Conformance checking: A state-of-the-art literature review

Sebastian Dunzer
sebastian.dunzer@fau.de
Friedrich-Alexander Universität Erlangen-Nürnberg
Nürnberg, Germany

Martin Matzner
martin.matzner@fau.de
Friedrich-Alexander Universität Erlangen-Nürnberg
Nürnberg, Germany

Matthias Stierle
matthias.stierle@fau.de
Friedrich-Alexander Universität Erlangen-Nürnberg
Nürnberg, Germany

Stephan Baier
stephan.baier@fau.de
Friedrich-Alexander Universität Erlangen-Nürnberg
Nürnberg, Germany

ABSTRACT
Conformance checking is a set of process mining functions that compare process instances with a given process model. It identifies deviations between the process instances’ actual behaviour (“as-is”) and its modelled behaviour (“to-be”). Especially in the context of analyzing compliance in organizations, it is currently gaining momentum – e.g. for auditors. Researchers have proposed a variety of conformance checking techniques that are geared towards certain process model notations or specific applications such as process model evaluation. This article reviews a set of conformance checking techniques described in 37 scholarly publications. It classifies the techniques along the dimensions "modelling language", "algorithm type", "quality metric", and "perspective" using a concept matrix so that the techniques can be better accessed by practitioners and researchers. The matrix highlights the dimensions where extant research concentrates and where blind spots exist. For instance, process miners use declarative process modelling languages often, but applications in conformance checking are rare. Likewise, process mining can investigate process roles or process metrics such as duration, but conformance checking techniques narrow on analyzing control-flow. Future research may construct techniques that support these neglected approaches to conformance checking.

CCS CONCEPTS
• Applied computing → Business process management; Business process modeling; Business intelligence; • Information systems → Data mining.

KEYWORDS
process mining, conformance checking, modelling languages, business process management, literature review

1 INTRODUCTION
The Association of Certified Fraud Examiners (ACFE) assembles the Report to the Nations once a year. The report’s 2018 edition [1] investigates 2,690 cases of professional fraud that were reported between January 2016 and October 2017 in 125 countries. In the average case, an affected company loses about five percent of its annual revenues to occupational frauds and scams. More than a fifth of all reported cases have caused losses of more than one million US dollars. According to the ACFE, either the lack of or the weak implementation of internal fraud control systems (i.e. four-eyes principle) leverages fraudulent behaviour in about half of all reported cases [1].

All these cases illustrate the need for instruments that support organizations in creating transparency over their internal processes and thereby in identifying irregularities in business process execution and management. Therefore, it is not surprising that conformance checking, one branch of process mining, is gaining momentum in research domain and in practice [22, 43]. Conformance checking techniques compare the behaviour of process instances recorded in an event log with a process model and analyze deviations of the process behaviour [43]. Thus, conformance checking measures the deviations of process executions from their normative or descriptive behaviour. As a result, business process managers can either determine whether their processes run as intended or at least as described in a process model. The conformance checking literature is rich on studies linking process models to process data, including the seminal work of Rozinat and van der Aalst [39] and more recent studies as for instance Burattin et al. [12]. We consider several articles as related works which we divide into three different groups. First, there are several articles that provide literature reviews of process mining projects in a specific field, such as oncology [27], primary care [52], and health care [37]. As all these research projects take place in the health care domain and additionally set the broader term ‘process mining’ as subject to the analysis instead of specializing in conformance checking, the results of our article strongly differ from these previous research projects. Third, Rozinat’s [38] dissertation contains one chapter that deals with conformance measurement using Petri nets. Whereas the author limits the conformance checking overview to Petri nets, we do
2 CONFORMANCE CHECKING OVERVIEW

Process Mining is the area within the research field of business process management that deals with the analysis of event data that several information systems generate during the execution of processes. It consists of three distinct sets of functions including discovery, conformance checking, and enhancement [43]. Discover
ey; the probably most common process mining function, aims to generate as-is process models from event data automatically. Conformance checking compares event data that the process execution produces to process models that define their normative or descriptive behavior. Last, enhancement includes all functions of process mining that enhance either a process model or an event log. In the remainder, we solely focus on the second set of functions, conformance checking.

