Process Compliance Measurement Based on Behavioural Profiles

Matthias Weidlich[1], Artem Polyvyanyy[1], Nirmit Desai[2], and Jan Mendling[3]

[1] Hasso Plattner Institute at the University of Potsdam, Germany
{Matthias.Weidlich,Artem.Polyvyanyy}@hpi.uni-potsdam.de
[2] IBM India Research Labs, India
Nirmit123@in.ibm.com
[3] Humboldt-Universität zu Berlin, Germany
Jan.Mendling@wiwi.hu-berlin.de

Abstract. Process compliance measurement is getting increasing attention in companies due to stricter legal requirements and market pressure for operational excellence. On the other hand, the metrics to quantify process compliance have only been defined recently. A major criticism points to the fact that existing measures appear to be unintuitive. In this paper, we trace back this problem to a more foundational question: which notion of behavioural equivalence is appropriate for discussing compliance? We present a quantification approach based on behavioural profiles, which is a process abstraction mechanism. Behavioural profiles can be regarded as weaker than existing equivalence notions like trace equivalence, and they can be calculated efficiently. As a validation, we present a respective implementation that measures compliance of logs against a normative process model. This implementation is being evaluated in a case study with an international service provider.

1 Introduction

Compliance management is becoming increasingly important. Companies seek for a better control of their processes not only to satisfy new legal requirements but also to leverage cost-saving opportunities through standardization of business operations. Once non-compliant cases of operations are detected, the company can either update its process model to cover the respective case or it can impose new mechanisms to enforce best practice execution. In this way, compliance management is a central piece in the puzzle of advancing a company towards a higher degree of process maturity.

In order to make compliance management work in practice, it is required to gather detailed information on the execution of particular business processes. In recent years, process mining has emerged as a technique that automatically reworks process logs data such that managerial decision making can be supported [1]. Most importantly, certain key measures of compliance management can easily be quantified. Therefore, some compliance metrics have been proposed in recent literature, e.g., by Medeiros et al. [2], and Rozinat and van der Aalst [3].

These metrics check a set of logs against a normative process model to calculate the degree of compliance.

The aforementioned metrics have a few drawbacks. On the one hand, they rely on state-based techniques that involve replaying the logs. They are based on the notion of trace equivalence, which is a weak notion in the linear time – branching time spectrum [4]. As a consequence, these approaches have to cope with the state explosion problem in order to achieve efficient computation [5]. That, in turn, leads to the application of heuristics that have to be tuned for a certain setting. On the other hand, these metrics have been tested by Gerke et al. [6] for their applicability in a case study with a German air carrier. As it turned out, these metrics yielded non-compliance values that are significantly smaller than the degree of non-compliance perceived by the business users. While adaptations to the metrics have been proposed, the conceptual basis in terms of trace equivalence remains unchanged. These results, along with the inherent complexity of state-space based approaches, suggest to choose a different approach for measuring compliance.

In this paper, we approach the problem from the perspective of relations between pairs of activities or log events, respectively, instead of trying to replay logs according to a rather strict notion of equivalence. Thus, our contribution is a foundation of compliance measurement in a notion that is more relaxed than trace equivalence. We utilize behavioural profiles as a base line to calculate novel compliance metrics. As behavioural profiles can be calculated efficiently, we avoid performance issues of existing state-based metrics. We implemented our approach and validated it on an industry case study. In this context, our metrics showed a good approximation of the compliance as perceived by the business users.

Against this background, the paper is structured as follows. Section 2 discusses the challenge of measuring compliance by means of an example and elaborates on existing compliance metrics. Subsequently, Section 3 presents preliminaries for our investigations. Section 4 introduces compliance metrics based on behavioural profiles. Based thereon, Section 5 presents findings from our validation, for which we implemented a prototype and tested it on real-world logs. Section 6 discusses our approach in the light of related work. Finally, Section 7 concludes the paper and identifies topics for future research.

2 Background

In this section, we discuss the challenges of measuring behavioural compliance. Fig. 1 shows the example of a BPMN process model that includes 11 activities, all named with a capital letter. The diamonds define the routing behaviour of the BPMN model. Once I and A have been executed, there is a choice being made whether the branch including B is executed or as an exclusive alternative, the branch leading to the diamond with the plus sign. The latter option leads to a parallel execution of the sequences C, D, E and F, G, H. Finally, these branches are synchronized and control is passed through the merge node (with the X in the diamond) towards the completion of the process after execution of O.
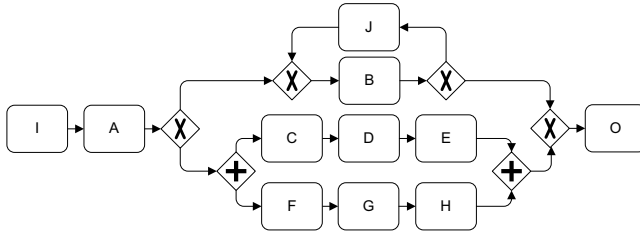
Fig. 1. Example of a BPMN process model

The challenge of compliance measurement is to quantify the degree to which a certain *set of logs* matches a normative process model. Logs (or log files) represent *observed* execution sequences of activities from the normative process model. In the desirable case the logs completely comply with the behaviour defined by the process model. In this case, the logs are *valid* execution sequences. In practice, however, observed execution sequences often deviate from predefined behaviour. This may be the case when the execution order is not explicitly enforced by the information system that records the logs. In fact, it is also possible that people deliberately work around the system [1]. This may result in logs such as the following for the process model in Fig. 1.

