Checking Process Compliance on the Basis of Uncertain Event-to-Activity Mappings

Han van der Aa[1], Henrik Leopold[1], Hajo A. Reijers[1,2]

[1] Department of Computer Sciences, VU University Amsterdam,
Amsterdam, The Netherlands
j.h.vander.aa|h.leopold||h.a.reijers@vu.nl
[2] Department of Mathematics and Computer Science, Eindhoven University of
Technology, Eindhoven, The Netherlands

Abstract. A crucial requirement for compliance-checking techniques is that observed behavior, captured in event traces, can be mapped to the process models that specify allowed behavior. Without a mapping, it is not possible to determine if observed behavior is compliant or not. A considerable problem in this regard is that establishing a mapping between events and process model activities is an inherently uncertain task. Since the use of a particular mapping directly influences the compliance of a trace to a specification, this uncertainty represents a major issue for compliance checking. To overcome this issue, we introduce a probabilistic compliance-checking method that can deal with uncertain mappings. Our method avoids the need to select a single mapping, but rather works on a spectrum of possible mappings. A quantitative evaluation demonstrates that our method can be applied on a considerable number of real-world processes where traditional compliance-checking methods fail.

Keywords: Compliance checking, event-to-activity mapping, process mining, matching, uncertainty

1 Introduction

Compliance management supports organizations by ensuring that their processes satisfy legal requirements and are executed in an efficient manner [27]. *Compliance-checking* techniques (cf. [3,20,26]) play an important role in this regard [17]. These techniques enable organizations to automatically check whether business processes are executed according to their specifications. Specifically, they check if any observed behavior, as recorded in an IT system and represented in the form of an *event trace*, conforms to the allowed process behavior, as captured in a *process model* [5]. A crucial requirement for compliance checking is that the events contained in an event log can be related to the activities of a process model [25]. Without knowing the relations between events and model activities, it is not possible to determine if the behavior within an event trace conforms to the behavior specified by a process model. Despite this dependence of compliance-checking techniques on the existence of such a, so-called, *event-to-activity mapping*, these mappings are often not readily available [8].

Furthermore, actually establishing event-to-activity mappings is a highly complex task. The effort required to *manually* perform this task is hardly manageable in practical scenarios, due to the task's combinatorial complexity [9]. *Automated* mapping techniques also face considerable challenges. These challenges are caused by, among others, cryptic event names, noncompliant behavior, and noise [7]. As a result, automated mapping techniques often cannot provide a certain solution to the mapping problem. In fact, the task of establishing event-to-activity mappings is conceptually equivalent to *matching* tasks found in the fields of *schema matching* and *process matching*. Such matching tasks have been shown to be inherently uncertain [14,28]. Due to this uncertainty, the goal of mapping techniques becomes choosing the *best* mapping from a number of potential ones [18]. Hence, there is always the risk that the selected mapping is wrong, i.e. that the selected mapping does not correctly capture the relations between event traces and a process model. In the context of compliance checking, selecting an incorrect mapping is particularly harmful. If the selected mapping is incorrect, the results obtained through compliance checking based on this mapping cannot be trusted.

To overcome this issue, this paper presents a compliance-checking method that can be applied in spite of an uncertain mapping of events onto activities. Our method assesses the compliance of a trace by considering the entire spectrum of potential mappings, rather than focusing on a single one. To capture this spectrum, we build on the notion of *probabilistic behavioral spaces*. These behavioral spaces provide a means to capture behavioral uncertainty, i.e. varying interpretations on described process behavior, in a structured manner. We originally introduced this notion to capture behavioral uncertainty caused by ambiguity in textual process descriptions [2]. We extend the original notion with probabilistic information in the current paper and apply it in the context of mapping uncertainty. These probabilistic behavioral spaces can be used for compliance checking without the need to resolve uncertainty, i.e. without the need to select a single event-to-activity mapping from a number of alternatives. As a result, our compliance-checking method avoids the risks associated with the selection of an incorrect mapping. A quantitative evaluation demonstrates that this method can be used to obtain comprehensive compliance-checking results for a considerably higher number of processes than traditional methods.

The remainder of this paper is structured as follows. Section 2 motivates the problem of compliance checking in the context of uncertain event-to-activity mappings. Then, Section 3 provides some necessary preliminary definitions. Section 4 describes our compliance-checking method. We evaluate the usefulness of our method in Section 5. Finally, we consider streams of related research in Section 6 and conclude the paper in Section 7.

2 Problem Illustration

In this section, we illustrate the problem of compliance checking in the context of mapping uncertainty. The goal of compliance checking is to determine if behavior captured in *event traces* is allowed by the behavior specified in the form of a *process model.* An event trace captures an execution sequence of events. These events correspond to the *actual* behavior of a process, because they are extracted from information systems that record the execution of process steps. By contrast, process models are used in compliance-checking scenarios to specify the *allowed* behavior of a process. A crucial prerequisite for compliance checking is that the events in event traces can be related to the activities of a process model. For example, given an event trace $t = < e_1, e_2, e_3, e_4, e_5 >$ and the process model M depicted in Figure 1, the events in t must be mapped to activities in model M. Otherwise, it is impossible to understand which activities have occurred in reality and, thus, whether or not t complies with M.