As an input, conformance checking methods require an event log and a technical process description. For the scope of the present literature review, we focus on process models with a graphical representation (i.e. Petri nets) as technical process descriptions. The description of process behavior in line with the specifications of a particular process modelling language creates a process model. Current research distinguishes between two types of process modelling languages. These are, first, the declarative (defining behavior that is not allowed) and, second, the procedural (i.e. defining behavior that is allowed) types of process modelling. The latter type includes popular modelling languages such as Event-driven Process Chains (EPC), Unified Modelling Language (UML) activity diagrams, Business Process Model and Notation (BPMN) and Petri nets [36]. Representatives of the declarative modelling languages are Declare, Dynamic Condition Response (DCR) Graphs, Case Management Model and Notation (CMMN) [40]. The event log is the second input for conformance checking. Since the data of events must be relatable to their respective process instance, a clear consensus towards the event log structure exists. Its minimum requirements of content are a unique case identifier, an activity label and a time-stamp [42, p.98]. If these content requirements are not fulfilled, conformance checking cannot be performed. Furthermore, event logs may include additional information on further process perspectives, like resources, costs, duration, etc., that conformance checking techniques can utilize for deeper analysis. Hence, the perspectives that a conformance checking technique considers is of high relevance.

After having determined a modelling notation and the scope of the event log (control-flow or multi, i.e. further perspectives), researchers dealing with conformance checking need to choose or develop an algorithm to compare the model to the event log. Carmona et al. [13] and van der Aalst [43] describe two general approaches to such algorithms which are log replay algorithms and trace alignment algorithms. Log replay algorithms firstly interpret the model and the log and secondly re-run every trace, event by event, on the model. Distinct computing techniques can thereafter determine a conformance metric. One example for such a conformance checking technique is the token-based log replay...
presented in [39]. Every time the model reaches a dead-end during the execution, before it is allowed to terminate, an additional token is generated to advance the current state of the respective model to the next state. Also, the approach sums up tokens that remained in the model after the log trace has terminated. In the end, the algorithm determines the process conformance based on the sum of all superfluous and generated tokens. Note that this is only one representative example from the variety of existing log-replay algorithms.

In contrast to log replay algorithms, trace alignments can additionally express deviations and conformance on event level. Figure 1 shows the formal notation of trace alignments [43].

\[
y_1 = \frac{a}{a} g c f e h, y_2 = \frac{a}{a} b d e h
\]

**Figure 1: Examples for trace alignments**

The upper section of an alignment expresses the executed steps in an event log and the lower section shows an aligned trace from the model. Whereas \( y_1 \) shows one trace alignment that ideally aligns the logged and the modelled trace, \( y_2 \) contains discrepancies between log and model. The ‘\( >> \)’ indicates that either the log could not make the same step as in the model or vice versa. By computing such trace alignments, process analysts get various insights in violating log and model traces, as well as the responsible event occurrences. Each asynchronous move in the log or model is related to a cost value to retrieve so-called optimal alignments. Subsequently, the minimum of a cost-function determines optimal alignments by choosing the cheapest alignment of model and log [2].

Concerning the output of a conformance checking technique, researchers need to decide what metrics they want to use to express conformance. Previous research suggests four quality metrics to measure conformance. *Fitness* expresses the ratio of traces in an event log that a process model can repeat. While a *fitness* of 1.0 indicates that a process model faultlessly reproduces every trace in the event log, a value of 0.0 means that the model cannot repeat any of the real-life cases. *Simplicity* is the quality metric which is often explained with the principle of Occam’s razor. It says, the simpler the model is, the better it is. This indicator is useful when the subject of analysis is to find the best model among different process models for the same process. The metric *precision* indicates how well a model represents the behavior seen in a log. *Precision* conformance checking techniques determine superfluous activities and connections to quantify precision. In contrast, *generalization* implies whether the process model represents the process paths in a generic way or only the behavior observed in the data. One example for a too general model is the so-called flower Petri net model. Figure 2 displays the concurrency between the generalization metric represented by the flower model (F) and the over-precise model (P) for the same process. The model F can execute any activity any time, whereby it expresses any process behavior. However, it does not seem to be a good model for a sequential process [6, 43, 46]. In contrast, model P depicts every conforming execution path separately and thereby becomes more incomprehensible. As a consequence, researchers must find a balance between generalization and precision for good process models.

