An Indexing Technique for Compliance Checking and Maintenance in Large Process and Rule Repositories

Stefanie Rinderle-Ma[*,a], Sonja Kabicher-Fuchs[a]

[a] University of Vienna, Faculty of Computer Science, Research Group Workflow Systems and Technology, Vienna, Austria

Abstract. *Business process compliance has become a crucial challenge for enterprises throughout different domains. However, little thought has been spent on the management and verification of compliance rules in large process repositories so far, even though several case studies show, that the amount of business processes can reach from a small set to hundreds of business processes being subject to several hundreds of compliance rules. In this paper we present activity-oriented indexing techniques for efficient compliance checking which are particularly applicable in process and rule repositories where no a-priori knowledge, e. g. based on policies, is available. Different applications beyond compliance checking are discussed such as process similarity notions or maintenance issues. The effects of applying indexing on the effort for compliance checks are discussed along with further aspects such as maintenance of process model and compliance rule repositories. Finally, a case study from the higher education domain in Austria is provided. The presented techniques constitute a first step towards a cost- and effort-aware management of large business process and compliance rule repositories.*

Keywords. Business Process Modelling • Business Process Compliance • Rule-Based Compliance Checking

Communicated by K. Turowski and S. Strecker. Received 2013-03-12. Accepted after 2 revisions on 2015-11-25.

1 Introduction

Nowadays, many organizations integrate business process management (BPM) in order to manage their daily business in an efficient way. Business process compliance (BPC) has become a crucial challenge for enterprises nowadays, since they have to prove that their business processes are executed in accordance with certain regulations, policies, or controls (Awad et al. 2008; Ghose and Koliadis 2007; Goedertier and Vanthienen 2006b; Governatori et al. 2006; Karagiannis 2008; Ly et al. 2012b). The associated compliance verification mechanisms are often accomplished manually by audits (Rinderle-Ma et al. 2008; van der Aalst et al. 2010) that might be complex and costly, specifically at the presence of large process model and compliance rule repositories. Process model repositories capture process models and model variants describing a certain process type such as an order process.

The IT-based support for checking business process compliance has gained significant momentum in research (Ghose and Koliadis 2007; Ly et al. 2008). Several approaches to design, integrate, and verify compliance rules over business processes in an at least semi-automatic manner have been proposed (e. g. Ly et al. 2008). Since these approaches are often based on model checking techniques, considerations on how to reduce the effort for compliance checking have been made (Awad et al. 2008; Hoffmann et al. 2012). However, no particular thought has been spent on the management and verification of compliance rules in large process repositories, even though several case studies show, that the amount of

* Corresponding author.
E-mail. stefanie.rinderle-ma@univie.ac.at
Note: This article revises and extends an earlier conference publication, cf. Rinderle-Ma et al. (2012).

business processes can reach from a small set to hundreds of business processes being subject to several hundred compliance rules (Valkenburg 2010). Without any further knowledge such as policy-based grouping of compliance rules, this means to verify all compliance rules for each business process in the repository. This might turn out very costly, since compliance verification itself—depending on the particular technique—is a non-trivial task (Hoffmann et al. 2012).

1.1 Research questions

Hence, an important question is how to support compliance checks as well as maintenance of large-scale process model and compliance rule repositories in an efficient way. The basic idea is to define index structures on the process model and/or compliance rule repositories as typically done in the database field (e. g. Guttman 1984). Monitoring the compliance of process instances (e. g. Ly et al. 2015) is outside the scope of this paper. This overarching question can be broken down into the following sub-questions which will be addressed in this paper:

1. How to find out which compliance rules are associated to which process models?
2. How to create index structures for compliance rule as well as process model collections?
3. How to cope with compliance rules that refer to several process models?
4. How to evaluate the performance gain of applying indexing techniques?
5. How to utilize index structures in order to determine semantic similarities between process models?
6. How to deal with changes of the compliance rule collection (maintenance issues)?

Research questions 1–3 refer to the elaboration of indexing techniques. Basically, it is possible to cluster compliance rules for each process model or vice versa. Possibly, clusters might be also aggregated if, for example, a set of compliance rules refers to different process models. Question 4 deals with the evaluation of applying indexing in terms of performance considerations. Question 5 asks how indexing information can be exploited in order to determine similarities between process models. Process model similarity constitutes an important prerequisite for maintaining process model repositories: similar process models or fragments can be detected and possibly merged together. In general, maintenance of process model repositories refers to the management and refactoring of different process model versions and variants (Weber et al. 2011).

So far, mostly structural or behavioral similarity metrics between process models have been considered (Becker et al. 2012; Dijkman et al. 2011). The question is whether the information that two process models are associated to the same or a similar set of compliance rules can be used for establishing a semantic similarity metric between process models.

A promising technique for associating process models and compliance rules and to subsequently reduce the effort for compliance checks is the creation of an *index structure* on the compliance rule and process model repositories. One approach to build an index structure is based on clustering; either the compliance rules can be clustered along the process models or vice versa. Note that the focus is on process model repositories, i. e. repositories that capture process models and model variants describing a certain process type such as an order process. Monitoring the compliance of process instances (Ly et al. 2015) is not considered yet.

Index structures support query processing in database applications such as data warehouse systems (Chaudhuri and Dayal 1997). Contrary, in the BPM area, indexing techniques have been applied only in selected scenarios, for example, by clustering process instances along their execution states in order to support their efficient migration to change process schemas (Kreher and Reichert 2010). For large process model collections, indexing has been investigated to support queries (Awad and Sakr 2010) on the process model repositories (Jin et al. 2010).

Altogether, the novelty of the presented technique can be seen in the application of well-known principles to the problem at hand and in the resulting conclusions, for example, whether the index structures can be used for supporting maintenance of process model and compliance rule collections or for new notions of process similarity.

The last Research Question 6 refers to maintenance of compliance rule collections, when, for example, adding new rules to the collection.

1.2 Contribution

This paper presents activity-oriented indexing techniques for compliance rules and process models. The basic idea is illustrated in Fig. 1. Instead of checking all compliance rules for all process models, rules and process models are clustered along their association to each other. The effect is that only those compliance rules have to be checked for a particular process model that are actually associated with this model.

In particular, three algorithms for indexing compliance rules along process models and vice versa are provided. The applicability of both approaches is discussed based on qualitative as well as quantitative considerations. Further on, it is shown how clusters can be further optimized by aggregating them along their similarity. All presented indexing techniques can be applied independently of any a-priori knowledge such as policies associated to compliance rules and independently of any process meta model. In addition, we investigate whether the knowledge on existing clusters can be exploited in order to establish a notion of semantic similarity between process models. We discuss the effectiveness of the different techniques based on performance considerations as well as on their effects on compliance rule consistency and maintenance. Exemplary, for conflict-freeness of the compliance rule base we introduce a theorem that reduces the number of necessary consistency checks. The techniques are illustrated based on the IT Baseline Security use case (Federal Agency for Security in IT 2006) as well as discussed considering different aspects such as transferability and applicability. We also provide a real-world case study from the area of higher education in Austria. Altogether, the presented techniques constitute a first step towards a cost- and effort-aware management of large business process and compliance rule repositories.

This paper constitutes a substantial extension of Rinderle-Ma et al. (2012). In detail, this paper provides a deeper discussion on the different indexing techniques including a discussion on updating the index structure as a consequence of process model or compliance rule repository changes. Additional contributions are provided by process model similarity notions based on indexing structures and a real-world case study. Further on, discussions on implementation issues and transferability of the proposed technique to other compliance rule notions and process meta models were added.

The paper starts with discussing related approaches in Sect. 2. In Sect. 3, the research methodology as well as the main assumptions are outlined. Sect. 4 summarises necessary background information. In Sect. 5, activity-oriented indexing techniques are provided and discussed ($\mapsto$ Research Questions 1 and 2). Sect. 6 introduces a process similarity notion based on indexing ($\mapsto$ Research Questions 3 and 5). Sect. 7 discusses qualitative as well as quantitative factors that influence the effectiveness of the proposed solution ($\mapsto$ Research Question 4). Sect. 8 provides the case study ($\mapsto$ Research Question 4 and 5). Sect. 9 closes with a summary of the presented results as well as an outlook on future research work in this area.

2 Related Work

As mentioned in the introduction, indexing structures have a long tradition in the database research and serve as a powerful mechanism for accelerating query execution. Examples comprise tree structures such as the R* tree (see Guttman 1984). Further on, the basic idea of the proposed index to split the overall set of compliance rules and process models into sub sets based on which the compliance checks are conducted. The novelty

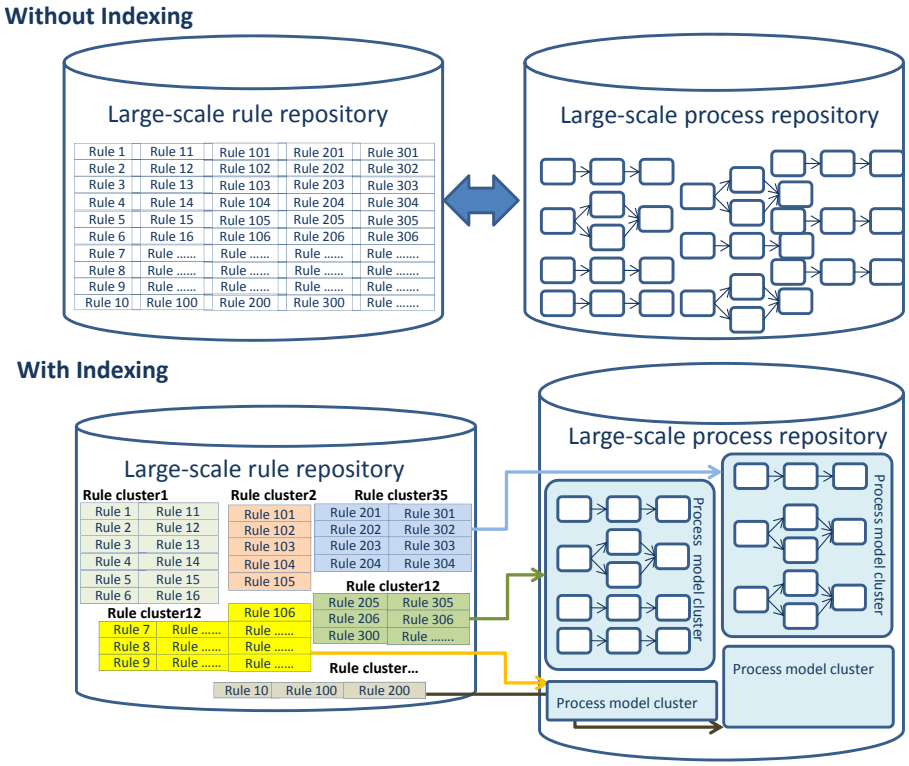

Figure 1: Indexing for compliance checking in large-scale process and rule repositories.

of this approach therefore is not to be found in the basic techniques, but their application to a novel question, i. e. using indexing techniques for accelerating compliance checks and can be also used for further questions such as maintenance of compliance rule collections and possibly for new process similarity notions.

