Fast Incremental Conformance Analysis for Interactive Process Discovery

P.M. Dixit[1,2], J.C.A.M. Buijs[1], H.M.W. Verbeek[1], and W.M.P. van der Aalst[3]

[1] *Eindhoven University of Technology, Eindhoven, The Netherlands*
[2] *Philips Research, Eindhoven, Netherlands*
[3] *RWTH, Aachen, Germany*
{p.m.dixit,j.c.a.m.buijs,h.m.w.verbeek}@tue.nl
wvdaalst@pads.rwth-aachen.de

Abstract. Interactive process discovery allows users to specify domain knowledge while discovering process models with the help of event logs. Typically the coherence of an event log and a process model is calculated using conformance analysis. Many state-of-the-art conformance techniques emphasize on the correctness of the results, and hence can be slow, impractical and undesirable in interactive process discovery setting, especially when the process models are complex. In this paper, we present a framework (and its application) to calculate conformance fast enough to guide the user in interactive process discovery. The proposed framework exploits the underlying techniques used for interactive process discovery in order to incrementally update the conformance results. We trade the accuracy of conformance for performance. However, the user is also provided with some diagnostic information, which can be useful for decision making in an interactive process discovery setting. The results show that our approach can be considerably faster than the traditional approaches and hence better suited in an interactive setting.

Keywords: incremental conformance, interactive process discovery, domain knowledge, process mining

1 Introduction

Process mining is a technique that can be used to analyze process-oriented event data from information systems in order to perform business process intelligence tasks. Process mining includes two important tasks: process discovery and conformance checking. Process discovery techniques aim to discover a process model from an underlying event log. The primary aim of a process discovery technique is to come to a visual representation of a process model using the information from the event log. Even though most of the process discovery techniques are automated, there is also a possibility to involve human-in-the-loop in order to interactively construct a process model. Interactive process discovery techniques combine the traditional worlds of manual process modeling with data support, thereby allowing a user to add domain knowledge during process discovery.

Conformance checking techniques use a process model and an event log to calculate how well an event log fits the process model and vice versa. Traditionally, the conformance checking techniques are used to perform *post-mortem* analysis of the execution of a process. That is, once a process model is made available, conformance checking techniques determine the goodness-of-fit of the event data and the process model. However, in a user-guided process discovery setting, there is a need for active feedback regarding the goodness of a process model after each user interaction (see Figure 1). That is, after the user interactively changes a process model by taking into account the domain knowledge, there is a need to quantify how good or bad the change was according to the event data. This task can indeed be performed by conformance analysis, and thereby conformance analysis forms a natural fit with user-guided process discovery, and can be used to provide feedback after each step in an interactive process discovery setting.

In an interactive user-guided discovery setting, it is required to have *fast* feedback depending on the change made to the process model. In order to address this problem, we present a framework of calculating incremental conformance depending on the change made in the process model. Unlike other conformance analysis techniques, the framework discussed in this paper exploits the underlying structure used for interactive process discovery in order to perform fast and approximate conformance analysis. Moreover, we present an application of the framework that shows that even though the conformance results are approximated, they still contain diagnostic information which could provide useful feedback for the user for decision making in interactive process discovery.

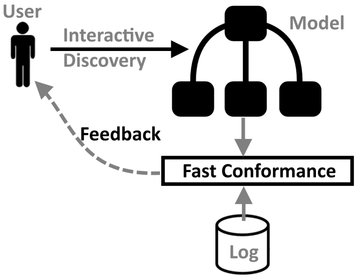


Fig. 1: Interactive process discovery setting. During discovery, conformance analysis is required to be computed fast enough for an uninterrupted experience for the user.

The rest of the paper is structured as follows. In Section 2 and Section 3, we discuss the related work from literature and the preliminaries resp. In Section 4 we discuss our approach and in Section 5 we discuss the implementation and visualization details. In Section 6 we evaluate our approach and in Section 7 we conclude and provide indicators of future work.