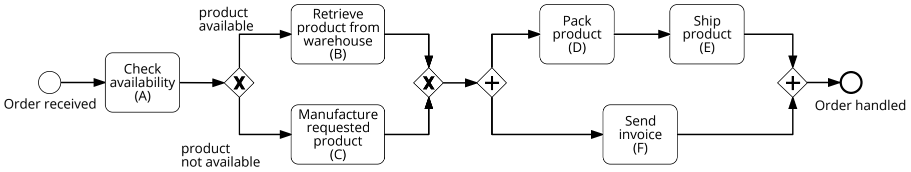

Fig. 1. Example of a BPMN process model

Unfortunately, establishing a correct mapping between events and activities is a considerable challenge. Existing techniques addressing this task can at best indicate potential mappings and their likelihoods, instead of providing a definite solution [8,9]. The reason why mapping techniques fail to provide definite solutions is that the information they can take into account when constructing mappings often does not suffice to identify relations with certainty. As an example, consider an event with the label "*Product obtained*". By considering this label, it is not possible to determine with certainty whether this event corresponds to activity B ("*Retrieve product from warehouse*") or to activity C ("*Manufacture requested product*"). Both of these activities *obtain* a product, but in a different way. Even more problematic are the commonly observed event labels with cryptic database field names such as *CDHDR* or *I_SM_E* [9]. In these cases, not even advanced linguistic analysis tools are able to identify reliable mappings.

The inability of techniques to reliably establish event-to-activity mappings leads to mapping uncertainty. As a result, mapping techniques generally construct a number of *potential* mappings without being able to determine with certainty which mapping is correct. Since existing compliance-checking techniques require a single event-to-activity mapping, mostly the mapping with the highest likelihood is selected as a basis for compliance checks. However, there is always the risk that this selected mapping is incorrect and that, consequently, compliance-checking results based on the selected mapping are incorrect as well.

To illustrate the risk of selecting a single mapping in the context of mapping uncertainty, assume that trace t corresponds to the activity sequence $\sigma =$

$< a, b, c, e, f >$. This means that t is *not* compliant with model M, because M does not allow for the activities B and C to be executed in the same process instance, while σ contains both of these. Further assume that, due to mapping uncertainty, a mapping technique returns two possible mappings, one corresponding to the *noncompliant* sequence σ, but the other to the *compliant* sequence $\sigma' =< a, b, d, e, f >$. In this scenario, the ability to correctly identify the noncompliance of t to M fully depends on the ability to select the appropriate mapping from the two alternatives. In case the mapping corresponding to σ' is selected, then t will be considered to be compliant with M, event though in reality the process behavior contained in t does *not* comply to the allowed behavior specified by M.

The previous example illustrates that compliance-checking results based on the selection of a single, potentially incorrect mapping are not trustworthy. To provide a comprehensive solution to this problem, this paper introduces a compliance-checking method that takes the entire set of potential mappings into account. Therefore, our method eliminates the need to select a single, possibly incorrect mapping. Hence, it mitigates the risk of drawing incorrect conclusions about process compliance.

3 Preliminaries

This section introduces the preliminaries on which we base our compliance-checking method. For the purposes of this paper, we use the *behavioral profile relations* from [24] to capture and compare behavior contained in event traces and process models. These behavioral relations build on a *weak order relation* $\succ$. For a single event trace $t =< e_1, \ldots, e_n >$ over a set of event classes E_t, the relation $\succ_t \subseteq (E_t \times E_t)$ contains all pairs $(x, y) \in (E_t \times E_t)$ such that there exist two indices $j, k \in 1, \ldots, m$ with $j < k \leq m$ for which holds that $e_j = x$ and $e_k = y$. Intuitively, the weak order relation contains any pair (x, y) for which an occurrence of event class x precedes an occurrence of event class y. A *behavioral profile* derives three distinct behavioral relations from this weak order relation: strict order, exclusiveness, and interleaving order. Definition 1 provides a formal definition for the behavioral profile of a single event trace.

Definition 1 (Behavioral Profile – Trace). *Let t be an event trace over a set of event classes E_t and with a weak order relation $\succ_t$. Then a pair of event classes $(x, y) \in E_t \times E_t$ is in at most one of the following relations:*

- *The* strict order relation $\rightsquigarrow_t$, *iff $x \succ_t y$ and $y \not\succ_t x$;*
- *The* exclusiveness relation $+_t$, *iff $x \not\succ_t y$ and $y \not\succ_t x$;*
- *The* interleaving order relation $||_t$, *iff $x \succ_t y$ and $y \not\succ_t x$;*

The set $BP_t = \{\rightsquigarrow_t, +_t, ||_t\}$ is the behavioral profile *of t.*

For a process model M, a behavioral profile BP_M is computed in a similar manner as for an event trace. The difference is that $\succ_M$ contains all pairs (x, y) for which there is an event trace t possible in M such that $(x, y) \in \succ_t$. Therefore, the behavioral profile of a process model builds on an aggregation of the weak order relation of all its possible traces. Definition 2 formally describes this.