- ◦ Log $L_1 = I, A, C, D, F, G, E, H, O$
- ◦ Log $L_2 = I, A, C, B, E, F, H, O$
- ◦ Log $L_3 = I, A, E, D, C, H, G, O, F$
- ◦ Log $L_4 = I, C, D, F$

From these four logs, only the first one is also a valid execution sequence, i.e., it can be completely replayed by the process model. Still, the other logs capture a good share of the behaviour defined in the model. Such cases make it necessary to measure compliance *a posteriori*.

Compliance measurement has recently been approached using state-based concepts from the Petri nets theory. The idea of the fitness measure proposed by Medeiros et al. [2] is to replay the log through the model. Activities that are enabled in the model when they appear in the log are counted and related to the overall number of activities. If an activity is not enabled, the respective Petri net transition is forced to fire, which produces a token on each of its output places. In this way, one can quantify compliance of the logs against the process model as a ratio of enabled activities to the total number of activities. For example, in Fig. 1, log L_1 can be completely replayed and has therefore a compliance value of 1. Instead, log L_2 can be replayed solely until B appears in the log. This activity is then fired without being enabled. The same holds for E and H. Therefore, three firings are non-compliant from the eight firings altogether, yielding a fitness value of 0.63. In log L_3, activities E, D, H, and G are forced to fire although they are not enabled. Thus, altogether, only I, A, C, O, and F are fired correctly from nine tasks. Therefore, the fitness is 0.56. Finally, for

log L_4, the absence of activity A in the log implies a non-compliant firing of C and D, such that fitness is two out of four, which is 0.5.

As mentioned above, the metrics proposed by Medeiros et al. are based on computationally hard state space exploration, which has to be addressed using heuristics for the general case. In addition, other work reports on these metrics as yielding compliance values which are significantly lower than what is considered to be correct by domain experts [6]. Therefore, for our notion of log compliance, we do not try to replay the log through the model. Instead, we identify different types of constraints that a process model can impose for a pair of activities on the execution order activities, such as *exclusiveness* (B and C in Fig. 1) and *order* (C and E in Fig. 1). Based thereon, the preservation of these constraints is leveraged to assess the compliance of a given log. Besides constraints in terms of execution order, we also take the obligation to execute a single activity in a process model into account. Evidently, activities that are defined as being *mandatory* in the process model have to occur in a log, or can be expected to occur in future if the process has not yet completed. An example for such a mandatory activity in the model in Fig. 1 is activity A. A third group of constraints that we evaluate relates to the *causal coupling* of executions for a pair of activities. That is, the occurrence of one activity implies the occurrence of another activity. For instance, both activities, C and E, are optional for the completion of the example process. Still, the occurrence of C implies the occurrence of E, i.e., their execution is causally coupled.

While these behavioural constraints are not addressed in existing compliance metrics, we show how they can be leveraged in order to assess the compliance of a log in the remainder of this paper.

3   Preliminaries

This section gives preliminaries for our work in terms of a formal framework. For our investigations we use a notion of a process model that is based on a graph containing activity nodes and control nodes, which, in turn, captures the commonalities of process description languages. Thus, the subset of BPMN used in our initial example can be traced back to the following definition of a process model.

Definition 1. *(Process Model) A* process model *is a tuple $P = (A, a_i, a_o, C, F, T)$ with*

- *A as a non-empty set of activity nodes, and C as a set of control nodes, A and C are disjoint,*
- *$a_i \in A$ as an initial activity, $a_o \in A$ as a final activity,*
- *$F \subseteq ((A \setminus \{a_o\}) \cup C) \times ((A \setminus \{a_i\}) \cup C)$ as the flow relation, and*
- *$T : C \mapsto \{and, or, xor\}$ as a function that assigns a type to control nodes.*

In the remainder of this paper, the identity relation for activities is denoted by id_A, i.e., $(a, a) \in id_A$ for all $a \in A$. Further on, we do not formalise the execution semantics of a process model, but assume an interpretation of the

model following on common process description languages, such as BPMN, EPCs, or UML activity diagrams. It is worth to mention that we do not assume a certain definition of semantics for the inclusive OR construct, which raises serious issues in cyclic structures. We solely assume the existence of such a definition. Under the assumption of a definition of execution semantics, the set of all valid execution sequences, as well as the notion of a process log, can be defined as follows.

Definition 2. *(Execution Sequence, Execution Log) The set of* execution sequences $\mathcal{E}_{\mathcal{P}}$ *for a process model* $P = (A, a_i, a_o, C, F, T)$ *is the set of all lists of the form* $\{a_i\} \cdot A^* \cdot \{a_o\}$, *that can be created following on the execution semantics of* P. *An* execution log L_P *that has been observed for* P *is a non-empty list of the form* A^*.

Note that we speak of an *execution sequence* of a process model, solely in case the sequence is valid regarding the process model, i.e., it can be completely replayed by the model. In contrast, a *log* is a non-empty sequence over the activities of a process model. As a short-hand notation, we use $A_L \subseteq A$ to refer to the subset of activities of a process model that is contained in log L.