Another relation to the database area is the relation between integrity rules and compliance rules. As discussed in Goedertier and Vanthienen (2006a), integrity rules are one type of compliance rules that specify pre-conditions for the activation of process activities. This is an interesting aspect for future work as the pre-conditions might further the restriction of the set of compliance rules to be checked, in particular, if also process instances are monitored with respect to compliance during runtime.

Approaches related to the work presented in this paper span from business process compliance over the management of large process collections to the specification and configuration of process models. Hence we will discuss these issues in the aforementioned order in the following.

2.1 Business process compliance

Since the indexing techniques presented in this paper do not necessitate any particular compliance checking approach, it can be combined with existing process model verification approaches such as Awad et al. (2008), Ghose and Koliadis (2007), Goedertier and Vanthienen (2006b), Governatori et al. (2006), Knuplesch et al. (2010), Liu et al. (2007) and Sadiq et al. (2007). Our approach can further be combined with approaches to manage compliance rules and their relations to process models such as Namiri (2008) and Namiri and Stojanovic (2008). Specifically, if a-priori knowledge based on, for example, policy-based compliance rule specification is available, this knowledge can be exploited in order to further develop semantic similarity notions for process models.

2.2 Querying and managing process collections

For querying large process repositories, query languages on process models have been developed (Awad and Sakr 2010; Francescomarino and Tonella 2008; Hornung et al. 2007). BPMN-Q proposed by Awad and Sakr (2010), for example, is a graph-based language for querying process models. A process model will be contained in the result set of a BPMN-Q query if the query graph matches the process graph. In the context of compliance checking, BPMN-Q can be used to query process model repositories for those process models containing activities or structures that are relevant to a compliance rule (see Awad et al. 2008). Hence, finding associated process models for compliance rules as necessary for indexing can be supported by such query languages, particularly in combination with platforms for large process repositories, for example, APROMORE (see La Rosa et al. 2011).

Another current stream of research deals with the efficient evaluation of queries on process model repositories. For this, indexing techniques on process models have been developed (Jin et al. 2010). As stated above, these indexing techniques can be applied to support the efficient finding of associations between process models and compliance rules. However, approaches for indexing and indexing process models for compliance checking as well as for the compliance rules themselves have not been addressed so far.

2.3 Process similarity

Defining and determining similarities between process models has been identified as an important issue in literature. Similarity notions and queries can be used for consolidation of process models within repositories, for example, when similar process models or fragments are merged. Dijkman et al. (2011) distinguishes similarity of process model elements, structural similarity, and behavioral similarity where the later two notions refer to process models. A semantic similarity notion is provided for process model elements 'based on equivalence between the words they consist of' as stated by Dijkman et al. (2011). The work presented by Becker et al. (2012) utilizes both, structural and behavorial similarity for determining similarity between activities. Approaches for activity similarity can be utilized for establishing the connection between compliance rules and process models.

Similarity notions and queries can be used for consolidation of process models within repositories, for example, when similar process models or fragments are merged. In a current study on process similarity provided by Dijkman et al. (2011), the following similarity notions are distinguished: similarity of process model elements, structural similarity, and behavioral similarity where the later two notions refer to process models.

The similarity metric introduced in Sect. 6 is activity-oriented, i. e. similarity metrics as proposed in literature can be utilized for activity matching (e. g. Becker et al. 2012). The distinction to existing metrics can be illustrated as follows: existing metrics compare process models in an imperative manner, i. e. the activities and structures should match as exactly as possible. The metrics proposed in the following, employs a declarative comparison, i. e. two (imperative) process models are similar if they possess a more or less common declarative description. A declarative description is based on some kind of rules that set out what can be done in a process model rather than strictly specifying what must be done (as for imperative models) (Weber et al. 2009).

Hence, the proposed metrics can be seen as a step towards a notion for semantic similarity between process models based on exploiting the fact which compliance rules are associated with the process models. This step is somehow a bottom-up derivation of the top-down policy-based modeling of compliance rules provided by Namiri and Stojanovic (2008), i. e. instead of grouping process models by policies imposed on them (constituting some kind of semantic similarity), we show how the indexing of compliance rules along process models can lead to semantic similarity metric that might result in deriving policies or compliance

rule packages. As already mentioned, the approach proposed in this paper can be used without any a-priori knowledge.

Another way to determine process similarity is based on change operations as proposed by Li et al. (2008). Specifically, the delta between two process models can be expressed by the number of high-level process change operations that transform one process model into the other. How the change operations can be determined is shown in Li et al. (2008). Also related to process similarity is process conformance (Rozinat and van der Aalst 2008) where it is measured how well process models that have been mined from process event logs conform to the corresponding 'original' process models. Note that this technique can be also applied for compliance checking or auditing (see van der Aalst et al. 2010). However, semantic similarity between process models with respect to the number and kind of compliance rules that they have in common has not been considered yet. The metrics for semantic similarity as provided in this paper could be used for, for example, the refactoring of process model collections as well, i. e. the metrics could be utilized for process merging as well.

2.4 Process configuration

The configuration of reference or baseline processes has been identified as major challenge in practice (Gottschalk et al. 2008). Companies can buy or adopt reference processes for different domains and application scenarios and configure them to their requirements. Basically, there are different possibilities to specify configuration of a reference process: by either using reference process modeling approaches such as configurable EPCs (La Rosa et al. 2007; Rosemann and van der Aalst 2007) or by applying pre-defined change operations on the reference process (Hallerbach et al. 2010). In the latter variant, process configuration results in different variants of the reference process. The reference process together with its variants is referred to as process family (Hallerbach et al. 2010).

2.5 Representation of process structures

In Kreher and Reichert (2010), different representations for process structures such as process models, process instances, and process changes have been presented. For efficiently migrating process instances to a new process schema version, indexing techniques based on process instance states have been provided. For the efficient representation of process changes, the DeltaLayer approach has been presented and implemented within the ADEPT prototype for an adaptive process management system (Rinderle et al. 2006). Its applicability to other application scenarios such as data warehousing have been discussed in Rinderle et al. (2007). All these approaches, however, cannot be directly transferred to the problem at hand. However, the principle techniques can be of inspiration for further development.

3 Research Methodology and Assumptions

The methodology followed in this paper bases on design research (see Peffers et al. 2007, Fig. 1):

1. *Problem identification:* The problem is to find an indexing structure for (large) process model and compliance rule repositories. The relevance of this problem is motivated by practical studies (e. g. Valkenburg 2010) as well as literature (e. g. Awad et al. 2008). The problem identification includes the specification of assumptions made to scope the subsequent elaboration of artifacts.
2. *Objectives of solution:* The main objective is the reduction of necessary compliance checks for business process models.
3. *Design and development:* Respecting the assumptions, algorithms (artifacts) for indexing compliance rules over process models as well as a similarity metrics (artifact) are elaborated in a systematic and comprehensive way, all possible combinations for constructing clusters for a set of process models and a set of compliance rules are respected. Precisely, one algorithm considers the clustering of compliance rules for process models. Another algorithm clusters

process models along the compliance rules. In addition, an algorithm is developed that aggregates similar clusters. A rule for deciding which algorithm to apply in which situation is provided. As an additional artifact, a similarity metric for process models based on indexing is suggested.

4. *Demonstration:* The artifacts or algorithms respectively are demonstrated based on use case IT baseline security (Federal Agency for Security in IT 2006).
5. *Evaluation:* The approach is evaluated based on a case study in the higher education domain along the following questions:
   - Does the index structure reduce the effort for compliance checking in terms of numbers of necessary compliance checks?
   - Can clusters be aggregated based on similarity between process models? Does this result in further reduction of the number of compliance checks?
   - Is the application of the proposed similarity metrics useful? A case study was chosen to demonstrate the applicability of the approach in a realistic setting. Generalization might be restricted from one case study (Hays 2004); hence, it is planned to conduct further case studies in future research.

   Specifically, the effort for checking compliance for the case study scenario is determined in terms of number of compliance checks at first. Following the decision rule provided in Step 2, the algorithm for clustering the rules along the process models is applied and the resulting effort reduction is determined. The proposed similarity metrics is applied to the scenario and yields the input for the follow-up aggregation.
6. *Communication* is fostered by the publication of this article.

Scoping the problem formulation requires a set of assumptions:

- This paper assumes a co-existence of process models and compliance rules. Literature (e. g. Ly et al. 2010b) as well as experience, for example, in the higher education domain (Ly et al. 2012a) show that assuming such hybrid scenarios of process models and compliance rules is realistic. Work on business rules claims that in practice business (and hence compliance) rules should be modeled and maintained separately from the business processes (Halle and Goldberg 2009).
- Large collections of process models and compliance rules (Valkenburg 2010).
- Compliance rules as defined in Rinderle-Ma and Mangler (2011) and Ly et al. (2015), i. e. it exists at least one process activity in the scope of each rule; the rule might refer to further process aspects such as data, time, and resources. Further on, compliance rules might be applied at different times of the process life cycle, i. e. design time, runtime, and change time.
- No a-priori knowledge on
  - process meta models.
  - compliance rule grouping for process models (such as policy-based compliance rule modeling as suggested by Namiri and Stojanovic (2008)).

The latter is motivated by the fact that companies own different process models and have to follow a possibly large number of compliance rules. Within the resulting repositories, often, compliance rules are not specified exactly for the process models they refer to, but are rather extracted from regulatory packages such as BASEL II (Basler Ausschuss für Bankenaufsicht 2006) and then 'arbitrarily' imposed on the process models. Then the question is how to determine which compliance rules are associated to which process models. Particularly, it cannot be assumed that there is a 1:1 association between process model and compliance rule, i. e. compliance rules might refer to different process models and process models might be subject to different compliance rules (Namiri and Stojanovic 2008).

4 Background Information

The developed indexing structure should not be confined to a certain process meta model or language.

Hence, in the following, process models are defined similar to control flow graphs (see Allen 1970), i. e. they are based on the set of activities N and set of edges E they consist of. A process model P is defined as $P := (N, E)$. To each node $n \in N$ either an activity type AT from the domain of interest $\mathcal{A}$ or a connector type $CT \in \{\text{ANDSplit, ANDJoin, XORSplit, XORJoin}\}$ is assigned to. An edge e is defined as tuple $e := (n_1, n_2)$ where $n_1, n_2 \in N$.[1]

Following this definition, a node reflects either an activity of a certain type or a connector. The proposed techniques can be directly applied to activity-oriented process meta models such as BPMN models or Activity Nets. State and activity-based process meta models such as Petri nets or Event-driven Process Chains can be also used for activity-oriented indexing when restricting considerations to transitions or functions respectively.

Note that in a first step, we abstract from data flow and temporal issues. A discussion on the effects of considering data and temporal information when building index structures can be found in Sect. 7.

For illustration of the following notions and the proposed techniques, we use the 'IT baseline security' example (see Federal Agency for Security in IT 2006). Fig. 2 depicts a fictive company's business process repository including six business processes that refer to password protection (P1), screen lock protection (P2), protection against internet services (P3), malware scan of the database (P4), malware scan of outgoing data (P5), and malware scan of incoming data (P6).