Definition 2 (Behavioral Profile – Process Model). *Let M be a process model with an activity set A_M and with a weak order relation $\succ_M$. Then an activity pair $(x, y) \in A_M \times A_M$ is in at most one of the following relations:*

- *The* strict order relation $\rightsquigarrow_M$*, iff $x \succ_M y$ and $y \not\succ_M x$;*
- *The* exclusiveness relation $+_M$*, iff $x \not\succ_M y$ and $y \not\succ_M x$;*
- *The* interleaving order relation $||_M$*, iff $x \succ_M y$ and $y \not\succ_M x$;*

The set $BP_M = \{\rightsquigarrow_M, +_M, ||_M\}$ is the behavioral profile *of M.*

The behavioral profile relations form the basis of our compliance-checking method. Given an event trace t and a process model M, we can determine the compliance of t with M by comparing the relations in BP_t to those in BP_M. It is crucial to understand the different nature of the behavioral profile of a trace and of a process model. BP_t provides information on *observed* behavioral relations for a single trace, whereas BP_M describes *constraints* for these traces. Therefore, to perform a compliance check, we do not check if the behavioral relations in BP_t and BP_M are equal. Rather, we check if the relations in BP_t are allowed within the relations in BP_M. This can be achieved by considering the *subsumption* of behavioral profile relations, as introduced in [26]. The subsumption predicate $S(R, R')$ determines if a relation type R of a process model subsumes a relation R' of a trace. $S(R, R')$ is defined as given by Definition 3. In this definition, the short-hand notation $x \rightsquigarrow^{-1} y$ is used to denote that $y \rightsquigarrow x$.

Definition 3 (Subsumption Predicate). *Given two behavioral relations $R, R' \in \{\rightsquigarrow, \rightsquigarrow^{-1}, +, ||\}$, the* subsumption predicate *$S(R, R')$ is satisfied iff $(R \in \{\rightsquigarrow, \rightsquigarrow^{-1}\} \wedge R' = +)$ or $R = R'$ or $R = ||$.*

Intuitively, the notion of subsumption builds on the different *strengths* of behavioral profile relations. For example, due to parallelism in the model M of the running example, the behavioral profile of M contains the relation $D \ || \ F$. However, in the behavioral profile of a trace, parallelism cannot be observed, because only a single execution of each of these activities should occur, e.g. $t =< D, F, E >$. Therefore, BP_t contains the relation $D \rightsquigarrow F$. Even though the two behavioral profile relations are not equal, it is clear that t does not violate the constraints expressed by M, because $D \rightsquigarrow F$ is a valid order in which D and F can be executed. This compliance is accounted for by the subsumption predicate, since the predicate $S(||, \rightsquigarrow)$ is satisfied. Similarly, an exclusion relation $c + d$ in a trace does not violate a strict order relation $c \rightsquigarrow d$ in a model.

A trace t is compliant with a process model M if all behavioral profile relations in BP_t are subsumed by the relations in BP_M. Definition 4 captures this for the situation when a mapping between the events of t and activities of M is known.

Definition 4 (Trace to Process Model Compliance). *Let M be a process model with an activity set A and $t =< e_1, \ldots, e_n >$ an event trace containing* *the activities* $A_t \subseteq A$*. Trace* t *complies with process model* M *if for each activity pair* $(x, y) \in (A_t \times A_t)$ *the relation in* t *is subsumed by the relation in* M*, i.e. the* compliance predicate *compl*(t, M) *is satisfied* iff $\forall R \in BP_t \cup \{\rightsquigarrow_t^{-1}\}, BP_M \cup \{\rightsquigarrow_I^{-1}\}$*, it holds* $(xRy \wedge xR'y) \implies S(R, R')$.

Next, we describe our compliance-checking method that employs the compliance notion provided by Definition 4.

4 Compliance-Checking Method

This section describes our method for compliance checking in the context of uncertain event-to-activity mappings. The two-step method we propose takes as input an event trace t, a process model M, and an uncertain mapping between the events of t and the activities M. Note that the question of how to obtain the mapping is not the focus of this paper, but it can be determined using techniques from e.g. [7,8,23]. In the first step, the method uses the uncertain mapping to generate a so-called *probabilistic behavioral space* for t. In the second step, we use this probabilistic behavioral space to perform a compliance check. In the remainder of this section, we describe the relevant concepts and steps of our method in detail.

4.1 From Uncertain Mapping to Probabilistic Behavioral Space

In the first step of our method, we generate a probabilistic behavioral space for an event trace. The notion of probabilistic behavioral spaces, in the remainder also simply referred to as *behavioral spaces*, provides the foundation to reason about process compliance in the context of uncertain event-to-activity mappings. The idea underlying this notion is that an uncertain event-to-activity mapping results in multiple views on what process behavior, in terms of process model activities, is described by a single event trace. A probabilistic behavioral space captures these views in a structured manner. To describe the generation of behavioral spaces, we first define *regular* and *uncertain* event-to-activity mappings.