In order to capture the constraints imposed by a process model on the order of activity execution, we rely on the concept of a behavioural profile [7]. Behavioural profile defines relations for all pairs of activities of a process model. These relations, in turn, might be interpreted as the essential behavioural characteristics specified by the models. All behavioural relations are based on the notion of *weak order*. That is, two activities are in weak order, if there exists an execution sequence in which one activity occurs after the other. Here, only the existence of an execution sequence is required.

Definition 3 (Weak Order (Process Model)). *Let* $P = (A, a_i, a_o, C, F, T)$ *be a process model and* $\mathcal{E}_{\mathcal{P}}$ *its set of execution sequences. The* weak order relation $\succ_P \subseteq (A \times A)$ *contains all pairs* (x, y), *such that there is an execution sequence* $\sigma = n_1, \ldots, n_m$ *in* $\mathcal{E}_{\mathcal{P}}$ *with* $j \in \{1, \ldots, m - 1\}$ *and* $j < k \leq m$ *for which holds* $n_j = x$ *and* $n_k = y$.

Based thereon, we define the relations of the behavioural profile for pairs of activities. Each pair can be related by weak order in three different ways.

Definition 4 (Behavioural Profile (Process Model)). *Let* $P = (A, a_i, a_o, C, F, T)$ *be a process model. A pair* $(x, y) \in (A \times A)$ *is in at most one of the following relations:*
- *The* strict order relation $\leadsto_P$, *iff* $x \succ_P y$ *and* $y \not\succ_P x$.
- *The* exclusiveness relation $+_P$, *iff* $x \not\succ_P y$ *and* $y \not\succ_P x$.
- *The* interleaving order relation $||_P$, *iff* $x \succ_P y$ *and* $y \succ_P x$.

The set $\mathcal{B}_P = \{\leadsto_P, +_P, ||_P\}$ *is the* behavioural profile *of* P.

Note that we say that a pair (x, y) is in *reverse strict order*, denoted by $x \leadsto_P^{-1} y$, if and only if $y \leadsto_P x$. Interleaving order is also referred to as observation concurrency. Further on, the relations of the behavioural profile along with reverse

strict order partition the Cartesian product of activities [7]. We illustrate the relations of the behavioural profile by means of our example model in Fig. 1. Here, for instance, it holds $I \rightsquigarrow D$. Evidently, strict order does not imply the actual occurrence, i.e., activity D might not be executed. Further on, $B + C$ as both activities will never occur in a single valid execution sequence of the model, and $C||G$ as C might occur before G and vice versa. Note that it holds $B||J$ due to the control flow cycle. That is, an occurrence of J is followed by an occurrence of B. Further on, an activity is either said to be *exclusive to itself* (e.g., $I + I$) or *in interleaving order to itself* (e.g., $B||B$). The former holds, when an activity cannot be repeated, whereas the latter implies that the activity is part of a control flow cycle.

The concept of a behavioural profile relates pairs of activities according to their *order* of potential occurrence, whereas further behavioural characteristics are not considered. Both, the fact that an activity is *mandatory* for process completion and *causality* between activities are not covered. Causality, in turn, involves two orthogonal aspects, i.e., the *order* of activity occurrences and their *causal coupling* (the occurrence of one activity enforces the occurrence of another activity). While the former is already addressed by the behavioural profile in terms of the (reverse) strict order relation, the latter is not captured. In order to cope with these aspects, the behavioural profile is extended by a fourth relation, yielding the causal behavioural profile.

Definition 5 (Causal Behavioural Profile (Process Model))
Let $P = (A, a_i, a_o, C, F, T)$ be a process model.
- *A pair $(x, y) \in (A \times A)$ is in the* co-occurrence relation $\gg_P$, *iff for all execution sequences $\sigma = n_1, \ldots, n_m$ in $\mathcal{E}_\mathcal{P}$ it holds $n_i = x$ with $1 \leq i \leq m$ implies that there is an index $j \in \{1, \ldots, m\}$, such that $n_j = y$.*
- *The set $\mathcal{B}_P^+ = \mathcal{B}_P \cup \{\gg_P\}$ is the* causal behavioural profile *of P.*

Given a process model $P = (A, a_i, a_o, C, F, T)$, the set of mandatory activities $A_M \subseteq A$ is given by all activities that are co-occurring with the initial activity, that is, $A_M = \{a \in A \mid a_i \gg_P a\}$. Moreover, causality holds between two activities $a_1, a_2 \in A$, if they are in strict order, $a_1 \rightsquigarrow_P a_2$ and the occurrence of the first implies the occurrence of the second, $a_1 \gg_P a_2$. Again, we refer to the example in Fig. 1 for illustration purposes. In this model, there are only three mandatory activity, namely $A_M = \{I, A, O\}$. Moreover, it holds $C \gg E$ and $E \gg C$. Thus, all complete execution sequences of the process model that contain activity C are required to also contain activity E, and vice versa. In addition, the model specifies a strict order relation between both activities, $C \rightsquigarrow E$, such that we speak of a causal dependency from C to E.

4   Measuring Compliance Based on Behavioural Profiles

This section introduces compliance metrics based on behavioural profiles. First, Section 4.1 shows how the concepts introduced in the previous section can be lifted to logs. Second, we elaborate on a hierarchy between the relations of the

behavioural profile in Section 4.2. Based thereon, we introduce the compliance metrics in Section 4.3.