2 Related Work

Conformance techniques relate the behavior of processes in real life as depicted by event logs, with the expected behavior as depicted by procedural or declarative process models. A large number of conformance analysis techniques have been introduced within the field of process mining. The authors of [15] were among the first one's to devise conformance checking techniques in process mining using the token based replay in Petri nets. In [7], the authors discuss conformance strategies for declarative models. The authors of [3, 4] use alignment based strategy for performing conformance checking. Some approaches look at the conformance problem from various other dimensions such as natural language processing, formal methods, real time setting etc. [5, 6, 17]. Most of these approaches focus on the accuracy of the results, and hence do not emhapsize on the performance dimension and thereby are not advisable in an interactive process discovery setting. In [18], the authors discuss strategies for incrementally repairing a prior alignment based on a change in a process tree for improving the performance of the ETM algorithm. However, as the authors noted, after a certain number of repairs, the repaired alignment might be drastically different from the so-called optimal alignment. Thereby, this could lead to highly inaccurate results. Also, the class of models supported by this approach is limited to block structured process models (process trees).

In order to improve the performance of conformance analysis, various divide and conquer techniques were discussed in [1, 13, 16, 19, 20]. Typically, the central idea behind these techniques is to decompose a process model into various sub-models based on a specific decomposition strategy, and then to compute alignments on the smaller model (and a corresponding smaller log) and aggregate the information across all models. However, in certain scenarios, the decomposed sub-models may still have large state spaces. Therefore, the conformance calculation time in such scenarios may be similar, or even worse owing to decomposition and aggregation time, compared to the complete model. In all these divide-and-conquer techniques, there is more emphasis on the accuracy of the results. However, in our case, we relax the accuracy of the results to ensure short response times, similar to the technique discussed in [11]. In both [11] and our approach, conformance is calculated and aggregated over all combinations of sets of activities for a given cardinality. The main difference between our approach and [11] is that we inherently exploit the incremental nature of process modeling during interactive discovery. That is, in the case of [11], the conformance for all the combinations of activities after each change in a process model is recalculated. However, in our technique, we *remember* the prior conformance results, and typically recalculate only those conformance results which may have changed depending on the change in the process model. Hence, the proposed approach is much faster, robust and provides diagnostic information useful during interactive process discovery.

3 Preliminaries

In this section, we discuss the relevant preliminaries. *Events* form the basic building blocks of an event log. An event represents an occurrence of an activity. Every event has a timestamp associated with it which indicates the time of occurrence of that activity [2]. A trace is a collection of events, and an event log is a bag (or multiset) of traces.

Having discussed event logs, we now discuss process models. Process models are graphical constructs used to represent the flow of activities. Typically, a process model has a fixed start and a fixed end point, and a navigation from start to end point of a process model results in a trace. Hence a process model corresponds to a set of traces. Conformance analysis in process mining aims at finding a fit between the traces from an event log and a process model.

We now define two concepts of projecting a log and a process model on a set of activities. An event log can be projected on a set of activities by removing all the events from the event log which are not a part of the projection set. Consider an event log $L = [\langle a, b, c, d, a, e\rangle^{10}, \langle a, b, c, d\rangle^{5}, \langle b, d, e\rangle^{5}]$. Then the event log projected on activities $\{a, c\}$ is $L \downarrow^{\{a,c\}} = [\langle a, c, a\rangle^{10}, \langle a, c\rangle^{5}, \langle\rangle^{5}]$.

A process model M can be projected on a set of activities A by making all the activities from the process model which are not a part of projection set as *invisible* (depicting no behavior) activities. These *invisible* activities can be further removed (if possible) using language preserving reduction rules. The reduction removes the unnecessary *invisible* activities from the net which do not have any behavioral impact on the net. The projected and reduced model is denoted by $M \downarrow^{A}$. Due to space limitations, we refer the user to [8] and [14] for more details about these reduction rules.

4 Incremental Conformance Framework

In this paper we introduce a framework for enabling fast conformance analysis. The high level overview of our framework is presented in Figure 2. In order to enable fast conformance analysis, we split the problem into two parts, *(i)* approximating the conformance value by calculating and aggregating the conformance of projected sets of activities and, *(ii)* incrementally calculating the conformance.