![Underfitting “Flower” (F) Overfitting “Precise” (P)](image)

Looking at the use of conformance checking in practice, *fitness* is certainly the most used metric as it states how well an observed behavior fits the defined process model, i.e. to evaluate the quality of the executed work. When making statements about the quality of the defined process model process analysts prefer the metrics *precision, generalization* and *simplicity* [45].

The analysis of in-, through- and output of the basic conformance checking literature leads to the following dimensions: (1) *Modelling language*, (2) *Perspective*, (3) *Algorithm type*, and (4) *Quality metric*. These four dimensions categorize all of the conformance checking concepts mentioned above.

### 3 RESEARCH METHOD

We conduct a systematic literature review to provide an overview of the state-of-the-art in conformance checking. The present review grounds on the eight-step guideline delivered by Okoli and Schabram [34]. These steps are: (1) purpose of the literature review, (2) protocol and training, (3) searching for the literature, (4) practical screen, (5) quality appraisal, (6) data extraction, (7) synthesis of studies, and (8) writing the review. Additionally, we take Webster and Watson into consideration as we use a concept matrix as a general framework for the presentation of our results [51].

![Figure 2: Contrast between generalization and precision](image)

![Figure 3: Presentation of literature search process](image)

The introduction section motivates and explains the *purpose of the literature review*. To conduct the step protocol and training, we...
We consider the concept matrix as our 37 articles are clustered regarding their relation to the concept dimensions elaborated in the previous section. The resulting concept matrix shows which concepts have and which have not been researched thoroughly. Thus, we consider the concept matrix as our synthesis of studies. In the following section, we describe every dimension and their related concepts by writing the review.

4 RESULTS

This section presents the key findings of the systematic literature review. We present the concept matrix in Table 1 by describing current research streams for each of the dimensions. Note that, during the examination of the articles we encountered two additional algorithm types which is why we added the column others to this dimension. Attentive readers might wonder how some articles, e.g., [6, 9, 11, 15, 24], check conformance without assessing a particular quality metric. This is a consequence of approaches that convert process models from one language into another to perform conformance checking. Likewise, some authors declare a conformance value that does not exactly fit the definition of one of the quality metrics. Furthermore, if there is no model marked in the modelling language dimension, the authors of the examined article claim their approach to be independent of the modelling language.

4.1 Modelling language

Concluding from the first dimension modelling language in the concept matrix presented in Table 1, Petri nets embody by far the most commonly used process modelling language in conformance checking literature with 28 occurrences. A Petri net is a visually presentable graph expressing execution logic such as restrictions, control-flow, and concurrency. Due to their mathematical background, their capability to capture concurrent behavior, and their function as state charts, Petri nets have been thoroughly developed as a formal process modelling language. Burattin et al. [12], for instance, use Petri nets as a representative for all procedural modelling languages in their conformance checking framework. Even though most of the articles that use Petri nets mainly evaluate the control-flow of a process, there are a few approaches concerning multiple perspectives. Whereas, Mannhardt et al. [31], and de Leoni and van der Aalst [17] utilize so-called data Petri nets to enhance Petri nets by means to evaluate data-constraints, Alizadeh et al. [6] employ a so-called “Create Read Update Delete” (CRUD) matrix. The CRUD matrix relates data objects to process logic (i.e. a Petri net places or transitions), so that an activity occurs only if the related data allows or enforces it. The data Petri nets attach data-constraints wherever necessary to places and transitions of a Petri net. Subsequently, the algorithms skim the event data related to the current state of the process execution [17, 31]. Thus, enhanced Petri nets accomplish complete conformance checking that evaluates the control-flow and the data-perspective.

