch, D., et al.: On enabling compliance of cross-organizational business processes. In: BPM'13. (2013) 146–154
19. Knuplesch, D., et al.: Visual modeling of business process compliance rules with the support of multiple perspectives. In: ER'2013. (2013) 106–120
20. Semmelrodt, F., Knuplesch, D., Reichert, M.: Modeling the resource perspective of business process compliance rules with the extended compliance rule graph. In: BPMDS'14, Springer (2014) 48–63
21. Governatori, G., et al.: Detecting regulatory compliance for business process models through semantic annotations. In: BPM'08 Workshops, Springer (2009) 5–17
22. Knuplesch, D., Reichert, M.: Ensuring business process compliance along the process life cycle. Technical Report 2011-06, Ulm University (2011)
23. Turetken, O., et al.: Capturing compliance requirements: A pattern-based approach. IEEE Software (2012) 29–36
24. Ly, L.T., et al.: Design and verification of instantiable compliance rule graphs in process-aware information systems. In: CAiSE'10. (2010) 9–23
25. Koetter, F., et al.: Integrating compliance requirements across business and it. In: EDOC'14. (2014)
26. Kumar, A., et al.: Flexible process compliance with semantic constraints using mixed-integer programming. INFORMS J on Comp **25**(3) (2013) 543–559
27. Berry, A., Milosevic, Z.: Extending choreography with business contract constraints. Coop Inf Sys **14**(2-3) (2005) 131–179
28. Ly, L.T., et al.: A framework for the systematic comparison and evaluation of compliance monitoring approaches. In: EDOC'13, IEEE (2013) 7–16
29. Knuplesch, D., Reichert, M., Kumar, A.: Visually monitoring multiple perspectives of business process compliance. In: BPM'15
30. Alles, M., Kogan, A., Vasarhelyi, M.: Putting continuous auditing theory into practice: Lessons from two pilot implementations. Inf Sys **22**(2) (2008) 195–214
31. Governatori, G., et al.: Compliance checking between business processes and business contracts. In: EDOC'06. (2006) 221–232