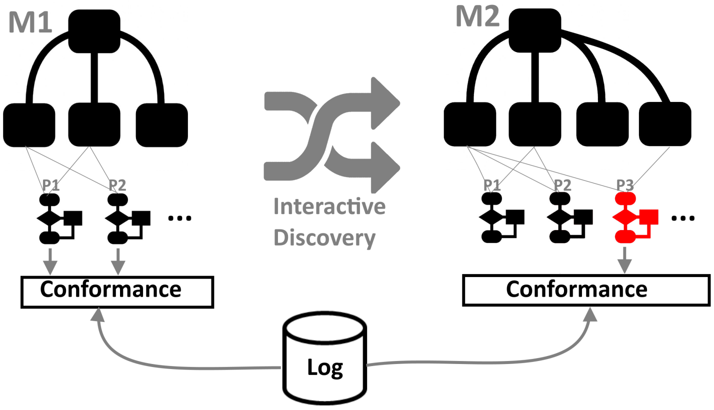


Fig. 2: Overview of conformance calculations when model M2 is derived from a model M1. Projections of models based on activity combinations within a given cardinality are calculated e.g. P1 and P2. For the activity combinations whose projected behavior does not change in M2 (compared to M1), e.g. P1 and P2, the projected conformance need not be recalculated. For all the other activity subsets, e.g. P3, the projected conformance needs to be recalculated or newly calculated.

4.1 Projected Conformance

The first part of our approach is similar to the the projected conformance checking [11] approach. Here we exploit the fact that calculating conformance for a smaller process model having a smaller state space and a smaller event log is often faster than calculating conformance for a large complex process model. Let A be all the activities present in a model M and let L be the corresponding event log. For a user defined cardinality k, using A we calculate the set of all possible activity combinations C, where $\forall_{c \in C} |c| = k$. Let $M \downarrow^c$ and $L \downarrow^c$ denote the process model M and event log L projected with the activities from activity combination c ($c \in C$). Let $\mathbf{Q}(M)$ define some function used to quantify the quality of the model M w.r.t. the event log. Then the quality of the overall model $\mathbf{Q}(M)$ is the aggregation of the quality values of all the projected models: $\mathbf{Q}(M) = \frac{\Sigma_{c \in C} \mathbf{Q}(M \downarrow^c)}{|C|}$.

Therefore, instead of calculating the conformance of one (complex) process model with an event log, conformance of several projected (smaller) process models with projected event logs are calculated and then aggregated. Here we exploit the fact that calculating conformance for one large process model is often time consuming, and thereby distributing the conformance calculation over several smaller projected process models improves efficiency.

4.2 Incremental Conformance

Sub-section 4.1 provides a way to calculate fast approximate conformance analysis, given a process model and an event log. However, in an interactive process discovery setting, a process model is expanded incrementally. In the second part of our framework, we make use of this principle in order to incrementally calculate the conformance using the projected conformance of the prior model. Instead of recalculating the projected conformance of all the activity combinations, we calculate the projected conformance of only those activity combinations which are necessary, and re-use the previously computed projected conformance for all the other activity combinations. Before introducing incremental conformance, we introduce the concept of behavioral equivalence in two models. Two models M_1 and M_2 are said to be behaviorally equivalent, represented as $M_1 \approx M_2$ iff all the behavior of M_1 is exhibited by M_2 and vice versa. Similarly, behavioral in-equivalence of two models is denoted by $M_1 \not\approx M_2$. Suppose a model M_{i+1} is interactively derived from a model M_i. Let C_{i+1} and C_i correspond to all the activity combinations of M_{i+1} and M_i, with a chosen cardinality k. Then, we can distinguish two cases:

1. Set of *same* activity combinations $C_S \subseteq C_{i+1}$ whose projected behavior is the same in models M_i and M_{i+1}, that is $\forall_{c \in C_S} M_i \downarrow^c \approx M_{i+1} \downarrow^c$.
2. Set of *different* activity combinations $C_D \subseteq C_{i+1}$ whose projected behavior is different in models M_i and M_{i+1}, that is $\forall_{c \in C_D} M_i \downarrow^c \not\approx M_{i+1} \downarrow^c$.

There is no need to calculate the conformance values for those activity combinations which exhibit the same projected behavior in M_i and M_{i+1} (C_S). However, the activity combinations C_D whose projected behavior is *not* the same, there is a need to calculate the conformance values. Therefore, in an interactive setting, we improve the projected conformance calculation times, by calculating conformance for only the activity combinations from the set C_D. It should be noted that, if a new activity is added to the model interactively, then there will be new activity combinations possible, which would all be a part of C_D. The amount of time needed for calculation of conformance for activity combinations C_S is saved by using an incremental way of calculating the conformance values.