In the subset of the literature, three articles use BPMN as the technical specification for their input model. Although BPMN is quiet popular in the corporate environment, only one of the three approaches conducts conformance checking with native BPMN [18]. The authors state that their approach is independent from the actual modelling language since it takes advantage of extended causal nets. As a result, the authors argue that process logic is generally expressible in extended causal nets. While [18] use native BPMN process models to describe their conformance checking approach, [24] and [21] utilize a different technique to perform conformance checking using BPMN process models. They develop an artifact to convert BPMN process models into Petri nets so that they can perform conformance checking with the resulting Petri net. Last, a few articles do not focus on process modelling languages
but instead abstractly describe conformance concepts using, for instance, so-called work-flow descriptions [9, 33, 35]. The remaining articles use Declare process models as an input. Two of the identified articles conduct multi-perspective conformance analysis by implementing business rules and data constraints [8, 11]. Declare is presently the only graphical declarative language that has been employed in conformance checking research.

4.2 Perspective

The perspective dimension shows that six articles develop multi-perspective and 31 articles control-flow perspective approaches. As further data and perspectives, the multi-perspective articles mostly used time, roles, and contextual data such as costs [6, 8, 11, 18, 31]. While performing conformance checking of process models based on constraints, these process-flow constraints can be enhanced by business data constraints to project the data-flow [8]. Alizadeh et al. [6] provide means to conduct extended process auditing, whereby they combine the process and data perspectives to perform conformance checking. De Leoni and van der Aalst [17] focus on the additional perspectives of data, time, and resource, whereas one of their previous articles takes only data and resources into consideration [18]. One article addresses a combination of the process, time, and data perspectives in combination with declarative business rule checking [11]. Another notable aspect in this particular dimension is that most of the approaches that use declarative approaches can include more perspectives than the control-flow. Although a few authors state that conformance checking techniques must be able to detect deviating behavior concerning the misuse of organizational resources [18], the number of existing articles itself indicates that the control-flow of processes is of higher relevance for the process mining domains.

4.3 Algorithm type

In this section, we elaborate our findings regarding the algorithm types that the approaches apply to perform conformance checking. Notably, we encountered two additional algorithm types during the examination of the articles. Therefore, we expand the two initial concepts of trace alignment and log replay by constraint-based conformance checking and artificial negative event precision checking in the concept matrix column "others".

Vanden Broucke et al. [49] design the artificial negative event algorithm aiming to improve models by checking their precision and generalization in a novel way. Here, they define negative events as events that should not occur at a specific time or state of a process execution. When the trace \( t(a, b, c) \) is a successfully executing process trace in the event log \( L \), then the \( t' \), with induced artificial negative events for \( t \), is \( t' \{(b^-, c^-), a(a^-, c^-), b(a^-, b^-), c\} \). Here, the superscript '-' indicate negative events and '(' artificially generated events. To determine the conformance metric precision, every traces in the event log \( L \) replayed including the identified negative events from trace \( t' \). If a negative event never occurs, the process path including the negative event can be removed from the model. Otherwise, if it occurs as often as previously defined, a new process path should be added to the model.

Borrego and Barba [8] and Burattin et al. [12] perform constraint-based conformance checking. Thus, they translate process logic into rules and constraints which can thence check conformance by evaluating the fulfillment of each rule in every trace. Furthermore, the rules may contain data related constraints which leverage multi-perspective conformance checking [8]. The other constraint-based approach employs a behavioral model approach for conformance checking [12]. For this technique the authors assume that a process analyst can derive more advanced behavioral patterns from an existing process model. These behavioral patterns may express for example that activity \( a \) must occur before \( \beta \). After having derived all behavioral patterns from the model, these are evaluated with all process traces in an event log.

All in all, seven articles in the present set of literature use the log replay as an algorithmic base to determine the conformance of a log or a model. In order to perform conformance checking with a log-replay algorithm, an algorithm designer needs to implement a model interpreter beforehand. The model interpreter extracts all the rules and allowed states as well as transitions from a process model. After that, the algorithm re-runs and re-evaluates every trace from the event log, whereby the interpreted model specifications imply whether the replay fails or succeeds for each trace. There are several ways to measure conformance using log-replay algorithms. For instance, the token-based log replay (see section