For a given trace $t =< e_1, \ldots, e_n >$ over a set of event classes E_t and a process model M with an activity set A_M, we use $EA(t, M)$ to denote a single event-to-activity mapping between the events in t and the activities in A_M. The mapping $EA(t, M)$ consists of a number of *correspondences* between individual events and activities. Each correspondence $e \sim a \in (E_t \times A_M)$ denotes a mapping relation between an event e and an activity a. For example, given a trace $t =< e_1, e_2, e_3 >$ a mapping $EA(t, M) = \{e_1 \sim a, e_2 \sim b, e_3 \sim c\}$ indicates that trace t corresponds to the execution of the sequence of process model activities $< a, b, c >$. We shall refer to such a sequence of process model activities as a *trace translation* of event trace t, because it represents a translation of the trace's events into process model activities. Definition 5 formalizes this notion. Note that, for the sake of readability, we here focus on one-to-one relations between events and activities in a trace translation. However, our compliance-checking method also works on trace translations which are based on one-to-many or many-to-many mappings between events and activities.

Definition 5 (Trace translation). *Given an event trace* $t =< e_1, \dots, e_n >$ *with a set of event classes* E_t*, a process model* M *with an activity set* A_M*, and an event-to-activity mapping* $EA(t, M) \subseteq (E_t \times A_M)$ *we define a* trace translation *as* $\sigma(t) =< a_1, \dots, a_n >$*, where for each* $0 < i \leq n$*, it holds that* $e_i \sim a_i \in EA(t, M)$*.*

We use $\mathbb{EA}(t, M)$ to denote an uncertain event-to-activity mapping between an event trace t and a process model M. $\mathbb{EA}(t, M)$ consists of a number of event-to-activity mappings, where each $EA_i \in \mathbb{EA}$ represents a potential way to map the events in t to the activities in A_M. Therefore, each mapping $EA_i \in \mathbb{EA}$ yields a different trace translation for t. For instance, in Section 2, we described an example where a single trace had two trace translations, $\sigma = < a, b, c, e, f >$ and $\sigma' =< a, b, d, e, f >$. Together, these translations represent the spectrum of process behavior that might be contained in t, i.e. the behavioral space of an event trace. Since each mapping can be associated with a probability, we include a probabilistic component in our definition of a behavioral space, as captured in Definition 6.

Definition 6 (Probabilistic Behavioral Space). *Given an event trace* $t = < e_1, \dots, e_n >$ *with a set of event classes* E_t*, a process model* M *with an activity set* A_M*, and an uncertain event-to-activity mapping* $\mathbb{EA}(t, M)$*, we define a* probabilistic behavioral space *as a tuple* $PBS_t = (\Sigma(t), \phi)$*, with:*

- $\Sigma(t)$*: the set of* trace translations *of trace t over the activity set* A *as given by the event-to-activity mappings in* $\mathbb{EA}(t, M)$*;*
- $\phi : \Sigma_t(A) \to [0, 1]$*: a function that assigns a probability to each trace translation in* $\Sigma(t)$*.*

The set $\Sigma(t)$ comprises the set of potential trace translations of trace t over the activity set A, where each translation $\sigma_i \in \Sigma(t)$ is based on a mapping $EA_i \in \mathbb{EA}(t, M)$. The probability function ϕ assigns a probability p_i to each translation $\sigma_i(t) \in \Sigma(t)$. These probabilities can generally be based on the confidence of an event-to-activity mapping technique. For instance, a technique based on the semantic similarity scores, such as [8], can quantify the probability as the product of the similarity scores associated with each correspondence in the trace translation. If no such probabilities are available, the most straightforward solution is to assign an equal probability $p_i = 1 \ / \ |\Sigma_t|$ to each translation.

4.2 Using Behavioral Spaces for Compliance Checking

In this section, we illustrate the usefulness of probabilistic behavioral spaces for compliance checking in the context of uncertain event-to-activity mappings. The goal of compliance checking is to determine if the behavior in a trace t is allowed by the behavioral specification of a process model M. Since uncertain event-to-activity mappings lead to multiple views on the process model behavior contained in a trace (i.e. its trace translations), different translations can lead to different compliance-checking results. By using probabilistic behavioral spaces, we can perform compliance checks in spite of such different translations. In the remainder of this section, we demonstrate how to perform compliance checking using behavioral spaces by introducing a probabilistic compliance measure. Furthermore, we discuss the valuable diagnostic information that these compliance checks can provide.

To perform our compliance checks, we introduce a compliance metric that quantifies the compliance of a probabilistic behavioral space PBS_t to a process model M. The metric combines the compliance assessments for individual trace translations with probabilistic information. The metric determines for each trace translation $\sigma \in \Sigma(t)$ in a behavioral space whether it is compliant or not. This is achieved by computing the behavioral profile BP_σ for a trace translation σ as described in Section 3. Since a trace translation contains a subset of the activities of a process model, we can proceed to determine if σ complies with a model M by comparing BP_σ with BP_M according to Definition 4. By taking the sum of the probabilities associated with all compliant translations, we obtain the probability that a trace t is compliant with a model M. Definition 7 formalizes this metric.