4.3 Application of the framework

We show the instantiation of the framework using a synthesis rules based interactive process modeling approach. In order to do so, we first introduce free choice workflow (FC WF) nets which are a well-known class of Petri nets to represent processes. A workflow (WF) net is a bi partite graph containing places (circles) and transitions (rectangles), such that a transition can only be connected to a place and vice versa. A WF net is navigated from left-to-right, such that the first transition is always a 'dummy' (silent) transition called $\top$, and the last transition is always a 'dummy' (silent) transition called $\perp$. Transitions in a WF net can be labeled with an activity, or they can be *silent*, i.e. not representing any behavior from the event log. Moreover, a WF net is a FC WF net if for any two places, the set of output transitions of both the places are exactly the same, or completely disjoint (that is, there is no overlap at all). Figure 3a shows an example of a FC WF net. A short circuited version of a FC WF net can be obtained by merging the places i and o, and renaming it as io. It should be noted that we can easily obtain a FC WF net from a short circuited FC WF net by splitting the place io into places i and o.

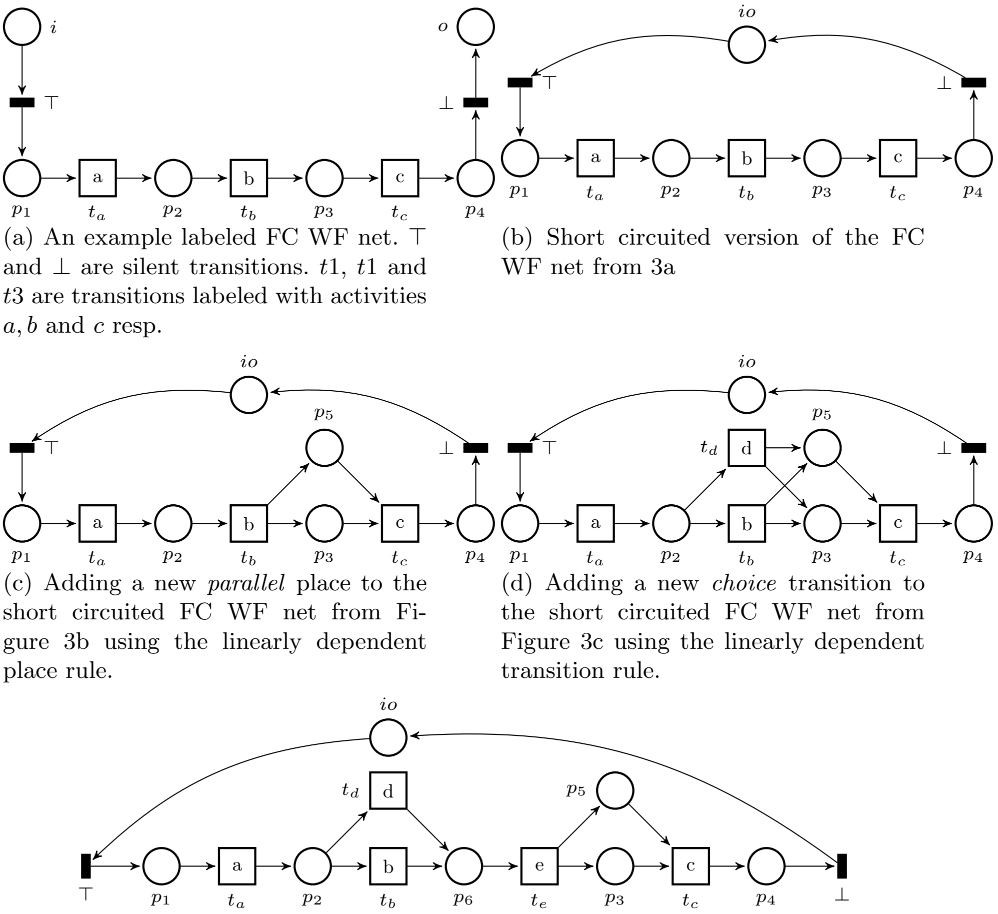
Three *rules* are used to interact and synthesize a short circuited FC WF net, such that each rule leads to addition of a new place and/or a new transition in the model. Using the synthesis rule based interactive modeling approach, at each iteration we identify the set of combinations of activities (C_S) whose projected behavior does not change. For all the other activity combinations (C_D), the conformance is recalculated. We describe the incremental way of calculating the change in the models depending on each type of rule.