4.1   Causal Behavioural Profiles for Logs

In order to lift the concept of behavioural profiles to logs, first and foremost, we have to clarify the notion of weak order for logs. Following on the definition given for process models, two activities are in weak order in a log, if the first occurs before the second.

Definition 6 (Weak Order (Log)). *Let $L_P = n_1, \ldots, n_m$ be a log of a process model $P = (A, a_i, a_o, C, F, T)$. The weak order relation $\succ_L \subseteq (A_L \times A_L)$ contains all pairs (x, y), such that there exists two indices $j, k \in \{1, \ldots, m-1\}$ with $j < k \leq m$ for which holds $n_j = x$ and $n_k = y$.*

Based thereon, we define the behavioural profile of a log.

Definition 7 (Behavioural Profile (Log)). *Let $L_P = n_1, \ldots, n_m$ be a log of a process model $P = (A, a_i, a_o, C, F, T)$. A pair $(x, y) \in (A_L \times A_L)$ is in at most one of the following relations:*
  ○ *The strict order relation $\leadsto_L$, iff $x \succ_L y$ and $y \nsucc_L x$.*
  ○ *The interleaving order relation $\|_L$, iff $x \succ_L y$ and $y \succ_L x$.*
The set $\mathcal{B}_L = \{\leadsto_L, \|_L\}$ is the behavioural profile of L.

Again, the pair (x, y) is in *reverse strict order*, denoted by $x \leadsto_L^{-1} y$, if and only if $y \leadsto_L x$. The relations of the behavioural profile along with reverse strict order partition the Cartesian product of elements of a log. For the exemplary log $L_3 = I, A, E, D, C, H, G, O, F$, for instance, it holds $C \leadsto F$ and $D \leadsto^{-1} A$.

It is worth to mention that there are fundamental differences when interpreting the behavioural profile for a process model and a log. On the one hand, in contrast to the profile of a process model, we cannot observe exclusiveness between two activities in a single log. On the other hand, activities that might be enabled concurrently in a process model (e.g., C and G in our example) are related by interleaving order in the behavioural profile of the model ($C\|G$). However, if both activities are not part of a control-flow cycle, they might be related by strict order or reverse strict order in the profile of a corresponding log. For instance, for the log $L_3 = I, A, E, D, C, H, G, O, F$, we observe $C \leadsto G$ as the behavioural relation for activities C and G.

Moreover, we can also lift the definition of a causal behavioural profile to process logs. Obviously, all elements of a log are co-occurring.

Definition 8 (Causal Behavioural Profile (Log)). *Let $L_P = n_1, \ldots, n_m$ be a log of a process model $P = (A, a_i, a_o, C, F, T)$*
  ○ *The co-occurrence relation $\gg_L = (A_L \times A_L)$ contains all pairs of log elements.*
  ○ *The set $\mathcal{B}_L^+ = \mathcal{B}_L \cup \{\gg_L\}$ is the causal behavioural profile of L.*

4.2   A Hierarchy of Behavioural Relations

As mentioned above, there is a fundamental difference between behavioural profiles of process models and of logs. The former defines relations based on the set of *all* possible execution sequences, whereas the latter considers only one observed execution sequence as defined by the log. In order to cope with this phenomenon, we introduce a hierarchy between the relations of behavioural profiles (we neglect the co-occurrence relation at this stage). The idea is to order the behavioural relations based on their *strictness*. We consider the exclusiveness relation as the strongest relation, as it completely disallows two activities to occur together in an execution sequence. In contrast, the interleaving order relation can be seen as the weakest relation. It allows two activities to occur in any order in an execution sequence. Consequently, the strict order and reverse strict order relation are intermediate relations, as they disallow solely a certain order of two activities. We formalize this hierarchy between behavioural relations as a *subsumption predicate*. Given two behavioural relations between a pair of two activities, this predicate is satisfied, if and only if the first relation is equal or weaker than the second.

Definition 9 (Subsumption Predicate). *Given two behavioural relations* $R, R' \in \{\rightsquigarrow, \rightsquigarrow^{-1}, +, ||\}$ *of the same or different behavioural profiles, the* subsumption predicate $\mathcal{S}(R, R')$ *is satisfied, iff* $(R \in \{\rightsquigarrow, \rightsquigarrow^{-1}\} \;\land\; R' = +)$ *or* $R = R'$ *or* $R = ||$.

Again, we illustrate this concept using the example model in Fig. 1 and the log $L_3 = I, A, E, D, C, H, G, O, F$. As mentioned above, for activities C and G, it holds $C||G$ in the profile of the process model and $C \rightsquigarrow G$ in the profile of the log. The former specifies that C and G might occur in any order in an execution sequence, owing to the interleaving semantics of activities that are enabled concurrently. The latter, in turn, captures the fact that the occurrences of C and G in the log are ordered. However, we see that there is a subsumption relation between both relations, as $\mathcal{S}(||, \rightsquigarrow)$ is satisfied. Evidently, this information has to be taken into account when assessing compliance of logs. This stems from the fact that logs do not hint at potential interleaving execution of activities.