| Article   | Modelling language | Perspective | Algorithm type | Quality metric |
|-----------|--------------------|-------------|----------------|---------------|
| [48]      | Declare            | Other       | Other          |               |
| [19]      | Declare            | Other       | Other          |               |
| [47]      | Declare            | Other       | Other          |               |
| [46]      | Declare            | Other       | Other          |               |
| [45]      | Declare            | Other       | Other          |               |
| [18]      | Declare            | Other       | Other          |               |
| [17]      | Declare            | Other       | Other          |               |
| [11]      | Declare            | Other       | Other          |               |
| [10]      | Declare            | Other       | Other          |               |
| [9]       | Declare            | Other       | Other          |               |
| [8]       | Declare            | Other       | Other          |               |
| [7]       | Declare            | Other       | Other          |               |
| [6]       | Declare            | Other       | Other          |               |

Table 1: Concept matrix using dimensions following [51]
execute the following steps to align multiple traces: pre-

tween an event log and a model. Another approach implements an 

Mannhardt et al. [31] evolve basic alignments into balanced multi-

CPU-efficient and faster solution to compute trace alignments. Re-

elaborates an algorithm based on estimated heuristics to create 

these two concepts focus on efficient usage of memory, van Dongen 

ing also derives from the automated planning domain [5]. Whereas 

problem. This approach bears the advantage of memory-efficiency. 

ally customizable costs for specific process violations [17], legal 

imization of a cost function [2, 3, 16]. These optimization functions 

required [4]. Therefore, cost functions are defined and evaluated for 

each alignment to get the optimal trace alignment based on the op-

mer. Bose and van der Aalst [9] and Prabhakara and van der 


take the arcs of a Petri net and to inserted or 

skipped activities. Afterwards, they calculate conformance ratios 

based on the outcome costs of the algorithm.

The major part of the identified literature employs trace align-

ment algorithms to compute the conformance of processes. Thus, 

trace alignments can be considered as the current standard of con-

formance checking techniques [28]. Trace alignment algorithms 

mostly utilize procedural process modelling languages. Two arti-

cles by the same main author promote a quite similar alignment 

framework. Bose and van der Aalst [9] and Prabhakara and van der 

Aalst [35] execute the following steps to align multiple traces: pre-

processing of event logs, computing scoring matrices, building a 

guide tree for multiple trace alignment, estimating the alignment 

quality, pruning and realignment, and an interactive visualization. 

Some of the research projects expanded the basic approaches with 

additional features to compute optimal alignments. Especially, since 

the A* algorithm returns multiple complete alignments for each 

log and model trace, a means to retrieve the optimal alignment is 

required [4]. Therefore, cost functions are defined and evaluated for 

each alignment to get the optimal trace alignment based on the op-

imization of a cost function [2, 3, 16]. These optimization functions 

rely on several types of input. These consist of distance [2, 3], man-

ually customizable costs for specific process violations [17], legal 

moves [16, 18], probabilities [26], synchronous moves [7], region 

theory and state similarity [10]. Furthermore, these approaches im-

plement cost-functions to create alignments which are customized 

to event monitoring points in a hospital setting to compute optimal 

alignments [4, 25]. Besides a cost function to compute an optimal 

alignment, another approach inserts the planning domain definition 

language to formulate computing alignment as a so-called planning 

problem [16]. Subsequently, off-the-shelf automated planners 

resolve optimal alignments based on the determined planning 

problem. This approach bears the advantage of memory-efficiency. 

Another method promoting memory-efficient alignment computing 

also derives from the automated planning domain [5]. Whereas 

these two concepts focus on efficient usage of memory, van Dongen 

[48] elaborates an algorithm based on estimated heuristics to create 

a CPU-efficient and faster solution to compute trace alignments. Re-

searchers have advanced the control-flow alignment approach to be 

able to handle multi-perspective process analysis [6, 17, 18, 31]. 

Mannhardt et al. [31] evolve basic alignments into balanced multi-

perspective alignments that align resource and context-data be-

tween an event log and a model. Another approach implements an 

integer linear programming approach by adding the dimensions 

time, data and resources to alignment cost functions [17]. Fur-

thermore, an approach to create multi-perspective alignments with 

BPMN process models uses causal nets enhanced with data [18]. 