After defining the process models it is necessary to provide a presentation for the compliance rules as well.

[1] CT might be extended by further connector types such as $ORSplit$.

This paper adopts Compliance Rule Graphs (CRGs) as proposed in Ly et al. (2010b) due to the following reasons:

- Pattern-based definition of compliance rules enables determination of which process activities are referred to by the compliance rule, specifically, which process activities trigger the rule, i. e. are 'responsible' for its activation.
- Graph-based visualization of compliance rules.
- CRGs have been successfully applied in different case studies, e. g. in the higher education domain (see Ly et al. 2012a).
- Pattern-based comparison with other compliance rule notions supports that most existing compliance rules can be expressed using CRGs.
- Prototypical implementation (see, e. g. Ly et al. 2010a).

An evaluation of the CRG formalism with respect to its expressiveness and in comparison with other existing formalisms, e. g. based on Linear Temporal Logic (LTL) can be found in Ly et al. (2015). Sect. 7 provides a discussion of transferability to other formalisms.

CRGs can be defined in a set-based manner as follows (see Rinderle-Ma et al. 2012):

Definition 1 (Compliance Rule Graph CRG) A compliance rule graph is a 7-tuple $R = (N_A, N_C, E_A, E_C, E_{AC}, nt, p)$ where:

- N_A: set of nodes of the antecedent graph of R
- N_C: set of nodes of the consequence graph of R
- E_A: set of directed edges connecting nodes of N_A
- E_C: set of directed edges connecting nodes of N_C
- E_{AC}: set of directed edges connecting nodes of the antecedent and the consequence graph of R
- $nt : N_A \cup N_C \rightarrow \{ANTEOCC, ANTEABS, CONSOCC, CONSABS\}$: function assigning a node type to the nodes of R
- p: function assigning a set of properties (e. g. activity type) to each node of R

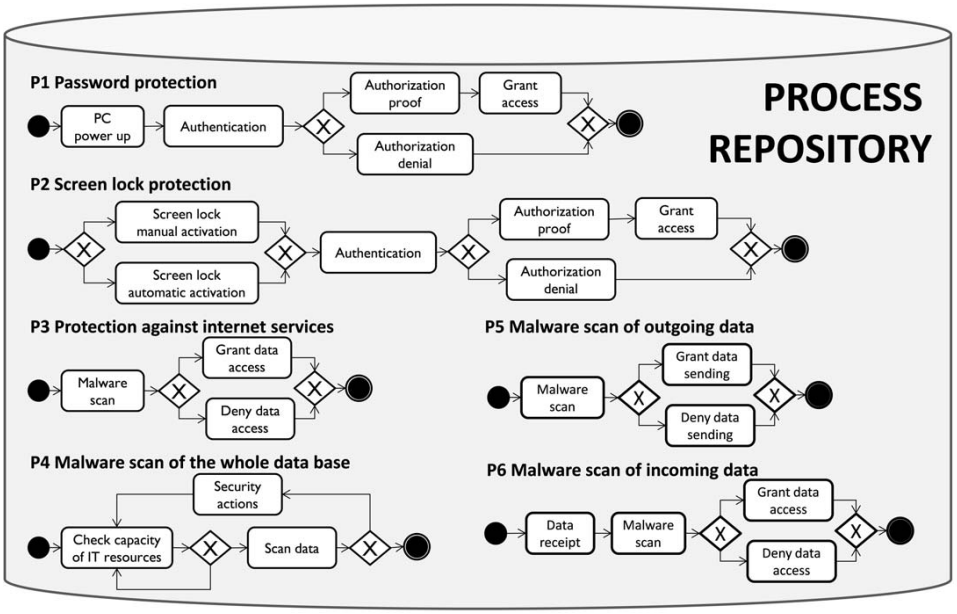

Figure 2: IT Baseline Security: Process Models (in BPMN Notation), taken from Rinderle-Ma et al. (2012).

A CRG comprises an antecedent pattern (reflecting the process activities that trigger the compliance rule) and a consequence pattern. Antecedent and consequence patterns consist of occurrence ($ANTOCC$ and $CONSOCC$) and absence nodes ($ANTEABS$ and $CONSABS$). Occurrence nodes refer to the presence of certain activities during process execution whereas absence nodes refer to the absence of certain activities respectively.

In Fig. 3, for example, compliance rule `R1` is triggered by occurrence of activity `PC Power up` within the process model of interest. It can be concluded that the process activities contained within the antecedent pattern decide whether a compliance rule refers to a process model or not. As the CRGs as well as the process model definitions are set-based, checking whether the antecedent pattern refers to at least one activity contained in a process model becomes very simple and constitutes the main idea of the following indexing technique. Antecedent patterns might also be empty as for, e. g. CRGs `R7` and `R8` in Fig. 3. This means that a compliance rule is triggered for every process. For `R7`, for example, this means that the security change setup must not be changed for any of the processes in the repository. By defining more complex antecedent and occurrence patterns, certain orders between activity occurrences or absences within the process models can be expressed, for example, that there must be no malware scan between data receipt and data access. As a last example, take CRG `R6` expressing that if activity `Data receipt` occurs within a process, activity `Malware scan` has to be conducted immediately after (cf. Fig. 3). Note that 'immediately after' does not refer to a notion of time, but expresses in this context that no other activity must be executed in between `Data receipt` and `Malware scan`.

5 Indexing Techniques for Compliance Rule and Process Model Collections

Without any further knowledge on compliance rule and process model collections, compliance rule and process model collections mostly co-exist, i. e. without any established relation between compliance rules and process models. This situation is sketched in Fig. 4, Initial Situation. Consequently, for verifying compliance of a process model, it must be verified against all rules in the compliance

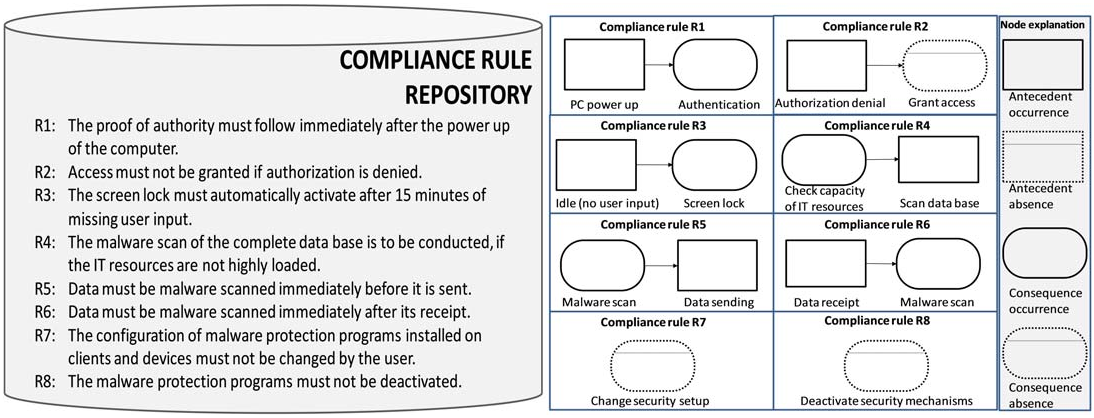

Figure 3: Use Case IT Baseline Security – The Compliance Rule Repository (Left) and the Compliance Rule Graphs (Right), taken from Rinderle-Ma et al. (2012).

rule collection. The effort can be estimated as discussed in Rinderle-Ma et al. (2012):

$$O(|C| * |\mathcal{P}| * CE_{max})$$

for

- set of process models $\mathcal{P}$
- set of compliance rules C
- maximum compliance checking effort CE_{max} $\forall P \in \mathcal{P}, \forall C \in C$; CE_{max} typically depends on the compliance checking technique, but might become quite complex. Many compliance checking techniques such as model checking depend on exploring the state space of a process (without further optimization) which is 'exponential in the size of the process' as stated by Hoffmann et al. (2012).

It is questionable whether in all cases, process models are subject to all rules in a compliance rule collection. Looking at real-world scenarios such as the case study presented in Sect. 8, process models are rather subject to subsets of the compliance rule collection, e. g. they obey a certain law or regulatory package. If the connection between process models and compliance rules is known beforehand, approaches such as proposed by Namiri and Stojanovic (2008) can be applied.

In this approach, it is assumed that there is no a priori knowledge on relations between compliance rule and process model repositories at hand. Instead, it works in an activity-oriented way by comparing the process activities referred to by the compliance rule and the process activities present in a process model. More specifically, a process model is considered as being subject to a compliance rule if the antecedent pattern of the compliance rule (cf. Def. 1) refers to at least one of the activities contained within the process. In this case, the indexing techniques either cluster compliance rules that refer to the same process model (see Fig. 4, Scenario 1), or cluster process models that are subject to the same set of compliance rules (Scenario 2), or cluster compliance rules and process models at the same time (Scenario 3).

Scenario 1: Activity-oriented compliance rule indexing determines all compliance rules that are to be checked for a particular process model.

According to Def. 1 a compliance rule is triggered over a process model, if the antecedent pattern of the compliance rule is potentially activated. This holds true if all activities associated with antecedent occurrence nodes of a compliance rule are contained in a process model. In general, this criterion can be used for optimization of compliance checks, e. g. as pre-selection before applying model-checking based techniques. Compliance rules that are not associated with any

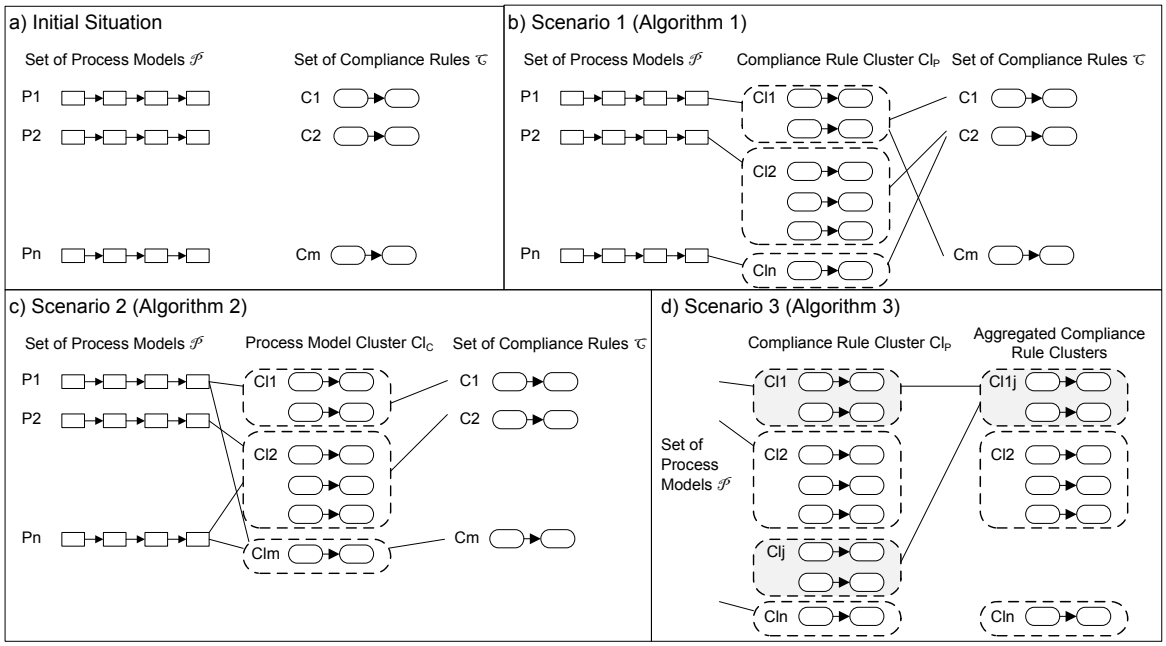

Figure 4: Basic Indexing Scenarios, taken from Rinderle-Ma et al. (2012).

process model are collected in complementary cluster Cl_{comp}. Based on the set-oriented definition of compliance rules and process models this can be determined as described by Alg. 1.