Definition 7 (Behavioral Space Compliance). *Let t be a trace with a probabilistic behavioral space $PBST(t) = (\Sigma(t), \phi)$ and BP_M a behavioral profile for a process model M with activity set A_M. Then we define:*

- *$\Sigma_C(t) \subseteq \Sigma(t)$ as the set of trace translations in $\Sigma(t)$ compliant with BP_M;*
- *$ProbCompl(t, M) = \sum_{\sigma \in \Sigma_C(t)} \phi(\sigma)$: as the* behavioral space compliance *of trace t to model M, where $\phi(\sigma)$ captures the probability of translation σ.*

Two interesting properties of this compliance metric are worth considering in more detail. First, when compared to traditional compliance checking, the metric provides probabilistic instead of binary results. In traditional compliance scenarios, i.e. without uncertainty, a trace is either compliant or noncompliant. In the scenario with uncertainty, traces are either compliant, noncompliant, or *potentially compliant.* Figure 2 visualizes this.

Potentially compliant traces are those traces for which some trace translations are compliant with a process model, whereas others are noncompliant. The compliance of these traces is associated with a certain probability $0 < p < 1$. Take, for instance, the trace $t_1 =< e_1, e_2, e_3, e_4, e_5 >$ and its two translations from the running example, $\sigma_1(t_1) =< a, b, d, e, f >$ and $\sigma_2(t_1) = < a, b, c, e, f >$. Assume that $\sigma_1(t_1)$ is associated with a probability of 0.8 and $\sigma_2(t_1)$ with probability 0.2. Given that $\sigma_1(t_1)$ is compliant with M and $\sigma_2(t_1)$ is noncompliant, the trace t_1 is potentially compliant with M, with a probability of 0.8. Therefore, we know that t_1 is is more likely to be compliant than not. Furthermore, we also know the mapping conditions under which t_1 is compliant or noncompliant. Namely, t_1 is compliant if the correspondence $e_3 \sim D$ holds, whereas the trace

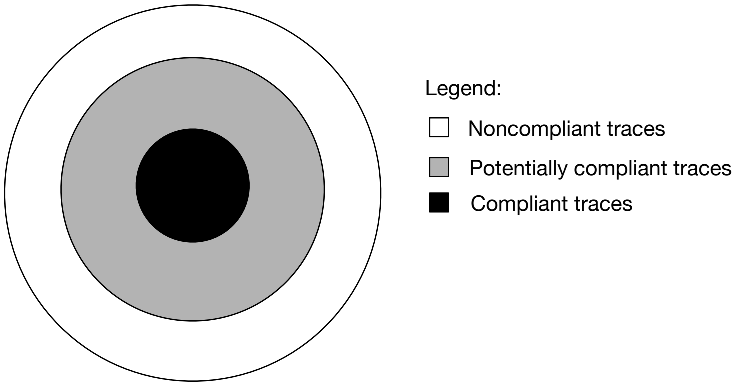

Fig. 2. Three types of compliance assessments for probabilistic compliance checking.

is noncompliant if $e_3 \sim C$ is true. This is the kind of diagnostic information we referred to earlier, which can be useful because it provides insights into which aspects of an uncertain mapping lead to uncertainty in compliance-checking results for observed behavior.

The second interesting property of the compliance metric is that, despite its probabilistic nature in the presence of mapping uncertainty, the metric often still yields non-probabilistic results. To illustrate this, consider a (partial) trace $t_2 =< e_1, e_2, e_3 >$ with translations $\sigma_1(t_2) = < b, a, d >$ and $\sigma_2(t_2) = < b, c, d >$. Although mapping uncertainty has resulted in two trace translations, $ProbCompl(t_2, M)$ yields a non-probabilistic results since neither of the translations is compliant with model M. Therefore, it is certain that t_2 is noncompliant. In a similar fashion, a partial trace $t_3 =< e_1, e_2, e_3 >$, with translations $\sigma_1(t_3) =< a, b, d >$ and $\sigma_2(t_3) = < a, c, d >$ can be said to be compliant with certainty. No matter if e_2 corresponds to activity B or C, the trace is compliant. Such cases occur in particular when activities are *behaviorally equivalent* compared to each other. In this case, B and C have such equivalence, because they present proper alternatives for each other.

The previous example illustrates that our compliance-checking method can be used to determine compliance with certainty in situations where traditional compliance-checking methods would not be able to make trustworthy compliance assessments. In Section 5, we demonstrate the usefulness of this property in practical settings.

5 Evaluation

In this section, we present an evaluation that we conducted to demonstrate the capabilities of the proposed compliance-checking method for uncertain event-to-activity mappings. The goal of this evaluation is to assess how the impact of mapping uncertainty on the compliance-checking task can be reduced by using our method. To achieve this, we compare results obtained through our method against results obtained by using a traditional compliance-checking method. We apply these methods on a collection of real-world process models and accompanying event logs. Specifically, we evaluate for how many traces in these event logs the two methods can provide compliance-checking results with certainty.

5.1 Test Collection

To perform the evaluation, we use a collection of real-world business process models from the *BIT process library*, first analyzed in an academic context by Fahland et al. [13]. The BIT process library consists of 886 process models from various industries, including the financial services and telecommunications domains. The same collection that has been used to test several event-to-activity mapping approaches [7,9], which motivates our choice for it. Hence, we believe that results obtained by using this collection present a realistic view on the applicability of the event-to-activity mapping approach against which we compare our compliance-checking method. Furthermore, due to the size of the collection and its broad coverage of real-world process models, the collection seems well-suited to achieve a high external validity of the results.