One technique enhances trace alignments to inter-level alignments 

that relates process steps to log steps and to respective data entries 

in a CRUD matrix to enable multi-perspective conformance check-

ing [6]. Moreover, a method converts a BPMN process model and 

its respective event log into primary event structures [21]. Conse-

quently, both structures can be merged to generate an alignment 

between the model and the log. Fahland and van der Aalst [20] 

focus on model repair, whereby they use an alignment approach 

to determine missing or superfluous activities and connections. A 

variety of trace alignment approaches address the issue of decom-

posing complex conformance checking scenarios [33, 41, 44, 50]. 

One Petri net decomposition splits the overall process model into 

sub-nets [44]. As a result, these sub-nets can be aligned with certain 

parts of traces in an event log. The single entry single exit concept 

adds to the conformance checking approaches dealing with large 

event logs [33]. If a Petri net contains sequences that start at a 

single place (entry), then run through different process paths, and 

reach the same end node in any case, this sequence can be extracted 

and evaluated individually as it is considered as a valid sub-net. 

Additionally, a generic divide-and-conquer framework supports 

the decomposition of process models [50]. Song et al. [41] propose 

alignments based on heuristics rather than the A* algorithm and 

the divide-and-conquer strategy. From this point on they utilize 

the alignments to design a model repair method for industrial scale 

processes. Even though the lion’s share of trace alignment research 

is dedicated to procedural modelling languages, there are artifacts 

to create alignments using the declarative process modelling lan-

guages [14, 15]. All the above-mentioned approaches rely on clean 

and correctly ordered event logs. To address the issue of only par-

tially ordered event logs, Lu et al. [30] designed a trace alignment 

approach that calculates partial alignments by setting up depend-

encies for each place instead of strictly focusing on a place. The 

last concept of creating trace alignments converts BPMN process 

models into Petri nets, so that the optimal trace alignments can be 

computed for the resulting Petri net instead of the input BPMN 

model [24].

4.4 Quality metric

In the present systematic literature review, four Quality metrics 
to measure conformance for event logs and models are identified. 

These metrics are fitness, simplicity, precision, and generalization.

Given that the main objective of conformance checking is to iden-
tify whether the execution of the process (i.e. the data) is conforming 

with the specified process (i.e. the model), it is not surprising that 

fitness is the quality measure most frequently used throughout the 

examined articles. There are several ways to measure fitness. First, 

by using the log replay algorithm, the number of re-playable log 

traces can be ascertained. The division of the re-playable traces by 

the number of all traces can indicate a fitness value [20, 29, 32, 46]. 

Second, there is the token-based log-replay method to calculate the 

fitness (see conformance checking overview section). Adriansyah 
et al. [4] present such an event-based fitness measure that counts
each non-conforming event and divides them by the total number of events. Third, trace alignments utilize different kinds of cost functions to compute a fitness metric [5, 7, 14, 15, 17, 20, 25, 26, 33, 35, 47]. Two declarative ones approach the fitness calculation regarding previously determined business rules [8, 12]. The approach records every trace whereby the beforehand defined business rules indicate whether a trace is sound or not at the time of the execution [8]. By dividing the overall number of recorded traces by the number of violating traces, a process analyst can determine a conformance ratio already during the run-time of the respective process. Additionally, these rules may refer to event data, such as costs, resources etc., whereby the approach calculates multi-perspective fitness values for processes [12]. Because trace alignments compute fitness values on the event-level instead of the trace level, they are considered to be an accurate tool to determine fine-grained conformance information [31]. Although this method of fitness calculations is most capable, some drawbacks come with it. Finding an optimal alignment between large event logs and a model is considered as an NP-hard problem since an increasing number of iterative routes in a process model increases the number of potential alignment paths to a considerable extent. Lee et al. [28] tackle this problem by proposing a decomposed fitness measure that shows significant improvements in performance while getting very close to the overall fitness measure based on the optimal alignment.

Another metric to assess model quality relates to the principle of Occam’s razor and is called simplicity. When concerning simplicity, a process analyst assumes that the simplest model that expresses the behavior of a process model is always the best model. However, simplicity is hard to quantify and often refers to the number of nodes, and transitions in a process model as quantifiers [21]. A distinct approach defines simplicity as the number of sub-models added in the log compared to the existing process model [20].