Algorithm 1 Activity-oriented compliance rule indexing.

Require: $\mathcal{P}, C$
Ensure: $Cl_P := \emptyset \forall P \in \mathcal{P}$, $\mathrm{Cl}_{comp} := \emptyset$
  for all $P = (N, E) \in \mathcal{P}$ **do**
    for all $C = (N_A, N_C, E_A, E_C, E_{AC}, nt, p) \in C$ **do**
      if $(\{n \in N_A | nt(n) = ANTEOCC\} = \emptyset) \vee (\{n \in N_A | nt(n) = ANTEOCC\} \subset N)$ **:=Cond then**
        $Cl_P := Cl_P \cup \{C\}$
      end if
    end for
  end for
  for all $C \in (C \setminus \bigcup_P Cl_P)$ **do**
    $Cl_{comp} := Cl_{comp} \cup \{C\}$
  end for
  return Cl_P, Cl_{comp}

Applying Alg. 1 to our use case results in the clusters depicted in Fig. 5. Note that compliance rules R7 and R8 are contained within every cluster since their antecedent pattern is empty and thus they are activated for every process model. As there are no compliance rules that do not refer to any process model, the complementary cluster Cl_{comp} is empty. The number of necessary compliance checks is reduced from 48 to 19.

The complexity of Alg. 1 is $O(|\mathcal{P}| * |C|)$ which has to be considered as initial effort for indexing, i. e. the effort typically occurs once. The effort for compliance checking can be determined as

$$O(\Sigma_P |Cl_P| * CE_{max}) \leq O(|\mathcal{P}| * |C| * CE_{max})$$

This means that each process model has to be checked for the compliance rules contained within the associated cluster. Based on the degree of the indexing the reduction in effort might be significant. In the worst case, no indexing is achieved, i. e. all compliance rules refer to all process models. In this case the effort for compliance checking remains the same as the effort without applying indexing techniques. When comparing effort for

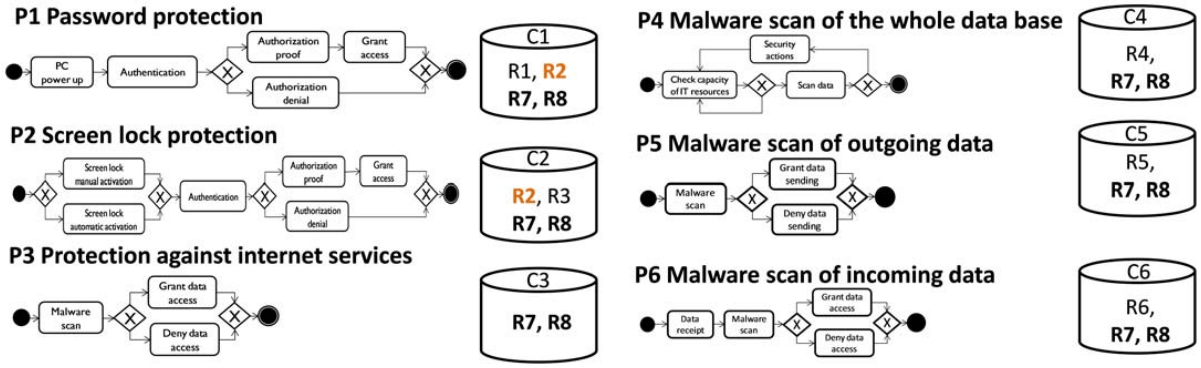

Figure 5: Use Case IT Baseline Security – Activity-Oriented Compliance Rule indexing, taken from Rinderle-Ma et al. (2012).

compliance checking and effort for building up the indexing we obtain the following conclusion:

$$O(|C| * |\mathcal{P}|) + O(\Sigma_P |Cl_P| * CE_{max}) \leq O(|C| * |\mathcal{P}| * CE_{max})$$

The effect of indexing on maintaining compliance rule and process model repositories will be discussed in Sect. 7.

Scenario 2: Compliance checking with process model indexing can be conducted based on Alg. 2. Process models are clustered for each compliance rule in C resulting in clusters $Cl_C \forall C \in C$. Again the membership within a cluster can be determined by evaluating condition **Cond** set out in Alg. 1. The complexity of Alg. 2 results in $O(|\mathcal{P}| * |C|)$. Effort for compliance checking can be determined as $\Sigma_C |Cl_C| \leq |\mathcal{P}| * |C|$.

For the use case depicted in Fig. 5, we obtain the following clusters: CL_{R1} = {P1}, CL_{R2} = {P1, P2}, CL_{R3} = {P2}, CL_{R4} = {P4}, CL_{R5} = {P5}, CL_{R6} = {P6}, CL_{R7} = {P1, P2, P3, P4, P5, P6}, CL_{R8} = {P1, P2, P3, P4, P5, P6}. Obviously, the clusters for R7 and R8 contain all process models. Hence, both compliance rules are either triggered for all process models or—more likely—are always triggered due to an empty antecedent pattern as the case in this scenario. Again, complementary cluster Cl_{comp} is empty.

Discussion on Scenario 1 and Scenario 2: Which indexing scenario (Alg. 1 or 2) should be applied in which situation? This depends on, for example, on the following factors: (a) the relation between

Algorithm 2 Activity-oriented process model indexing.

Require: $\mathcal{P}, C$
Ensure: $Cl_C := \emptyset \forall C \in C$, $\mathrm{Cl}_{comp} := \emptyset$
  for all $C = (N_A, N_C, E_A, E_C, E_{AC}, nt, p) \in C$ **do**
    for all $P = (N, E) \in \mathcal{P}$ **do**
      if $(\{n \in N_A | nt(n) = ANTEOCC\} = \emptyset) \vee (\{n \in N_A | nt(n) = ANTEOCC\} \subset N)$ **:=Cond then**
        $Cl_C := Cl_C \cup \{P\}$
      end if
    end for
  end for
  for all $P \in (\mathcal{P} \setminus \bigcup_C Cl_C)$ **do**
    $Cl_{comp} := Cl_{comp} \cup \{P\}$
  end for
  return Cl_C, Cl_{comp}

cardinalities of $|\mathcal{P}|$ and $|C|$ and b) number of compliance rules contained within C having an empty antecedent pattern. For case a) the following rules of thumb for deciding between Scenario 1 and Scenario 2 can be applied:

- for $|\mathcal{P}| << |C|$ apply Scenario 1
- for $|C| << |\mathcal{P}|$ apply Scenario 2

In our case study presented in Sect. 8, the number of compliance rules is by four times more than the number of process models. Hence, indexing

compliance rules for each process model seems to be more favorable due to the intially smaller number of resulting clusters. In addition, as we will discuss in Sect. 6, the comparison between the different clusters with respect to finding out about process model similarities is supported by compliance rule rather than process model indexing.

As shown for the use case displayed in Fig. 5, for each compliance rule with empty antecedent pattern, all process models in $\mathcal{P}$ have to be added to the resulting clusters. At the presence of a large number of compliance rules with empty antecedent patterns, the resulting number of consequently 'unclustered' process model clusters might counteract the effort of indexing.

Scenario 3: Aggregated rule indexing addresses the question whether the results of Alg. 1 could be still optimized by aggregating clusters. Cl_{P1} and Cl_{P2}, for example, both contain rule R2 (cf. Fig. 5). Hence it could be considered to aggregate those clusters as well as the associated process models. The decision to aggregate can only be answered by evaluating the trade-off between the benefit of reducing the number of clusters and the potential performance penalty by increasing the number of unnecessary compliance checks.

Basically, different relations between two clusters Cl_{P1} and Cl_{P2} are conceivable: a) $Cl_{P1} = Cl_{P2}$, b) $Cl_{P1} \subset Cl_{P2}$, c) $Cl_{P1} \supset Cl_{P2}$, d) $Cl_{P1} \cap Cl_{P2} = \emptyset$, and e) $Cl_{P1} \cap Cl_{P2} \neq \emptyset$. In case a), both clusters Cl_{P1} and Cl_{P2} are equal if all of the compliance rules contained with the clusters refer to process models P_1 and P_2. By merging compliance rule clusters Cl_{P1} and Cl_{P2} into one cluster, the number of clusters is reduced by one and there is no additional effort for any of both process models P_1 and P_2. Thus, in this case, cluster aggregation is advisable. In all other cases, the number of clusters will be also reduced by one, but at the expense of additional (unnecessary) compliance checks: either for P_1 against Cl_{P2} or P_2 against Cl_{P1} or both. For case d), clusters would not be aggregated as their intersection is empty. Case e) is most interesting and hard to decide as the overlap might be 'big' such that aggregation could be advisable, but also 'small' such that aggregation might lead to many unnecessary checks which even out the effect of aggregation.

In a first step, Alg. 3 aggregates two clusters only if they are equal. Based on the cluster-based similarity notions provided in Sect. 6, the condition for aggregating clusters can be adapted by aggregating clusters if their overlapping degree exceeds a certain threshold.

Algorithm 3 Aggregated rule indexing.

Require: Cl_P (cf. Alg. 1)
Ensure: $Cl_{P_{i,j}} = \emptyset$
  for all $Cl_{P_i}, Cl_{P_j} \in Cl_P$ with $Cl_{P_i} = Cl_{P_j}$ **do**
    $Cl_{P_{i,j}} = Cl_{P_i}$
    $Cl_P = (Cl_P \setminus (Cl_{P_i} \cup Cl_{P_j})) \cup Cl_{P_{i,j}}$
  end for
  return Cl_P

Generally, an aggregation of a process-model indexing as suggested by Alg. 2 is also conceivable. Here, similar considerations can be made as for the rule indexing. In addition, relations between process models in the sense of process families and variants (see Hallerbach et al. 2010), configurable process models (see Gottschalk et al. 2008; La Rosa et al. 2007; Rosemann and van der Aalst 2007), or similiar process models (see Dijkman et al. 2011; Li et al. 2008) can support a more sophisticated process model indexing. Such questions are part of future work.