From the test collection, we omitted any process model with soundness issues such as *deadlocks* or *livelocks*. Furthermore, we omitted a number of large models for which the event-to-activity mapping approach was not able to produce a results due to memory shortage. Note that the same filtering steps are also applied in [7]. As a result of the filtering, a collection of 598 process models remains available for usage in our evaluation.

5.2 Setup

Figure 3 depicts the steps of our evaluation approach. To perform these steps, we employ the *ProM6 framework*, which provides a vast amount of so-called *plug-ins* that implement process mining techniques.[3] In the first two steps of our approach we build on existing plug-ins for event-to-activity mapping techniques, as described in [7]. For step 3 and 4, we have implemented the generation of behavioral spaces and our proposed method for compliance checking as a plug-in, which is available as part of the *BehavioralSpaces* package in ProM6.

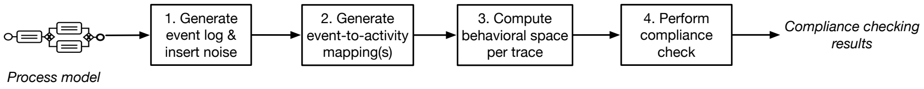
Process model
1. Generate event log & insert noise
2. Generate event-to-activity mapping(s)
3. Compute behavioral space per trace
4. Perform compliance check
Compliance checking results

Fig. 3. Overview of the evaluation setup

Step 1 of the evaluation approach first generates an event log or each of the 598 process models in the filtered test collection. Staying true to the evaluation

[3] See www.promtools.org for more information and to download the framework.

of [7], we generate a log containing 1000 traces for each model. For process models that include loops, we generate traces with a maximum length of 1000 events. Since we are interested in compliance checking, we transform these fully compliant logs into partially non-compliant logs. We achieve this by using a *noise-insertion* plug-in in ProM.[4] This plug-in randomly adds noise to a log (i.e. possible noncompliance) by shuffling, duplicating, and removing events for a given percentage of traces. In this manner, we generate six different event logs, respectively containing noise in 0, 20, 40, 60, 80, and 100% of the traces.

In step 2, we take a process model and an accompanying event log and use the mapping technique from [7] to establish an event-to-activity mapping. We have selected this particular technique because it returns all potential mappings in case of uncertainty. Furthermore, the technique is relatively robust in the context of noncompliant behavior. In case the approach can compute a single mapping, i.e. there is no mapping uncertainty, we can conclude that for this process model and event log, traditional compliance-checking techniques suffice to determine the compliance of all traces in the log. If the mapping approach returns multiple possible mappings, i.e. there is mapping uncertainty, we continue with the third step of the evaluation.

Step 3 computes a behavioral space for a trace based on an (uncertain) event-to-activity mapping $\mathbb{EA}$ established in the previous step. We obtain a behavioral space by creating a trace translation for each of the potential event-to-activity mappings included in $\mathbb{EA}$.

Lastly, in step 4 we assess if we can determine the compliance or noncompliance of a trace despite the presence of mapping uncertainty. We achieve this by computing the $ProbCompl$ metric for the behavioral space of a trace t. If this metric returns a compliance level of 0.0 or 1.0, we know if t is compliant or not with certainty. For other values, also the consideration of behavioral spaces does not suffice to determine the compliance in a certain manner. Nevertheless, we still obtain probabilistic results and diagnostic information for these cases.

5.3 Results

Figure 4 presents the results of our evaluation experiments. The figure illustrates for what percentage of traces deterministic compliance-checking results are obtained by our proposed method and traditional methods.

For noise level 0, where all traces in the event logs are compliant with the process models, we can observe that the mapping approach establishes an event-to-activity mapping for 70.2% of the models in the collection. Since all of the traces are compliant in this log, these issues are caused by activities which are behaviorally identical to each other. An example of this is seen for activities B and C of the running example. Because of these issues, traditional compliance-checking techniques cannot assess the compliance of 29.8% of the traces. However, by using behavioral spaces, we can still determine the compliance of a trace when mapping uncertainty is caused by such behavioral equivalent activities.

[4] Provided by the *Event2ActivityMatcher* package.