The two remaining metrics, precision and generalization, are contradicting themselves. The model should be as general as possible, meaning the model should not overfit the log – and as precise as possible, meaning the model should not underfit the log. The majority of the identified articles utilize an alignment-based precision metric that is the result of dividing the available actions in the model by the executed actions [2, 3, 5, 44]. Similarly, de Leoni et al. [15] calculate precision for their declarative approach by dividing all executed activity sequences by the number of possible activity sequences. Furthermore, they use a probabilistic approach based on the alignment automaton of the Declare process model to estimate the probability of new events occurring that are not reflected in the event log [15]. One technique leverages the calculation of precision and generalization by using model-to-model comparison [29]. The authors compare the model generated from the data with the provided process model and divide them into smaller sub-models in order to reduce the complexity. An especially different method to measure precision and generalization implements the artificial negative event approach [49]. Here, the log is replayed with artificial negative events. By utilizing the number of replayed positive (allowed occurrence) and weighted negative (erroneous occurrence) events, the precision and generalization metric is calculated. This approach leverages process analysts to determine the exact point of the model that allows or restricts too much behavior.

5 DISCUSSION OF THE FINDINGS AND FUTURE RESEARCH DIRECTIONS

The present literature review examines the state of the existing techniques and reveals aspects of conformance checking that have not been examined as thoroughly as others. For that, we analyzed 37 articles to outline differences regarding Modelling language, Perspective, Algorithm type and Quality metric. In the following subsections, we discuss the findings of our review, state its limitations and conclude with an outlook for future activities building on our work.

5.1 Findings

Our literature review reveals that most articles examine the quality metric fitness regarding the quality metrics computed by conformance checking techniques. After all, this is not a big surprise given the fact that especially practitioners want to assess the quality of process execution rather than to assess the quality of the defined process model. In addition to the various concepts to calculate fitness, many researchers improve alignment based approaches in terms of their customizability regarding their cost-functions. Besides fitness, a few approaches deal with precision and generalization of process models. Simplicity, however, is rarely discussed in detail and there is no general quantitative approach to calculate it. After all, simplicity seems to result from the trade-off between precision and generalization which is hard to quantify. Regarding the algorithm types, techniques from the early days of conformance checking employ log replay algorithms. Today, almost every article builds on the concept of trace alignments. While trace alignments offer powerful capabilities, they come with the disadvantage of the complexity related to the pre-computing of model traces. Some articles propose solutions for more efficient ways to calculate aligned traces. Nevertheless, this area requires more research.

Regarding the process modelling languages, our research reveals a strong focus on procedural models specifically on Petri nets. Although research about procedural models is widely spread, regarding perspectives, algorithm types, and quality metrics, we find it noticeable that the variety of existing procedural modelling techniques, like event process chains, subject-oriented business process management or even BPMN, is not well covered in the state-of-the-art literature. Moreover, only four articles use the only representative of declarative process modelling languages, Declare. This fact provides evidence for a gap between the research efforts towards procedural and declarative process modelling. However, more and more research in the field of business process management is dedicated to declarative languages because they promise potential when dealing with complex and unstructured processes [8, 14]. While Declare is undoubtedly a prominent representative of the declarative languages, it has some drawbacks like slow adaptation by users due to its complexity [23]. Therefore, we argue to consider further declarative languages for the development of new conformance checking techniques.

Last but not least, more than 80% of all reviewed articles approach conformance checking by exclusively examining the control-flow of processes omitting further perspectives. It is astonishing to a certain extent that only six articles develop conformance checking techniques taking a multi-perspective view on processes and
We call for further research in the area of conformance checking. Therefore, we call for multi-perspective conformance checking techniques that allow a holistic conformance analysis of the process. A challenge for such techniques will certainly be posed by the increasing complexity and hence, the need for efficient techniques in order to maintain performance [13, p.262].