Above, the effort for building the index structure is considered only. In real-world scenarios, changes of process models or the compliance rule repository might occur due to, for example, changed environmental conditions or new laws and regulations (Ly et al. 2008). As a consequence, the index structures proposed in this paper might have to be adapted accordingly. In the following, the effort for an update of the index structure is discussed along Alg. 1:

- *Adding new process model P_{new} to the repository:* determine the compliance rule cluster

$Cl_{P_{new}}$ for P_{new} based on Alg. 1 resulting in $O(|C|)$.

- *Deleting process model P from the repository:* delete corresponding compliance cluster Cl_P.
- *Adding new compliance rule* $C_{new} = (N_A^{new}, \ldots)$ *to the repository:* if $N_A^{new} = \emptyset$, add C_{new} to all existing clusters ($O(|\bigcup Cl_P|)$); otherwise find all process models $P = (N, \ldots)$ with $N_A^{new} \cap N \neq \emptyset$ and add C to the respective clusters ($O(|\mathcal{P}|)$).
- *Delete compliance rule C from repository:* delete C from all clusters ($O(|\bigcup Cl_P|)$).
- *Inserting a new activity n into process model P:* check C for rules C^n that refer to n; add C^n to cluster Cl_P associated to P ($O(|C|)$).
- *Delete activity n from process model P:* check cluster Cl_P associated to P; remove all compliance rules from $Cl_P = (N_A, \ldots)$ for which $N_A \setminus \{n\} = \emptyset$ ($O(|Cl_P|)$).

Overall, the effort for updates turns out as the maximum of $O(|C|)$ and $O(|\mathcal{P}|)$. If, for example, a new process model is added, the effort for determining relevant compliance rules is balanced by the reduction of the effort of resulting compliance checks. If a process model is deleted, the effort is limited to deleting the corresponding cluster. If a new compliance rule is added, the effort for sorting it into the right clusters, is balanced by checking all process models against the new rule. Deleting a compliance rule requires an overhead for cleaning up the clusters. The update effort for adding a new activity to a process model is balanced by the reduced effort for the compliance checks. Deleting an activity results in overhead for cleaning up the cluster associated with the process model.

6 Process Similarity Measure based on Indexing Technique

In this section, we discuss how the index structure proposed in the previous section can be used to define a similarity metric for process models. The basic idea is to consider two process models as similar if they adhere to the same or a similar set of compliance rules, cf. Def. 2:

Definition 2 Cluster-based semantic process similarity metric Let P_1, P_2 be two process models, $P_1, P_2 \in \mathcal{P}$. Let further C be a set of compliance rules and Cl_{P_1}, Cl_{P_2} be clusters associated with P_1 and P_2 by applying Alg. 1. Then, the cluster-based process similarity metric m between P_1 and P_2 is defined as follows:

$$m(P_1, P_2) := \frac{|Cl_{P_1} \cap Cl_{P_2}|}{|C|}$$

For the use case depicted in Fig. 5, all process models P1, . . ., P6 'share' compliance rules R7 and R8. Hence, initially m(P_i,P_j)=$\frac{1}{4}$ for $i, j = 1, \ldots, 6$, $i \neq j$, $i \neq 1 \vee j \neq 2$. For P1 and P2 additionally R2 is associated, resulting in m(P1,P2)=$\frac{3}{8}$. These values, however, are not very meaningful, since they consider the always triggered compliance rules R7 and R8. Such compliance rules do not contribute by any semantic information except the fact that certain activities must not be present in the process models what is completely independent of any process model structure. Hence, in the following we exclude always triggered compliance rules from our considerations. Then, in the IT baseline security scenario, without considering R7 and R8, m(P1,P2)=$\frac{1}{6}$ and for the remaining process model pairs m(P_i,P_j)=0 results.

The numerator of metric m, i. e. $|Cl_{P_1} \cap Cl_{P_2}|$ corresponds to condition **Cond** in Alg. 1, however, part $\{n \in N_A | nt(n) = ANTEOCC\} = \emptyset$ is omitted for m since it refers to compliance rules that are always triggered. For the activities that trigger the associated compliance rules by the respective ANTEOCC nodes of the antecedent pattern, the condition corresponds to the activity coverage metric (e. g. Rinderle-Ma and Reichert 2010). Activity coverage determines the ratio of activities contained in both process models and the overall number of activities in the process models. In the cluster-based similarity metric, the corresponding activities within both process models are grouped

with respect to the different compliance rules. Further, the cluster-based similarity metric weighs the number of correspondences of activities that trigger compliance rules by the number of compliance rules that are not triggered by the process models. The question is whether weighing by the overall number of compliance rules is meaningful or not, since compliance rules might refer to completely different processes and have completely different semantics. Probably, it might be more meaningful to weigh the common compliance rules by the overall number of compliance rules contained in the associated clusters, i. e. by:

$$m'(P_1, P_2) := \frac{|Cl_{P_1} \cap Cl_{P_2}|}{|(Cl_{P_1} \cup Cl_{P_2})|} \qquad (1)$$

For clusters Cl_{P1} and Cl_{P2} (cf. Fig. 5), m' turns out as $\frac{1}{3}$ reflecting the ratio of shared compliance rules, i. e. $R2$, and compliance rules that are different for the clusters, i. e. $R1$ and $R3$.

As discussed in Sect. 5, similarity metric m' can be used for aggregating compliance rule clusters as well. Then Alg. 3 can be adapted as follows:

Algorithm 4 Aggregated rule indexing based on similarity metrics.

Require: $\mathcal{P}, C, Cl_P$ (cf. Alg. 1)
Ensure: $\mathcal{P}' = \mathcal{P}$, $Cl_{P_{i,j}} = \emptyset$, $Cl_{P_k} = \emptyset \; \forall k = 1, \ldots, |\mathcal{P}|$
  for all P_k with $Cl_{P_k} = \emptyset$ **do**
    Cl_{P_k}:=$Cl_{P_k} \cup \{P_k\}$
    remove P_k from $\mathcal{P}$
  end for
  for all $P_i, P_j \in \mathcal{P}, i \neq j$ **do**
    $m'(P_i, P_j) = \frac{|Cl_{P_i} \cap Cl_{P_j}|}{|(Cl_{P_i} \cup Cl_{P_j})|}$
  end for
  for all $P_i, P_j \in \mathcal{P}$ with $m'(P_i, P_j) > x$ **do**
    $Cl_{P_{i,j}} := Cl_{P_i} \cup Cl_{P_j}$
    remove Cl_{P_i}, Cl_{P_j}
    $Cl_{P_{i,j}} := \{P_i\} \cup \{P_j\}$
    $\mathcal{P}' := \mathcal{P}' \setminus (\{P_i\} \cup \{P_j\})$
  end for
  return indexing $Cl_{P_k}, Cl_{P_{i,j}}, \mathcal{P}'$

Alg. 4 is slightly abstracted, since it aggregates pairs of process models within one 'run'. However, as we will show within our case study (cf. Sect. 8), aggregation could be repeatedly possible, as long as the similarity of newly created process clusters exceeds threshold x (this requires an adaptation of m' to process clusters instead of process models). Further on, before starting aggregating, all process models with empty compliances rule cluster are merged into one cluster and removed from the considerations in order to avoid division by 0 in m'.

7 Discussion

Maintenance Aspects: On top of the effort considerations presented in Rinderle-Ma et al. (2012), indexing can be of help for maintaining compliance rule sets. By applying Alg. 1 (or 3 respectively), the set of compliance rules that do not refer to any process model are filtered out. Reason for such orphaned compliance rules might be the continuous evolution of the compliance rule set. The other way round, we can also detect which process models are not subject to any compliance rule. Finally, by aggregating compliance rule clusters as done in Alg. 3 might yield interesting results, depending on the aggregation strategy. Recall that the presented algorithm only aggregates equal clusters. Depending on the cluster relation other strategies might be pursued. In any case, if clusters can be aggregated for several process models, this might also point to the existences of similar processes or process families. In summary, indexing contributes to the quality of compliance rule and process model sets (repositories) in the following ways:

- decreased effort for compliance checks and maintenance
- filtering out orphaned or outdated compliance rules (cf. Cl_{comp} in Alg. 1)
- filtering out process models that are not subject to any compliance rules
- finding process similarities with respect to the imposed compliance rules.

Transferability: The proposed techniques have been defined based on the representation of compliance rules by CRGs. As the basis for the presented techniques is the set of activities $A \subseteq N$ the compliance rule is defined on, in principle, the proposed indexing techniques can be transferred to other languages used to describe compliance rules such as LTL as well. Take, for example, rule R1 expressed in terms of a CRG, stating that execution of activity `PC Power Up` must be followed by execution of activity `Authentication`. Following, for example, the approach proposed by Declare (Pesic et al. 2007), R1 can be realized as LTL expression

$\Box$ `PC Power Up` $\longrightarrow \Diamond$ `Authentication`

For the indexing technique proposed in this paper, the relevant information is the set of activities referred to by the compliance rule, in both cases, {`PC Power Up, Authentication`}. More specifically, the index structure takes into consideration those activities that trigger the activation of the compliance rule (for CRGs represented as the nodes of the antecedent pattern).

The index structures as proposed in this paper refer to process design time. An important question is the transferability of the techniques to process runtime. As for the optimization techniques presented in Knuplesch et al. (2010), the number of compliance rules in a cluster can be possibly reduced at runtime based on data values provided by executing certain activities. Consider the IT Baseline Security use case, process model `P1 Password protection` as depicted in Fig. 2. Obviously, `P1` contains an alternative branching with a decision whether the `Authentication` is proofed or denied. Assume a compliance rule R': 'after the `Authentication proof` access must be granted'. As R' refers to activities contained within `P1`, R' is grouped into the associated compliance cluster during design time. At process runtime, based on the data value written for activity `Authentication`, i. e. whether it is proofed or denied, R' might be still to be considered (proofed) or not (denied). This could be used to dynamically reduce compliance clusters during process monitoring. Similar considerations can be made for temporal information referred to by compliance rules.

Implementation: As the approach presented in this paper constitutes a first step towards developing and applying indexing techniques on compliance rule repositories in Process-Aware Information Systems (PAIS), several assumptions have been made. First of all, we focused on the control flow perspective of compliance rules, i. e. the activities the compliance rules are referring to. As compliance rules might refer to other process aspects such as time, data, or resources as well (see, e. g. Mangler and Rinderle-Ma 2011), respective extensions of the proposed indexing techniques become conceivable. Moreover, it has not been considered that compliance rules might be only valid for certain time frames. This might be addressed by regularly updating the index. Regarding the implementation of the proposed indexing technique, we are currently implementing the proposed indexing technique based on the existing SeaFlows prototype as presented in Ly et al. (2010a) which is fully integrated with the adaptive process management system AristaFlow (refer to, e. g. Lanz et al. 2010). The SeaFlows editor supports the design of compliance rules as CRGs (cf. Sect. 4) and implements compliance checks during process design and run time (i. e. for compliance monitoring). We aim at implementing a set of different indexing techniques (including the one presented in this paper) based on SeaFlows providing the basis for comparing the performance of compliance checks in different scenarios.