4.3   Compliance Metrics

For our measurements of compliance between a process model and a log, we consider three aspects separately, namely *execution order*, *mandatory activities*, and *casual coupling*. While the last two aspects address the question *what* should be contained in the log (activities that are mandatory in a global sense or that are implied by the occurrence of other activities in the log), the first aspect clarifies *how* these activities should be ordered in the log. While each of these aspects is assessed by a separate compliance degree, their values may be aggregated into a single compliance degree.

Execution order compliance is motivated by the fact that the order of execution of activities as specified by the log should be in line with the ordering

constraints as imposed by the process model. Evidently, the question whether there is such a difference is of limited usefulness. Instead, any deviation has to be quantified to allow for thorough compliance analysis. We achieve such a quantification based on the notion of behavioural profiles and the hierarchy of behavioural relations. That is, we analyse the Cartesian product of activities that are contained in a log and determine, whether the behavioural relation for a pair of activities in the log is subsumed by the relation specified in the process model. Based thereon, the degree of behavioural compliance is defined as a ratio of consistent behavioural relations relative to the number of activity pairings in the log. Note that for the case of a behavioural relation that holds between an activity and itself, the subsumption predicate cannot be applied. If an activity can occur at most once in the process model (it is exclusive to itself), it is not allowed to occur multiple times in the log (it cannot be in interleaving order to itself), whereas the opposite case (exclusive in the log, in interleaving order in the model), as well as equal relations, are considered to be compliant. Based thereon, we define execution order compliance.

Definition 10 (Execution Order Compliance). *Let $L_P = n_1, \ldots, n_m$ be a log of a process model $P = (A, a_i, a_o, C, F, T)$. Then, the set $SR \subseteq (A_L \times A_L)$ contains all pairs of activities (x, y), for which the behavioural relation in L_P is subsumed by the relation in P, i.e., $\forall~R \in (\mathcal{B}_P \cup \{\leadsto_P^{-1}\}), R' \in (\mathcal{B}_L \cup \{\leadsto_L^{-1}\})$ $[~(xRy \wedge xR'y) \Rightarrow (\mathcal{S}(R, R')~\vee~(x = y~\wedge~R = ||_P~\wedge~R' = +_L))~]$. The degree of execution order compliance of L_P to P is defined as*

$$\mathcal{EC}_{L_P} = \frac{|SR|}{|(A_L \times A_L)|}.$$

Note that the compliance degree $\mathcal{EC}_{L_P}$ for a log is between zero, i.e., no execution order compliance at all, and one indicating full execution order compliance. It is worth to mention that this degree is independent of the size of the process model, as solely the activity pairs found in the log are considered in the computation. Given the log $L_3 = I, A, E, D, C, H, G, O, F$ and our initial example (cf., Fig.1), we see that various order constraints imposed by the model are not satisfied. For instance, $D \leadsto E$ and $G \leadsto H$ are specified in the model, whereas we have $D \leadsto^{-1} E$ and $G \leadsto^{-1} H$ in the profile of the log. That, in turn, leads to an execution order compliance degree of $\mathcal{EC} = \frac{67}{81} \approx 0.83$ for this particular log.

Due to the fact that execution order compliance relates solely to the ordering constraints, we embrace further process characteristics for compliance measurements. In particular, we take the mandatory activities that are required for completion of the process into account. The ratio of mandatory activities of the process model that are contained in the log, and all mandatory activities would be a straight-forward measure for this aspect. However, we want to consider also logs that might not have completed yet, such that missing mandatory activities might be added later on. In order to cope with this, not all mandatory activities of the process model are required to occur in the log. Instead, we consider only those mandatory activities, for which we can deduce from the log that they

should have been observed already. That is, a mandatory activity is considered, if it is either in the log, or it is in strict order with one of the activities in the log.

Definition 11 (Mandatory Execution Compliance). *Let $L_P = n_1, \ldots, n_m$ be a log of a process model $P = (A, a_i, a_o, C, F, T)$ and $A_M = \{a \in A \mid a_i \gg_P a\}$ the set of mandatory activities of P. Then, the set $EA_M \subseteq A_M$ contains all mandatory activities a_1 that are in the log or can be expected to be in the log, i.e., $a_1 \in A_L$ or $\exists\, a_2 \in A_L \,[\, a_1 \neq a_2 \,\wedge\, a_1 \rightsquigarrow_P a_2 \,]$. The degree of mandatory execution compliance of L_P to P is defined as*

$$\mathcal{MC}_{L_P} = \begin{cases} 1 & \text{if } EA_M = \emptyset, \\ \frac{|A_L \cap A_M|}{|EA_M|} & \text{else.} \end{cases}$$

Consider the log $L_4 = I, C, D, F$ of our initial example. The process model in Fig. 1 specifies three mandatory activities, i.e., activities I, A, and O. While activity I is in the log, activity A is not, although we know that it should have been observed already owing to the strict order relation between A and C, D, and E, respectively. The mandatory activity O is not required to occur in the log, as the log does not contain any activity that is in strict order with O. Therefore, the log contains one out of two expected mandatory activities, such that the execution compliance degree is $\mathcal{MC} = 0.5$. Note that the mandatory execution compliance degree might be overestimated. That is, a mandatory activity might be part of a control flow cycle, while the log does not contain any activity that is in strict order with the mandatory activity. In this case, the mandatory activity will not be considered in the mandatory execution compliance degree. This holds, even though the log might already contain activities that can only be reached after executing the mandatory activity in the process model.