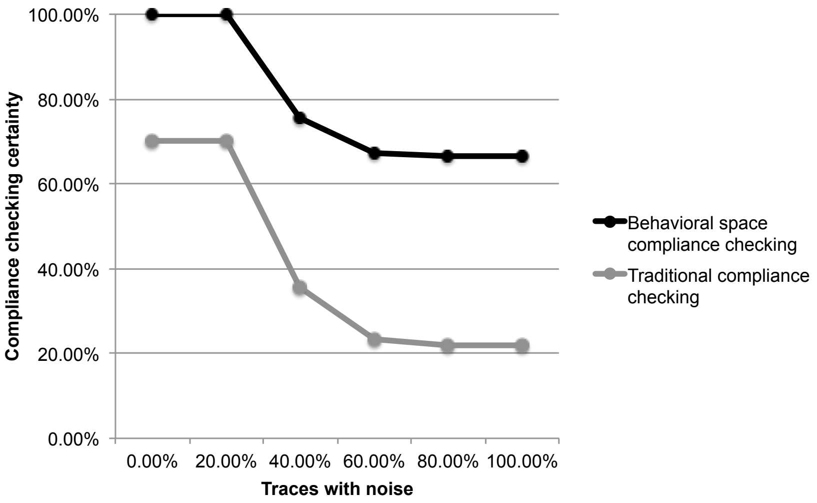

Fig. 4. Overview of the evaluation results

Hence, by using our proposed compliance-checking method, we can determine the compliance of traces with certainty in 100% of the cases. Due to its relative robustness to noise, the mapping approach obtains the same results for logs in which 20% of the traces contain noisy behavior. Therefore, the results obtained by our method are equal for this set of logs.

The results change for higher noise levels. For these sets of logs, the mapping approach fails to establish certain event-to-activity mappings for increasing numbers of processes. At 40% noise, the approach fails to establish certain mappings for 62.2% of the processes. This means that traditional compliance-checking techniques can only make compliance assessments in 36.8% of the cases. By contrast, our compliance-checking method still succeeds to determine the compliance of 75.5% of the traces with certainty. The gap between our compliance-checking method and traditional methods is even bigger for noise levels of 60% and higher. As Figure 4 illustrates, the performance of the mapping approach and, thus, also of both compliance-checking methods stabilizes for these noise levels. However, traditional compliance-checking methods can only determine compliance for approximately 22.0% of the traces. By contrast, our proposed compliance-checking method still provides deterministic results for 66.4% of the traces, i.e. for 3 times as many traces.

In summary, traditional compliance-checking techniques become less and less useful. For high noise levels, they can provide results for as little as 22.0% of the traces. While the certainty obtainable through compliance checking with behavioral spaces is also affected by the increased levels of noise, the impact is much smaller. Therefore, we can conclude that in practical scenarios our compliance-checking method is much wider applicable than traditional compliance-checking methods. Furthermore, even in cases where also our method can provide compliance-checking results with certainty, our method still provides useful probabilistic and diagnostic compliance information.

6 Related Work

The work presented in this paper primarily relates to two major research streams: process matching and conformance checking.

Techniques for *process matching* concern the establishment of links between process concepts in different artifacts. The most commonly addressed use case for this is process model matching, where links are established between activities in different process models [10]. So-called process model matchers address this task by exploiting different process model features, including natural language [12], model structure [11], and behavior [15]. Therefore, they use similar techniques as the works, considered throughout this paper, that relate events to process model activities. Similar to the event-to-activity mapping task, it has been found that model-model matching is also inherently uncertain [16]. Other process matching techniques focus on different use cases, such as the alignment of natural language texts to process models [1] and the alignment of events from different event logs [19].

Process compliance checking techniques are applied in various application scenarios, including process querying [6], legal compliance [22], and auditing [4]. A plethora of techniques exist for this purpose (cf. [3,5,20,21]). In this paper, we have used techniques that perform compliance checks based on behavioral profile relations, introduced in [26]. These techniques are computationally highly efficient, which makes them an ideal choice for compliance checking in the context of the potentially vast number of translations per trace. Other commonly used techniques perform compliance checks based on so-called *alignments*. These techniques, introduced in [3,5], provide different diagnostic information than compliance checks based on behavioral profiles. Furthermore, the compliance checks can be considered to be more accurate in certain situations, because behavioral profile relations abstract from some details of process behavior. However, these techniques are computationally much more demanding than the highly efficient compliance checks based on behavioral profiles.

7 Conclusion

In this paper, we introduced a compliance checking method that can be used in the presence of uncertain event-to-activity mappings. Our method considers all potential mappings generated by automated mapping approaches. As such, it can provide compliance checking results without the need to select a single, possibly incorrect mapping to base compliance checks on. Therefore, it avoids the risk of drawing incorrect compliance conclusions. A quantitative evaluation based on a large collection of real-world process models demonstrated that our method can provide deterministic compliance checking results for a considerable amount of situations where traditional compliance checking methods fail.

Our proposed method has to be considered in light of a considerable limitation. Namely, the obtained compliance checking results are dependent on the quality of the generated event-to-activity mappings. Most importantly, its results can be negatively affected if the correct mapping is not included in the set of potential mappings generated by any approach. Still, by applying our method, we eliminate the need to select a mapping from the set of potential methods. Hence, our method significantly reduces the possibility of drawing incorrect conclusions.

In future work, we intend to extend the coverage of our compliance checking method. For example, we want to provide instantiations based on other notions of compliance, such as alignment-based compliance or by considering data associated with events. Furthermore, we want to investigate possibilities to use our compliance checking method to improve existing event-to-activity mapping techniques or to support selection among potential mappings.