8 Case Study

In this section, we present the case study 'Large Process and Rule Repositories in an Austrian Higher Education Institution' in which the optimization techniques for compliance checking were applied and discussed. The case study is described in an anonymous way, and a fictitious institution name (UNIQUE) is used.

ProcessID	No. of Activities	Gateways/ Decisions	Max. Possible Execution Traces	ClusterID	Rules	No. of Rules
P0001	4	0	1	C0001	{R0001, R0002}	2
P0002	4	0	1	C0002	{R0003,R0004,R0005,R0006,R0007,R0008,R0009,R0010,R0011,R0012,R0002,R0013}	12
P0003	14	4	16	C0003	{R0014,R0015,R0017,R0018,R0019,R0016,R0020,R0021}	8
P0004	3	1	2	C0004	{R0022, R0023, R0024}	3
P0005	2	2	4	C0005	{R0025, R0026, R0027}	3
P0006	4	0	1	C0006	{}	0
P0007	4	1	2	C0007	{R0028, R0029, R0030, R0031}	4
P0008	4	1	2	C0008	{R0032, R0033, R0034, R0035}	4
P0009	4	0	1	C0009	{}	0
P0010	6	1	3	C0010	{}	0
P0011	7	8	256	C0011	{R0036, R0037, R0038, R0039}	4
P0012	6	6	64	C0012	{R0040, R0041}	2
P0013	7	3	8	C0013	{R0042, R0043, R0044}	3
P0014	6	2	4	C0014	{R0045, R0043}	2
P0015	3	0	1	C0015	{R0046}	1
P0016	4	3	8	C0016	{R0047, R0048, R0049}	3
P0017	6	1	2	C0017	{}	0
P0018	3	2	4	C0018	{R0050, R0051}	2
P0019	49	9	768	C0019	{R0052, R0053, R0054, R0055, R0056, R0057, R0058, R0059, R0060, R0061, R0062, R0063, R0064, R0065, R0066, R0067, R0068, R0069, R0070, R0071, R0072}	20
P0020	12	2	4	C0020	{R0073, R0074, R0075, R0076}	4
P0021	9	3	8	C0021	{R0077, R0078, R0079}	3
P0022	2	0	1	C0022	{R0080, R0081, R0082, R0083, R0084, R0085}	5
P0023	10	3	8	C0023	{}	0
P0024	12	3	12	C0024	{R0086, R0087, R0088, R0089, R0090, R0091, R0092, R0093}	8
P0025	2	0	1	C0025	{}	0
P0026	7	0	1	C0026	{R0094, R0095, R0096, R0097, R0098}	5
P0027	6	1	2	C0027	{R0099, R0100, R0101, R0102, R0103}	5
P0028	14	0	1	C0028	{R0104}	1
P0029	5	0	1	C0029	{R0105, R0106, R0107, R0108}	4
P0030	2	1	2	C0030	{}	0
P0031	22	0	1	C0031	{R0109, R0110, R0111, R0112, R0113, R0114, R0115, R0116, R0117, R0118, R0119, R0120, R0121, R0122, R0123, R0124, }	15
P0032	10	5	48	C0032	{R0125, R0126, R0120, R0121, R0122, R0123, R0124, R0125, R0126, R0127, R0128, R0129, R0130, R0101}	14
P0033	2	0	1	C0033	{R0131, R0132, R0133, R0134, R0135, R0136, R0137, R0138}	8
P0034	9	1	2	C0034	{R0139, R0140}	2
P0035	8	1	2	C0035	{R0141, R0140}	2
P0036	7	1	2	C0036	{R0142, R0143, R0140}	3
P0037	7	1	2	C0037	{R0144, R0140}	2
P0038	12	3	8	C0038	{R0140}	1
P0039	7	2	4	C0039	{R0145, R0146, R0147, R0148, R0149, R0150, R0151, R0152, R0153, R0154, R0155}	10
P0040	3	0	1	C0040	{R0156, R0157, R0158, R0159, R0160, R0161, R0162, R0163}	8
P0041	21	1	2	C0041	{}	0
P0042	18	2	4	C0042	{R0140, R0164}	2
P0043	18	7	128	C0043	{R0164, R0165, R0166, R0167, R0168, R0169, R0170, R0171, R0172}	9
P0044	11	2	4	C0044	{R0173}	1
P0045	19	4	16	C0045	{R0174, R0175, R0176}	3
P0046	10	2	4	C0046	{R0177, R0178, R0179, R0180, R0181, R0182, R0183, R0184, R0185, R0186, R0187}	10
P0047	5	2	4	C0047	{R0188, R0189, R0190, R0191}	4
P0048	7	2	4	C0048	{R0192, R0193, R0194, R0195, R0196, R0197, R0198, R0199}	8
P0049	3	1	2	C0049	{R0200, R0201, R0202, R0203}	4
P0050	3	0	1	C0050	{R0204, R0205, R0206, R0207, R0208, R0209, R0196}	7
P0051	25	8	256	C0051	{R0210, R0211, R0212, R0213}	4
P0052	14	2	4	C0052	{R0214, R0215}	2
P0053	11	1	2	C0053	{R0216, R0217, R0218, R0219, R0220, R0221}	5
P0054	7	2	4	C0054	{R0223, R0223, R0224}	3
P0055	10	1	2	C0055	{R0225}	1
P0056	4	1	2	C0056	{R0226, R0227}	2
P0057	25	4	16	C0057	{R0228, R0229, R0230, R0231, R0232, R0233, R0234, R0235, R0236, R0237, R0238, R0239, R0240, R0241, R0242, R0243}	15
P0058	10	3	8	C0058	{R0244, R0245, R0246, R0247}	4
P0059	9	3	8	C0059	{R0248, R0249, R0250, R0251, R0252, R0253}	6
P0060	11	4	16	C0060	{R025e, R0255, R0256, R0257, R0258, R0259}	6
P0061	27	8	384	C0061	{R0260, R0261, R0140, R0223}	4
P0062	11	3	8	C0062	{R0262}	1
P0063	38	0	1	C0063	{R0263, R0264, R0265}	3
P0064	39	9	1152	C0064	{R0266, R0267, R0268, R0269, R0270, R0271, R0272, R0273, R0274, R0275, R0276, R0140}	12
P0065	23	2	4	C0065	{ R0276, R0277, R0278, R0263}	4
P0066	1	0	1	C0066	{ R0279, R0280}	2
P0067	7	0	1	C0067	{}	0
P0068	15	7	972	C0068	{R0281, R0282, R0283, R0284}	4
P0069	15	7	972	C0069	{R0286, R0281, R0282, R0283}	4
P0070	7	0	1	C0070	{R0287, R0288, R0289, R0290, R0291, R0292, R0293, R0294, R0295, R0296, R0297}	10
P0071	15	4	16	C0071	{R0298, R0299, R0300, R0301, R0302, R0303, R0304, R0305, R0306, R0307, R0308, R0309, R0310, R0311, R0312, R0313, R0314, R0315, R0316, R0317, R0318, R0319, R0320, R0321, R0322, R0323}	25
P0072	3	1	2	C0072	{R0324, R0325, R0326}	3
P0073	5	1	2	C0073	{}	0
P0074	10	2	4	C0074	{R0327, R0328, R0329, R0330}	4
P0075	8	2	4	C0075	{ R0331, R0332, R0333, R0334, R0335}	5
P0076	15	4	16	C0076	{ R0336}	1
P0077	18	5	32	C0077	{}	0
P0078	15	3	8	C0078	{}	0
P0079	8	4	16	C0079	{}	0
P0080	8	1	2	C0080	{R0337}	1
P0081	12	3	12	C0081	{R0337, R0339}	2
P0082	17	5	32	C0082	{R0337, R0339}	2
P0083	12	2	4	C0083	{}	0
P0084	5	1	2	C0084	{R0341, R0342, R0343, R0344}	4
P0085	11	1	2	C0085	{}	0
P0086	5	0	1	C0086	{R0345}	1
P0087	5	0	1	C0087	{R0346}	1
P0088	13	4	16	C0088	{}	0
P0089	13	3	8	C0089	{}	0
P0090	7	1	2	C0090	{}	0
P0091	13	1	2	C0091	{}	0
P0092	7	0	1	C0092	{R0347}	1
P0093	15	2	4	C0093	{R0348}	1
P0094	10	2	4	C0094	{R0348}	1
P0095	6	0	1	C0095	{}	0
P0096	7	0	1	C0096	{}	0
P0097	9	0	1	C0097	{R0349}	1
P0098	5	1	2	C0098	{R0350, R0351}	2
P0099	7	1	2	C0099	{R0350}	1
P0100	33	6	64	C0100	{R0352}	1
P0101	30	6	96	C0101	{R0352}	1
P0102	33	6	96	C0102	{R0353, R0354, R0352}	3
P0103	16	5	32	C0103	{R0355}	1
P0104	9	4	16	C0104	{R0356, R0357, R0358, R0359, R0360, R0361, R0362, R0363, R0364, R0365, R0366, R0367, R0368, R0369, R0370, R0371, R0372, R0373, R0374}	19
P0105	26	4	16	C0105	{R0375, R0376}	2
P0106	23	3	8	C0106	{R0377, R0376}	2
P0107	26	6	64	C0107	{}	0
P0108	17	4	16	C0108	{R0378}	1

Figure 6: Case Study UNIQUE: Activity-Oriented Compliance Rule indexing.

8.1 Case Study Setting and Findings

UNIQUE is a big educational and research institution with a wide range of various business processes. The business processes were mainly captured and described in a business process manual, retrieved in 2008. The business process manual was partly out-dated, however that did not affect our application analysis.

The business process manual contained a total of 108 imperatively modeled business processes consisting of a total of 5851 activities and 257 decisions. A business process contained between 1 to 49 activities, and between 1 to 9 decisions. The business processes referred to finances, Information Technology, career and advanced training, teaching and examination, personnel, facility and resources, strategy, and studies. About 375 compliance rules could be identified.

On average, the compliance rules contained 1 to 2 antecedent nodes reflecting the precondition for the rule and 1 to 7 consequence nodes that express the actions to be set once the preconditions hold. These consequence nodes were ordered as sequence or as alternatives. The compliance rules also contained data and time-related information, e. g. necessary information or deadlines, and were included in the activity descriptions. During the analysis of the data, the compliance rules were manually separated from the general activity descriptions.

The effort for compliance checking without applying any indexing techniques turned out as:

$$108 \text{ (business process models)} * 375 \text{ (compliance rules)} = 40,500 \text{ (checks)}.$$

UNIQUE contained four times as many compliance rules as process models. Therefore, activity-oriented compliance rule indexing was conducted (cf. Alg. 1). As illustrated in Fig. 6, the compliance rules were bundled for each process model. Applying the activity-oriented indexing technique resulted in a reduction from 40,500 to 399 checks (factor 100).