The mandatory execution compliance relates to activities that should be observed in each log as they are required for completion of the process. Still, there might be dependencies between the occurrences of activities even though they are not mandatory. In the model in Fig.1, for instance, neither activity C nor E are mandatory for completion. However, there is a causal relation between both, as the occurrence of C implies the occurrence of E. While the correctness of the order of occurrence for both activities has already been incorporated in the notion of execution order compliance, the causal coupling of their occurrences has still to be taken into account. Therefore, for each activity a in the log, we also check whether activities for which the occurrence is implied by a are also in the log. Again, we have to consider the eventuality of incomplete logs, such that this requirement must hold only in case there is sufficient evidence that the implication is not satisfied. That is, there is another activity in the log for which the model specifies strict order to the activity under investigation.

Definition 12 (Causal Coupling Compliance). *Let $L_P = n_1, \ldots, n_m$ be a log of a process model $P = (A, a_i, a_o, C, F, T)$. Then, the set $EC_M \subseteq (\gg_P \setminus id_A)$ contains all pairs of co-occurring activities (a_1, a_2), such that the first activity is in the log, while the second activity is in the log or can be expected to be in the*

Table 1. Compliance results for the logs of the initial example (cf., Section 2)

Log	$\mathcal{EC}$ Execution Order	$\mathcal{MC}$ Mandatory Execution	$\mathcal{CC}$ Causal Coupling	$\mathcal{C}$ Overall Compliance
$L_1 = I, A, C, D, F, G, E, H, O$	1.00	1.00	1.00	1.00
$L_2 = I, A, C, B, E, F, H, O$	0.88	1.00	0.80	0.89
$L_3 = I, A, E, D, C, H, G, O, F$	0.83	1.00	1.00	0.94
$L_4 = I, C, D, F$	1.00	0.50	0.69	0.73

log, i.e., $a_1 \in A_L$ and $a_2 \in A_L \ \vee \ \exists \ a_3 \in A_L \ [\ a_2 \leadsto_P a_3 \ ]$. The degree of causal coupling compliance *of L_P to P is defined as*

$$CC_{L_P} = \begin{cases} 1 & \text{if } EC_M = \emptyset, \\ \frac{|(A_L \times A_L)\backslash id_{A_L} \cap \gg_P|}{|EC_M|} & \text{else.} \end{cases}$$

We illustrate causal coupling compliance using our initial example and the log $L_2 = I, A, C, B, E, F, H, O$. Although all mandatory activities are contained in the log, we see that, for instance, $C \gg D$ is not satisfied. This is penalised as the log contains E and it holds $D \leadsto E$ in the process model. In other words, the occurrence of E in the log provides us with evidence that we should have observed D, too. Thus, the absence of this activity impacts on the causal coupling compliance. Note that the same holds true for all co-occurrence relations involving activity G, such that the degree of causal coupling compliance is $CC = \frac{33}{41} \approx 0.80$ for this particular log. It is worth to mention that the causal coupling compliance degree might be *overestimated*. An example for this phenomenon would be the log I, A, J for the model in Fig. 1. There is a causal coupling $J \gg B$ in this model. However, the absence of B would not be penalised as there is no activity in the log that is in strict order for B and, therefore, would provide us with sufficient evidence that B should have been observed already.

Given compliance degrees for the three aspects, *execution order*, *mandatory activities*, and *casual coupling*, we can aggregate the compliance degree of a log.

Definition 13 (Log Compliance). *Let $L_P = n_1, \ldots, n_m$ be a log of a process model $P = (A, a_i, a_o, C, F, T)$. The compliance of L_P to P is defined as*

$$C_{L_P} = \frac{1}{3}(\mathcal{EC}_{L_P} + \mathcal{MC}_{L_P} + \mathcal{CC}_{L_P}).$$

Applied to our example process model and the four exemplary logs introduced in Section 2, our compliance metrics yield the results illustrated in Table 1. As expected, the first log L_1, which represents a valid execution sequence of the process model satisfies all constraints, such that our metric indicates full overall compliance. In contrast, the execution order constraints are not fully satisfied in the second log L_2 (e.g., the exclusiveness in the model between B and C is broken in the log). In addition, the causal coupling is not completely respected in the
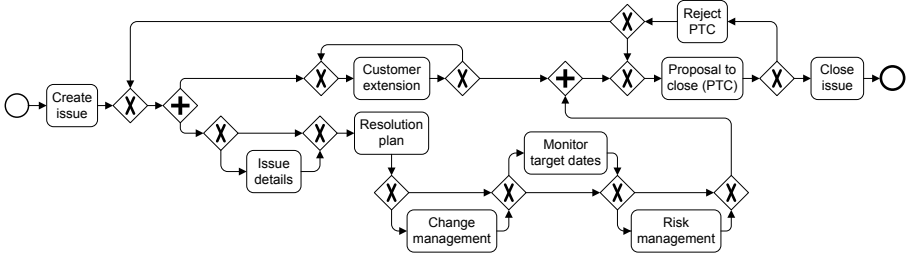
Fig. 2. BPMN model of the Security Incident Management Process (SIMP)

log either, as discussed above. Further on, for the log L_3, the execution order is not completely in line with the process model. Regarding the log L_4, we already discussed the absence of a mandatory activity that should have been observed, leading to a mandatory execution compliance degree below one. Note that this also impacts on the causal coupling compliance. Thus, a mandatory execution compliance degree smaller than one, will also be reflected in the causal coupling compliance degree. However, the opposite does not hold true, as illustrated for the log L_2. Although this log shows full mandatory execution compliance, its causal coupling compliance degree is below one.