References

1. Van der Aa, H., Leopold, H., Reijers, H.A.: Comparing textual descriptions to process models – the automatic detection of inconsistencies. Information Systems (in press) (2016)
2. Van der Aa, H., Leopold, H., Reijers, H.A.: Dealing with behavioral ambiguity in textual process descriptions. In: International Conference on Business Process Management. pp. 271–288. Springer (2016)
3. Van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Replaying history on process models for conformance checking and performance analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2(2), 182–192 (2012)
4. Accorsi, R., Stocker, T.: On the exploitation of process mining for security audits: the conformance checking case. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. pp. 1709–1716. ACM (2012)
5. Adriansyah, A., van Dongen, B., van der Aalst, W.: Conformance checking using cost-based fitness analysis. In: Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International. pp. 55–64. IEEE (2011)
6. Awad, A., Decker, G., Weske, M.: Efficient compliance checking using bpmn-q and temporal logic. In: International Conference on Business Process Management. pp. 326–341. Springer (2008)
7. Baier, T., Di Ciccio, C., Mendling, J., Weske, M.: Matching of events and activities-an approach using declarative modeling constraints. In: International Conference on Enterprise, Business-Process and Information Systems Modeling. pp. 119–134. Springer (2015)
8. Baier, T., Mendling, J., Weske, M.: Bridging abstraction layers in process mining. Information Systems 46, 123–139 (2014)
9. Baier, T., Rogge-Solti, A., Weske, M., Mendling, J.: Matching of events and activities-an approach based on constraint satisfaction. In: IFIP Working Conference on The Practice of Enterprise Modeling. pp. 58–72. Springer (2014)
10. Cayoglu, U., Dijkman, R.M., Dumas, M., Fettke, P., Garcıa-Banuelos, L., Hake, P., Klinkmüller, C., Leopold, H., Ludwig, A., Loos, P., et al.: The process model matching contest 2013. In: 4th International Workshop on Process Model Collections: Management and Reuse (PMC-MR'13) (2013)

11. Dijkman, R.M., Dumas, M., Van Dongen, B.F., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems 36(2), 498–516 (2011)
12. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring similarity between semantic business process models. In: Proceedings of the fourth Asia-Pacific conference on Comceptual modelling-Volume 67. pp. 71–80 (2007)
13. Fahland, D., Favre, C., Koehler, J., Lohmann, N., Völzer, H., Wolf, K.: Analysis on demand: Instantaneous soundness checking of industrial business process models. Data & Knowledge Engineering 70(5), 448–466 (2011)
14. Gal, A.: Uncertain schema matching. Synthesis Lectures on Data Management 3(1), 1–97 (2011)
15. Künze, M., Weidlich, M., Weske, M.: Behavioral similarity–a proper metric. In: Business Process Management, pp. 166–181. Springer (2011)
16. Kuss, E., Leopold, H., Van der Aa, H., Stuckenschmidt, H., Reijers, H.A.: Probabilistic evaluation of process model matching techniques. In: The 35th International Conference on Conceptual Modeling (ER (2016)
17. Ly, L.T., Maggi, F.M., Montali, M., Rinderle-Ma, S., van der Aalst, W.M.P.: Compliance monitoring in business processes: Functionalities, application, and tool-support. Information systems 54, 209–234 (2015)
18. Madhavan, J., Bernstein, P.A., Domingos, P., Halevy, A.Y.: Representing and reasoning about mappings between domain models. AAAI/IAAI 2002, 80–86 (2002)
19. Mannhardt, F., de Leoni, M., Reijers, H.A.: Extending process logs with events from supplementary sources. In: 3rd Workshop on Data- & Artifact-centric BPM. pp. 235–247. No. LNBIP, Springer (2014)
20. Munoz-Gama, J., Carmona, J., Van Der Aalst, W.M.: Single-entry single-exit decomposed conformance checking. Information Systems 46, 102–122 (2014)
21. Ramezani, E., Fahland, D., Van der Aalst, W.M.P.: Where did i misbehave? diagnostic information in compliance checking. In: International conference on business process management. pp. 262–278. Springer (2012)
22. Sadiq, S., Governatori, G., Namiri, K.: Modeling control objectives for business process compliance. In: International conference on business process management. pp. 149–164. Springer (2007)
23. Senderovich, A., Rogge-Solti, A., Gal, A., Mendling, J., Mandelbaum, A.: The road from sensor data to process instances via interaction mining. In: International Conference on Advanced Information Systems Engineering. pp. 257–273. Springer International Publishing (2016)
24. Smirnov, S., Weidlich, M., Mendling, J.: Business process model abstraction based on behavioral profiles. In: Service-Oriented Computing, pp. 1–16. Springer (2010)
25. Van Der Aalst, W., Adriansyah, A., De Medeiros, A.K.A., Arcieri, F., Baier, T., Blickle, T., Bose, J.C., et al.: Process mining manifesto. In: International Conference on Business Process Management. pp. 169–194. Springer (2011)
26. Weidlich, M., Mendling, J., Weske, M.: Efficient consistency measurement based on behavioral profiles of process models. IEEE Transactions on Software Engineering 37(3), 410–429 (2011)
27. Weidlich, M., Polyvyanyy, A., Desai, N., Mendling, J., Weske, M.: Process compliance analysis based on behavioural profiles. Information Systems 36(7), 1009–1025 (2011)
28. Weidlich, M., Sagi, T., Leopold, H., Gal, A., Mendling, J.: Predicting the quality of process model matching. In: Business Process Management, pp. 203–210. Springer (2013)