The result of activity-oriented compliance checking was optimized by aggregating the compliance rule clusters. For aggregated rule indexing, the clusters containing compliance rules, that affected more than one business process of UNIQUE, were of particular interest. The rules that were identified in more than one cluster, were highlighted in red and gold color, cf. Fig. 6. The red color indicates that the rules did not refer to any policies and laws, for example, informing an applicant about a negative decision by email. The golden-colored rules, by contrast, referred to policies and laws, e. g. referring to the Data Protection Act. Compliance rule clusters were aggregated if their relation was ≥ 0.5. The results of the aggregation are illustrated in Fig. 7. By performing the aggregated rule indexing algorithm, the number of clusters could be reduced to 80 (108 − 35 clusters + 7 new aggregated clusters).

Furthermore, compliance rules could be identified that referred to particular policies and laws such as the University Act 2002 (UG 2002), Institution's Statute (Satzung), Federal Archive Law (Bundesarchivgesetz), Occupational health and Safety (ArbeitnehmerInnenschutz), Company Agreements (Betriebsvereinbarungen), Austrian Law on Public Procurement (Bundesvergabegesetz), Act on Equal Treatment (Bundes Gleichbehandlungsgesetz), and the Data Protection Act (Datenschutzgesetz). The information was reported in the textual rule description, here an example: 'The head of the department has to check the admissibility of the proposal according to paragraph 4 of the Institution's Statute.' Fig. 8 gives an overview of all the identified policies and laws to which some of the compliance rules of UNIQUE referred to (these rules are marked in blue color in Fig. 6).

8.2 Lessons Learned and Open Questions

First of all, the number of compliance checks could be reduced when applying indexing. In particular, the case suggested to cluster compliance rules along the process models (Alg. 1). Further aggregation of clusters was applicable for 28 clusters

AggClusterID	Aggregation	Aggregation Result
aggC0001	C0006 ∪ C0009 ∪ C0010 ∪ C0017 ∪ C0023 ∪ C0025 ∪ C0030 ∪ C0041 ∪ C0067 ∪ C0073 ∪ C0077 ∪ C0078 ∪ C0079 ∪ C0083 ∪ C0085 ∪ C0088 ∪ C0089 ∪ C0090 ∪ C0091 ∪ C0095 ∪ C0096 ∪ C0107	{}
aggC0002	C0068 ∪ C0069	{R0281, R0282, R0283, R0284, R0286}
aggC0003	C0080 ∪ C0081 ∪ C0082	{R0337, R0339}
aggC0004	C0037 ∪ C0038	{R0144, R0140}
aggC0005	C0093 ∪ C0094	{R0348}
aggC0006	C0098 ∪ C0099	{R0350, R0351}
aggC0007	C0100 ∪ C0101	{R0352}

Figure 7: Case Study UNIQUE: Aggregated Rule indexing.

as they shared more than 50% of the constraints contained within these clusters.

The feasibility of the similarity metrics proposed in Sect. 6 could be investigated in more detail based on the case study. It was recognized that the business processes affected by, for example, the University-Act compliance rules highly varied in their subject matter (e. g. P0029 Register for lecture, P0071 Appeal hearing, P0084 Intellectual capital report). In turn, a thematic dependency between, for example, the business processes affected by the Law on Public Procurement compliance rules (P0064 Tender of orders, and P0070 Tender of investment projects) was observed. For the identification of similarities in business processes based on the knowledge of shared policy- or law-related compliance rules, wide-ranging (low-granular) laws and policies seemed to be not sufficient as the bundled business processes covered various thematic sectors. This is an interesting observation that demands for further refinement of the similarity metric. A possible procedure might be the fragmentation of the compliance rules according to, e. g. (highly-granular) sub-paragraphs which often focus on one subject matter.

In the UNIQUE case study, the compliance rules were captured within activity or process descriptions and hence had to be manually separated. The advantage here was that the activity labels were largely corresponding within rules and process models. In other settings such as the EBMC[2][2] project on skin cancer treatment (Dunkl et al. 2011), the rules are captured within large chunks of text and have to be extracted and modeled manually. During this step an alignment between corresponding activity labels within rules and processes can be taken into consideration. Overall, matching activity labels are often assumed for compliance checking over processes, but also for other purposes such as process mining. Rinderle-Ma et al. (2011) suggest a transition from activity equivalence towards semantic equivalence of activities based on, for example, attributes such as input or output data. Activity-oriented indexing can benefit from techniques for activity matching as proposed by Zhuge (2002) and Becker et al. (2012).

9 Summary and Outlook

In this work we presented activity-oriented indexing techniques that particularly support the management of large business process and compliance rule repositories independent of any a-priory knowledge (like policies or process meta models). In a nutshell, activity-oriented indexing clusters compliance rules for each process model they

[2] EBMC[2] stands for Evidence Based Medical Compliance Cluster, see also http://forschungscluster.meduniwien.ac.at/ebmc/

Processes and Laws	Rules	Processes	
		ID	Process Title
University Act 2002	R0108, R0109, R0150, R0151, R0152, R0156, R0163, R0301, R0302, R0303, R0304, R0305, R0306, R0307, R0308, R0309, R0310, R0311, R0312, R0313, R0314, R0315, R0316, R0317, R0318, R0319, R0320, R0321, R0322, R0323, R0332, R0340, R0341	P0029	Register for lecture
		P0031	Lecture Evaluation
		P0039	Registration for examination
		P0040	Examination planning
		P0071	Appeal hearing
		P0075	Proposal for a study program
		P0084	Intellectual capital report
Austrian Law on Public Procurement	R0270, R0271, R0274, R0287, R0288, R0289, R0290, R0291, R0292, R0293, R0294, R0295, R0296, R0297	P0064	Tender of orders
		P0070	Tender of investment project
Institution's Statute	R0054, R0057, R0110, R0113, R0114, R0173, R0327	P0019	Habilitation
		P0031	Lecture Evaluation
		P0044	Plagiarism check
Occupational Health and Safety	R0211, R0212	P0051	Evaluate place of work
Company Agreements	R0227	P0056	Lecturing compensation
Federal Archive Law	R0180	P0046	Archive inquiry
Act on Equal Treatment	R0281	P0068	Job offer for D without libraries
		P0069	Job offers for F
Data Protection Act	R0337	P0080	Data request category A
		P0081	Data request category B
		P0082	Data request category C

Figure 8: Case Study UNIQUE: Policies and Laws.

refer to. As an extension to this, the aggregated indexing technique merges two compliance rules clusters based on the number of rules present in both clusters. This potentially reduces the number of compliance rule clusters and as a consequence the effort for compliance checks. In addition to compliance rule indexing, process model indexing has been proposed and discussed. The case study from the area of higher education in Austria showed that the indexing concepts can lead to reduced effort for compliance checking.

As a second finding, the resulting indexing information was utilized for a first notion on semantic similarity between process models that states that two process models are similar to certain degree if they have to adhere to the same or similar compliance rules. The metrics can be used to, for example, aggregation of clusters. The application of the similarity metrics to the case study revealed that its straightforward application might not lead to useful results. The reason is that the granularity of the compliance rules might be quite coarse. As a result several process models might be subject to such compliance rules (and hence would be identified as similar) which are actually not similar with respect to their content. Hence, future research demands for a refinement of the first suggestions of a semantic similarity metric.

In future work, it is planned to define further techniques for managing large collections of business processes and compliance rules, particularly focusing on, for example, indexing techniques, or indexing according to data and time in business processes. This will become specifically interesting when extending the presented approach to process runtime. In addition, we aim at further elaborating on the notions for semantic similarity between process models. Particularly we are also interested in applying our results to other case studies, for example, within the health care domain.

References

Allen F. (1970) Control flow analysis. In: SIGPLAN Notices 5(7), pp. 1–19

Awad A., Decker G., Weske M. (2008) Efficient Compliance Checking Using BPMN-Q and Temporal Logic. In: Dumas M., Reichert M., Shan M. C. (eds.) Int'l Conference on Business Process Management. Lecture Notes in Computer Science Vol. 5240. Springer, Milan, Italy, pp. 326–341

Awad A., Sakr S. (2010) Querying Graph-Based Repositories of Business Process Models. In: Yoshikawa M., Meng X., Yumoto T., Ma Q., Sun L., Watanabe C. (eds.) Int'l Conference on Database Systems for Advanced Applications. Lecture Notes in Computer Science Vol. 6193. Springer, Tsukuba, Japan, pp. 33–44

Basler Ausschuss für Bankenaufsicht (2006) Internationale Konvergenz der Eigenkapitalmessung und Eigenkapitalanforderungen. Bank für Internationalen Zahlungsausgleich

Becker J., Breuker D., Delfmann P., Dietrich H. A., Steinhorst M. (2012) Identifying Business Process Activity Mappings by Optimizing Behavioral Similarity. In: Americas Conference on Information Systems. Association for Information Systems, Seattle, Washington

Chaudhuri S., Dayal U. (1997) An overview of data warehousing and OLAP technology. In: ACM Sigmod Record 26(1), pp. 65–74

Dijkman R., Dumas M., van Dongen B., Käärik R., Mendling J. (2011) Similarity of business process models: Metrics and evaluation. In: Information Systems 36(2), pp. 498–516

Dunkl R., Fröschl K. A., Grossmann W., Rinderle-Ma S. (2011) Assessing Medical Treatment Compliance Based on Formal Process Modeling. In: Holzinger A., Simonic K. M. (eds.) Conference on Information Quality in e-Health. Lecture Notes in Computer Science Vol. 7058. Springer, Graz, Austria, pp. 533–546

Federal Agency for Security in IT (2006) IT Baseline Security - Catalogues. www.bsi.bund.de. Last Access: 2014.06.18

Francescomarino C., Tonella P. (2008) Crosscutting Concern Documentation by Visual Query of Business Processes. In: Ardagna D., Mecella M., Yang J. (eds.) Business Process Management Workshops. Lecture Notes in Business Information Processing Vol. 17. Springer, Milano, Italy, pp. 18–31

Ghose A., Koliadis G. (2007) Auditing Business Process Compliance. In: Krämer B. J., Lin K. J., Narasimhan P. (eds.) Int'l Conference on Service-Oriented Computing. Lecture Notes in Computer Science Vol. 4749. Springer, Vienna, Austria, pp. 169–180

Goedertier S., Vanthienen J. (2006a) Compliant and flexible business processes with business rules. In: Regev G., Soffer P., Schmidt R. (eds.) Workshop on Business Process Modelling, Development, and Support. CEUR Workshop Proceedings Vol. 236. CEUR-WS.org, Luxembourg, pp. 94–104

Goedertier S., Vanthienen J. (2006b) Designing Compliant Business Processes with Obligations and Permissions. In: Eder J., Dustdar S. (eds.) Business Process Management Workshops. Lecture Notes in Computer Science Vol. 4103. Springer, Vienna, Austria, pp. 5–14

Gottschalk F., van der Aalst W. M. P., Jansen-Vullers M. H., La Rosa M. (2008) Configurable workflow models. In: Int'l Journal of Cooperative Information Systems 17(02), pp. 177–221