5   Case Study: Security Incident Management Process

To demonstrate and evaluate our approach, we apply it to the Security Incident Management Process (SIMP) — which is an issue management process used in global service delivery centres. The process and the logs have been minimally modified to remove confidential information. Fig. 2 shows the BPMN model of SIMP solicited from domain experts.

SIMP is used in one of IBM's global service delivery centres that provides infrastructure management and technical support to customers. When a customer reports a problem or requests a change, an *issue* is created, spawning a new instance of the process. Details about the issue *may* be updated, a plan to resolve the issue *must* be created, and change management related activities *may* be performed if required. Then, target dates for issue resolution *may* be monitored and relevant risks *may* be documented. A *Customer Extension* of target dates may be processed during any of the above activities (parallel path). Once the steps for resolution are taken and verified, the resolver *must* propose to close the issue. Based on the evidence that the issue is indeed resolved, the issue creator *may* close the issue. Otherwise, the proposal *must* be rejected.

For the SIMP process we analysed 852 logs, each consisting of a set of log entries. Such a log entry has an activity name, activity description, and the time-stamp marking the time of execution of the activity. Although the process is standardized and documented, it is not orchestrated via workflow tools in the IBM's global service delivery centre under investigation. Instead, it is manually

Table 2. Compliance results for the SIMP process derived from 852 logs

	$\mathcal{EC}$ Execution Order	$\mathcal{MC}$ Mandatory Execution	$\mathcal{CC}$ Causal Coupling	$\mathcal{C}$ Overall Compliance
Average Value	0.99	0.96	0.95	0.96
Min Value	0.31	0.60	0.50	0.61
Max Value	1.00	1.00	1.00	1.00
Logs with Value of 1.00	78.64%	84.15%	84.15%	76.17%

carried out. Hence, the employees are free to deviate from the process. As a result, the logs may or may not specify valid execution sequences of the process model.

For each log, we analysed its compliance using the metrics proposed in this paper. Table 2 gives a summary of this analysis in terms of the average compliance value of all logs, the observed minimal and maximal compliance values, and the share of fully compliant logs. The compliance values were discussed with the manager of the process. The average values reflect the manager's perception that SIMP is running satisfactory and most cases are handled in a compliant way. As the minimum values show, it was also possible to identify cases of low compliance.

We are not able to directly compare our results with the approach proposed in [2]. This is mainly due to the inherent complexity of the state space exploration, which is exponential in the general case. Even a maximally reduced Petri net of our SIMP process contains a lot of silent steps owing to several activities, for which execution is optional. That, in turn, leads to a significant increase of the state space to investigate when trying to replay a log. While compliance values might still be derived following the most greedy strategy, these results are of a limited validity as they highly underestimate the degree of compliance. However, it is worth to mention that even with the most greedy strategy, computation of the compliance values for all logs took around 15 seconds. In contrast, computing the compliance metrics proposed in this paper for all logs, in turn, could be done within milliseconds. Moreover, an isolated analysis of a sample of 30 logs for which computation of the fitness metric is possible with a 5-step-ahead strategy revealed that the fitness compliance values are all lower than the values derived by our metrics. This is in line with the criticism of [6] that the fitness concept appears to be too strict. These differences probably stem from the different normalisation of any behavioural deviation. The fitness metric relates tokens additionally created (removed) to the number of all consumed (produced) tokens when replaying a log. In contrast, our approach considers violations on the level of relations between activities. Therefore, a single violation according to our notion, can be reflected multiple times in the fitness metric.

The adaptations of [6] could not be compared either to our values since the respective implementation is not publicly available. But as they are also defined on state concepts, we can assume results similar to those obtained above.

6 Related Work

In this section we discuss the relation of our work to compliance measurement, behavioural equivalence, and process similarity.

Compliance measures are at the core of process mining. Similar relations, but not exactly those of behavioural profiles, are used in [8] to characterise a process as a pre-processing step for deriving a model. As mentioned before, the work in [2,3] proposes a fitness measure for process mining, while, based thereon, adaptations are discussed in [6]. In this paper, we demonstrated that our approach benefited from the efficient calculation of the behavioural profiles from free-choice process models as defined in [7]. In fact, behavioural profiles can be derived in $O(n^3)$ time with n as the number of activities of a process model. Therefore, in contrast to the fitness calculation, our metrics can be computed within milliseconds. Further on, following on the criticism of [6], our case study provided us with evidence that our metrics are close to managers' perception of compliance. The detection of differences between process models, not their measurement, is also discussed in related work. The approaches presented in [9,10] provides a systematic framework of diagnosis and resolution of such mismatches.