5.2 Contributions
We contribute to academia by providing an overview of the state-of-the-art to researchers who work in the domain of conformance checking. Such an overview helps to steer future research by outlining research gaps and subsequently displaying research needs for new techniques. Furthermore, the concept matrix in the results section shows exactly, which topics and concepts are not as well researched as others. Thus, our review may also function as a starting point to address new concepts in the research area.

Besides our contribution to the research domain, we provide a well-structured overview of recent relevant conformance checking approaches including a categorization of each of the identified techniques. This is particularly helpful for software vendors aiming at including conformance checking into their products. It might give business process management in corporate environments an overview of all the dimensions they may employ as well. Thus, we believe that our artifact supports managers’ decisions regarding the distinct dimensions when they want to implement conformance checking for their processes.

5.3 Limitations
Despite our best efforts to undertake a comprehensive search and review of the articles, we point out a few limitations to our review. Some of the articles were excluded due to the language barrier as they were not available in English which may have resulted in omitting important papers. Another minor limitation of our review might be the usage of Scopus for the evaluation of bibliometrics since the articles may have different numbers of citations on different search engines. However, we chose Scopus since it covers several research fields and numerous peer-reviewed journals and conference papers and does not customize search results based on preferences. Although the application of quantitative criteria in the literature selection might lead to missing out a “late-blooming” article, we believe that we captured the general picture of conformance checking approaches that use a process model and an event log as input. One of the articles meeting the quantitative criteria was not available as a full-text version to us. Despite these limitations, the review provided important insights into existing conformance checking techniques, their differences and blind-spots.

5.4 Outlook
We call for further research in the area of conformance checking based on the findings of our systematic literature review. In the following, we want to present how future researchers may extend our work to provide further insight into the field of conformance checking. First, we see potential in a more detailed review of articles working with trace alignments to identify differences in terms of their implementations. As trace alignments seemingly present the most suitable algorithm type for conformance checking, this would help to steer future research in the right direction. Especially, differences regarding their applied optimization-functions to compute optimal alignments might be important to practice and future research. Second, an evaluation of existing conformance checking techniques with real-life event logs should be conducted in order to assess the potential of conformance checking for organizations. Thus, the validation of some of the current conformance checking results lack empirical data, although we found several articles about conformance checking used in hospitals. Still, only few articles about the real-life application in distinct companies exist. Insights about the feasibility of conformance checking techniques in practice and potential benefits would help to identify requirements for new techniques. Last, we encountered two articles in our keyword search concerned with constraint- or rule-based conformance checking. Since we want to streamline strictly the “conformance checking” keyword which is solely related to the set of functions delivered in process mining. Thus, in future research, it might be of good value to provide an in-depth analysis of non-graphical process definitions, such as business rules etc., as well. Furthermore, related research fields such as fraud and anomaly detection, and compliance checking could be skimmed for articles that are of relevance to the conformance checking research domain.

5.5 Conclusion
With organizations looking for ways to efficiently and effectively manage their processes, and to avoid fraud and non-compliance, conformance checking is gaining momentum in practice. In the research domain, the number of publications is rising year by year. Thus, we wanted to identify trends and shortcomings of the conformance checking research of recent years to streamline the academic work. To address this objective, we conducted a systematic literature review of existing conformance checking techniques. We classified the 37 identified articles regarding the dimensions of Modelling language, Perspective, Algorithm type and Quality metric. The classification of the articles is displayed in a concept-matrix that relates distinct concepts of the particular dimension to the article. First, the identified Modelling languages consist of BPMN, Declare and Petri nets. Second, we classified the concepts belonging to the dimension Perspective by the control-flow perspective and multi-perspective approaches. Third, we encountered four different Algorithm types, whereby we summarized the approaches of rule-based conformance checking and artificial negative events as Others. The remaining two Algorithm types are the log-replay type and the most popular approach, the so-called trace alignments. Last, we found four distinct Quality metrics, fitness, simplicity, precision, and generalization. The systematic literature review presented in this research reveals that there is a lack of techniques based on declarative modelling languages. Furthermore, we observe that conformance checking techniques currently focus on the analysis of the process control-flow. To conclude the present paper, we want to encourage future researchers to address multi-perspective techniques that take account of further perspectives like time or roles so that organizations can analyze their complex business processes from various angles.