Governatori G., Milosevic Z., Sadiq S. (2006) Compliance checking between business processes and business contracts. In: Int'l Enterprise Distributed Object Computing Conference. IEEE Computer Society, Hong Kong, China, pp. 221–232

Guttman A. (1984) R-Trees: A Dynamic Index Structure for Spatial Searching. In: Yormark B. (ed.) SIGMOD, Annual Meeting. ACM, Boston, US, pp. 47–57

Halle B., Goldberg L. (2009) The Decision Model: A Business Logic Framework Linking Business and Technology. CRC Press, Taylor & Francis Group, Boca Raton, London, New York

Hallerbach A., Bauer T., Reichert M. (2010) Capturing variability in business process models: the Provop approach. In: Journal of Software Maintenance and Evolution: Research and Practice 22(6–7), pp. 519–546

Hays P. A. (2004) Case study research. In: DeMarrais K. B., Lapan S. D. (eds.) Foundations for research: Methods of inquiry in education and the social sciences. Lawrence Erlbaum Associates, Mahwah, NJ, pp. 217–234

Hoffmann J., Weber I., Governatori G. (2012) On compliance checking for clausal constraints in annotated process models. In: Information Systems Frontiers 14(2), pp. 155–177

Hornung T., Koschmider A., Oberweis A. (2007) A Recommender System for Business Process Models. In: Chari K., Kumar A. (eds.) Seventeenth Annual Workshop on Information Technologies and Systems in conjunction with International Conference on Information Systems. Montréal, Canada, pp. 127–132

Jin T., Wang J., Wu N., La Rosa M., ter Hofstede A. (2010) Efficient and Accurate Retrieval of Business Process Models through Indexing. In: Meersman R., Dillon T. S., Herrero P. (eds.) Confederated Int'l Conferences On the Move to Meaningful Internet Systems, Part I. Lecture Notes in Computer Science Vol. 6426. Springer, Hersonissos, Crete, Greece, pp. 402–409

Karagiannis D. (2008) A Business Process-Based Modelling Extension for Regulatory Compliance. In: Bichler M., Hess T., Krcmar H., Lechner U., Matthes F., Picot A., Speitkamp B., Wolf P. (eds.) Multikonferenz Wirtschaftsinformatik. GITO-Verlag, Berlin, Munich, Germany, pp. 255–256

Knuplesch D., Ly L. T., Rinderle-Ma S., Pfeifer H., Dadam P. (2010) On Enabling Data-Aware Compliance Checking of Business Process Models. In: Parsons J., Saeki M., Shoval P., Woo C. C., Wand Y. (eds.) Int'l Conference on Conceptual Modeling. Lecture Notes in Computer Science Vol. 6412. Springer, Vancouver, Canada, pp. 332–346

Kreher U., Reichert M. (2010) Storage efficient representation of instance-specific changes in Process Management Systems. UIB-2010-02. University of Ulm. Ulm

La Rosa M., Lux J., Seidel S., Dumas M., ter Hofstede A. (2007) Questionnaire-driven Configuration of Reference Process Models. In: Krogstie J., Opdahl A. L., Sindre G. (eds.) Int'l Conference Advanced Information Systems Engineering. Lecture Notes in Computer Science Vol. 4495. Springer, Trondheim, Norway, pp. 424–438

La Rosa M., Reijers H. A., van der Aalst W. M. P., Dijkman R., Mendling J., Dumas M., García-Bañuelos L. (2011) APROMORE: An advanced process model repository. In: Expert Systems with Applications 38(6), pp. 7029–7040

Lanz A., Reichert M., Dadam P. (2010) Robust and Flexible Error Handling in the AristaFlow BPM Suite. In: Soffer P., Proper E. (eds.) Information Systems Evolution - CAiSE Forum 2010, Selected Extended Papers. Lecture Notes in Business Information Processing Vol. 72. Springer, Hammamet, Tunisia, pp. 174–189

Li C., Reichert M., Wombacher A. (2008) On Measuring Process Model Similarity based on High-level Change Operations. In: Li Q., Spaccapietra S., Yu E. S. K., Olivé A. (eds.) Int'l Conference on Conceptual Modeling. Lecture Notes in Computer Science Vol. 5231. Springer, Barcelona, Spain, pp. 248–264

Liu Y., Müller S., Xu K. (2007) A static compliance-checking framework for business process models. In: IBM Systems Journal 46(2), pp. 335–261

Ly L. T., Indiono C., Mangler J., Rinderle-Ma S. (2012a) Data Transformation and Semantic Log Purging for Process Mining. In: Ralyté J., Franch X., Brinkkemper S., Wrycza S. (eds.) Int' Conference on Advanced Information Systems Engineering. Lecture Notes in Computer Science Vol. 7328. Springer, Gdansk, Poland, pp. 238–253

Ly L. T., Knuplesch D., Rinderle-Ma S., Göser K., Pfeifer H., Reichert M., Dadam P. (2010a) SeaFlows Toolset - Compliance Verification Made Easy for Process-Aware Information Systems. In: Soffer P., Proper E. (eds.) Information Systems Evolution - CAiSE Forum 2010, Selected Extended Papers. Lecture Notes in Business Information Processing Vol. 72. Springer, Hammamet, Tunisia, pp. 76–91

Ly L. T., Maggi F. M., Montali M., Rinderle-Ma S., van der Aalst W. M. P. (2015) Compliance monitoring in business processes: Functionalities, application, and tool-support. In: Information Systems 54, pp. 209–234

Ly L. T., Rinderle S., Dadam P. (2008) Integration and verification of semantic constraints in adaptive process management systems. In: Data & Knowledge Engineering 64(1), pp. 3–23

Ly L. T., Rinderle-Ma S., Dadam P. (2010b) Design and Verification of Instantiable Compliance Rule Graphs in Process-Aware Information Systems. In: Pernici B. (ed.) Int'l Conference on Advanced Information Systems Engineering. Lecture Notes in Computer Science Vol. 6051. Springer, Hammamet, Tunisia, pp. 9–23

Ly L. T., Rinderle-Ma S., Göser K., Dadam P. (2012b) On enabling integrated process compliance with semantic constraints in process management systems. In: Information Systems Frontiers 14(2), pp. 195–219

Mangler J., Rinderle-Ma S. (2011) UPC: Identification and Unification of Process Constraints. abs/1104.3609. arxiv.org

Namiri K. (2008) Model-Driven Management of Internal Controls for Business Process Compliance. PhD thesis, University of Karlsruhe

Namiri K., Stojanovic N. (2008) Towards A Formal Framework for Business Process Compliance. In: Bichler M., Hess T., Krcmar H., Lechner U., Matthes F., Picot A., Speitkamp B., Wolf P. (eds.) Multikonferenz Wirtschaftsinformatik. GITO-Verlag, Berlin, Munich, Germany, pp. 1185–1196

Peffers K., Tuunanen T., Rothenberger M. A., Chatterjee S. (2007) A design science research methodology for information systems research. In: Journal of management information systems 24(3), pp. 45–77

Pesic M., Schonenberg M., Sidorova N., van der Aalst W. M. P. (2007) Constraint-Based Workflow Models: Change Made Easy. In: Meersman R., Tari Z. (eds.) Confederated Int'l Conferences On the Move to Meaningful Internet Systems 2, Part I. Lecture Notes in Computer Science Vol. 4803. Springer, Vilamoura, Portugal, pp. 77–94

Rinderle S., Jurisch M., Reichert M. (2007) On Deriving Net Change Information From Change Logs – The DELTALAYER-Algorithm. In: Kemper A., Schöning H., Rose T., Jarke M., Seidl T., Quix C., Brochhaus C. (eds.) Datenbanksysteme in Business, Technologie und Web. LNI Vol. 103. GI, Aachen, Germany, pp. 364–381

Rinderle S., Reichert M., Jurisch M., Kreher U. (2006) On Representing, Purging, and Utilizing Change Logs in Process Management Systems. In: Dustdar S., Fiadeiro J. L., Sheth A. P. (eds.) Int'l Conference on Business Process Management. Lecture Notes in Computer Science Vol. 4102. Springer, Vienna, Austria, pp. 241–256

Rinderle-Ma S., Kabicher S., Ly L. T. (2012) Activity-Oriented Clustering Techniques in Large Process and Compliance Rule Repositories. In: Daniel F., Barkaoui K., Dustdar S. (eds.). Lecture Notes in Business Information Processing Vol. 100. Springer, Clermont-Ferrand, France, pp. 14–25

Rinderle-Ma S., Ly L. T., Dadam P. (2008) Business Process Compliance. In: EMISA Forum 28(2), pp. 24–29

Rinderle-Ma S., Mangler J. (2011) Integration of Process Constraints from Heterogeneous Sources in Process-Aware Information Systems. In: Nüttgens M., Thomas O., Weber B. (eds.) Int'l Workshop on Enterprise Modelling and Information Systems Architectures. LNI Vol. 190. GI, Hamburg, Germany, pp. 51–64

Rinderle-Ma S., Reichert M. (2010) Advanced Migration Strategies for Adaptive Process Management Systems. In: Chao K. M., Huemer C., Hofreiter B., Li Y., Shah N. (eds.) Conference on Commerce and Enterprise Computing. IEEE Computer Society, Shanghai, China, pp. 56–63

Rinderle-Ma S., Reichert M., Jurisch M. (2011) On utilizing web service equivalence for supporting the composition life cycle. In: International Journal of Web Services Research 8(1), pp. 41–67

Rosemann M., van der Aalst W. M. P. (2007) A configurable reference modelling language. In: Information Systems 32(1), pp. 1–23

Rozinat A., van der Aalst W. M. P. (2008) Conformance Checking of Processes Based on Monitoring Real Behavior. In: Information Systems 33(1), pp. 64–95

Sadiq S., Governatori G., Naimiri K. (2007) Modeling Control Objectives for Business Process Compliance. In: Alonso G., Dadam P., Rosemann M. (eds.) Int'l Conference on Business Process Management. Lecture Notes in Computer Science Vol. 4714. Springer, Brisbane, Australia, pp. 149–164

Valkenburg M. (2010) Van Ameyde International Case Study. http://www.bptrends.com/van-ameyde-international-case-study/. Last Access: 2014.06.18

van der Aalst W. M. P., van Hee K. M., van Werf J. M., Verdonk M. (2010) Auditing 2.0: Using Process Mining to Support Tomorrow's Auditor. In: IEEE Computer 43(3), pp. 90–93

Weber B., Reichert M., Mendling J., Reijers H. A. (2011) Refactoring Large Process Model Repositories. In: Computers in Industry 62(5), pp. 467–486

Weber B., Reijers H. A., Zugal S., Wild W. (2009) The Declarative Approach to Business Process Execution: An Empirical Test. In: van Eck P., Gordijn J., Wieringa J. (eds.) Int'l Conference on Advanced Information Systems Engineering. Lecture Notes in Computer Science Vol. 5565. Springer, Amsterdam, The Netherlands, pp. 470–485

Zhuge H. (2002) A process matching approach for flexible workflow process reuse. In: Information and Software Technology 44(8), pp. 445–450