The concept of behavioural profile in general relates to different notions of *behavioural equivalence* such as trace equivalence and bisimulation. These notions build on state concepts and can often not be calculated as efficiently as behavioural profiles. They also yield only a true or false answer and they are not directly applicable to execution sequences [11,12]. *Behaviour inheritance* is closely related to these notions. Basten et al. [13] define protocol inheritance and projection inheritance based on labelled transition systems and branching bisimulation. A model inherits the behaviour of a parent model, if it shows the same external behaviour when all actions that are not part of the parent model are either *blocked* (protocol inheritance) or *hidden* (projection inheritance). Similar ideas have been presented in [14,15]. The boolean characteristics of these notions have been criticized as inadequate for many process measurement scenarios [2].

The question of *process similarity* has been addressed from various angles. Focussing on behavioural aspects, [16,17] introduce similarity measures based on an edit distance between workflows. Such an edit distance might be based on the language of the workflow, the underlying automaton, or based on the n-gram representation of the language. A similar approach is also taken in [18], in which the authors measure similarity based on high-level change operations that are needed to transform one model into another. Close to our behavioural abstraction of a behavioural profile are causal footprints as introduced in [19]. The authors also show how the footprints can be leveraged to determine the similarity between process models. All these similarity notions are either expensive in terms of calculation, whereas behavioural profiles can be calculated efficiently.

7 Conclusion

In this paper, we have discussed the challenges of providing compliance measurements in an appropriate and efficient way. Our contribution is a novel proposal

for metrics based on behavioural constraints on pairs of tasks for compliance measurement. In this way, we avoided performance problems of state-based metrics by using behavioural profiles as the underlying equivalence notion. Our log compliance metric covers three aspects including execution order, mandatory execution, and causal coupling, which can all be derived from the causal behavioural profile. We implemented these metrics and tested them in a case study.

In future work, we aim to study the merits of our novel approach in further industry collaborations. The performance gain of using behavioural profiles is of serious importance for various use cases. Up until now, compliance measurement had to be conducted offline in a batch mode due to being very time consuming. We aim to investigate those scenarios where an instantaneous compliance measurement is valuable. In particular, compliance measurement in the financial industry might eventually benefit from this innovation, e.g., to cancel running transactions that exhibit non-compliant behaviour.

References

1. van der Aalst, W., Reijers, H., Weijters, A., van Dongen, B., de Medeiros, A., Song, M., Verbeek, H.: Business process mining: An industrial application. Inf. Syst. 32(5), 713–732 (2007)
2. de Medeiros, A., van der Aalst, W., Weijters, A.: Quantifying process equivalence based on observed behavior. Data Knowl. Eng. 64(1), 55–74 (2008)
3. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Inf. Syst. 33(1), 64–95 (2008)
4. Glabbeek, R.: The Linear Time – Brancing Time Spectrum I. The semantics of concrete, sequential processes. In: Handbook of Process Algebra. Elsevier, Amsterdam (2001)
5. Valmari, A.: The state explosion problem. In: Reisig, W., Rozenberg, G. (eds.) APN 1998. LNCS, vol. 1491, pp. 429–528. Springer, Heidelberg (1998)
6. Gerke, K., Cardoso, J., Claus, A.: Measuring the compliance of processes with reference models. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2009, Part I, LNCS, vol. 5870, pp. 76–93. Springer, Heidelberg (2009)
7. Weidlich, M., Mendling, J., Weske, M.: Computation of behavioural profiles of process models. Technical report, Hasso Plattner Institute (June 2009)
8. Aalst, W., Weijters, A., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE Trans. Knowl. Data Eng. 16(9), 1128–1142 (2004)
9. Küster, J.M., Gerth, C., Förster, A., Engels, G.: Detecting and resolving process model differences in the absence of a change log. In: [20], pp. 244–260
10. Dijkman, R.M.: Diagnosing differences between business process models. In: [20], pp. 261–277
11. Glabbeek, R., Goltz, U.: Refinement of actions and equivalence notions for concurrent systems. Acta Inf. 37(4/5), 229–327 (2001)
12. Hidders, J., Dumas, M., Aalst, W., Hofstede, A., Verelst, J.: When are two workflows the same? In: CATS. CRPIT, vol. 41, pp. 3–11 (2005)
13. Basten, T., Aalst, W.: Inheritance of behavior. JLAP 47(2), 47–145 (2001)
14. Ebert, J., Engels, G.: Observable or Invocable Behaviour - You Have to Choose. Technical Report 94-38, Leiden University (December 1994)

15. Schrefl, M., Stumptner, M.: Behavior-consistent specialization of object life cycles. ACM Trans. Softw. Eng. Methodol. 11(1), 92–148 (2002)
16. Wombacher, A.: Evaluation of technical measures for workflow similarity based on a pilot study. In: Meersman, R., Tari, Z. (eds.) OTM 2006, Part I, LNCS, vol. 4275, pp. 255–272. Springer, Heidelberg (2006)
17. Wombacher, A., Rozie, M.: Evaluation of workflow similarity measures in service discovery. In: Service Oriented Electronic Commerce. LNI, vol. 80, pp. 51–71 (2006)
18. Li, C., Reichert, M., Wombacher, A.: On measuring process model similarity based on high-level change operations. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 248–264. Springer, Heidelberg (2008)
19. Dongen, B., Dijkman, R.M., Mendling, J.: Measuring similarity between business process models. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 450–464. Springer, Heidelberg (2008)
20. Dumas, M., Reichert, M., Shan, M.-C. (eds.): BPM 2008. LNCS, vol. 5240. Springer, Heidelberg (2008)