4.4 Addition of a new place

Adding a new place to a net allows introduction of concurrency in the net. An introduction of a place does not result in any new activity in the model, and hence no new activity combinations are possible. There exists a set of bags of places P_{set} in the short circuited FC WF net, which has the same effect as the newly added place. In Figure 3c, this set corresponding to the new place p_5 is $\{[p_3]\}$. Typically, all the activity combinations are added to C_S, as the projected behavior of activity combinations remains unchanged. For example, in Figure 3c, the projected behavior between activities of the net does not change at all after the addition of the new place. However, in very few cases the projected behavior of activity combinations might have changed, if at least one of the bags in P_{set} contains the place io of the short circuited FC WF net. For example, in Figure 4b, P_{set} corresponding to the newly added place p_7 is $\{[p_6, p_4, io, p_1, p_2]\}$. Since P_{set} contains io, no activity combinations are added to C_S. This is also because the projected behavior between some activities has indeed changed as shown in Figure 4 (e.g. t_d and t_b changed from parallel to a sequential construct).

4.5 Addition of a new transition

Figure 3d shows the addition of a new transition using the so-called linearly dependent transition rule [8]. Addition of a new transition usually results in the introduction of a choice or loop in the model. There exists a set of bags of transitions T_{set}, which have the same effect on the short circuited FC WF net, as the newly added transition. In Figure 3d, this set corresponding to the



(a) An example labeled FC WF net. ⊤ and ⊥ are silent transitions. $t1$, $t1$ and $t3$ are transitions labeled with activities a, b and c resp.

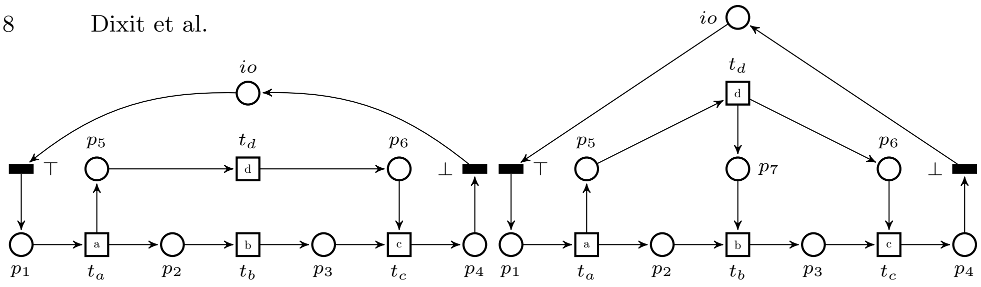
(b) Short circuited version of the FC WF net from 3a

(c) Adding a new *parallel* place to the short circuited FC WF net from Figure 3b using the linearly dependent place rule.

(d) Adding a new *choice* transition to the short circuited FC WF net from Figure 3c using the linearly dependent transition rule.

(e) Adding a new transition and place in sequence to the short circuited FC WF net from Figure 3d using the ψ'_A rule in between transitions t_b, t_d and places p_3, p_5.

Fig. 3: Synthesis rules [9] applied to short circuited FC WF nets.



(a) A short circuited FC WF net, where t_d is in parallel with t_b.

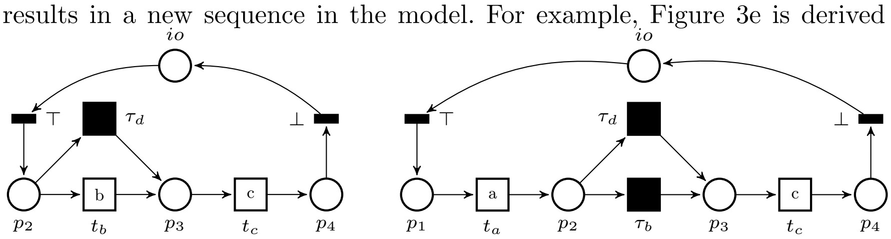
(b) Adding a new place to the net from Figure 4a using linearly dependent place rule. The result is that d and b are now in sequence (that is, d is followed by b).

Fig. 4: Addition of a new place resulting in introduction of sequential construct.

newly added transition t_d is $\{[t_b]\}$. We use this information to calculate the set of activity combinations C_S whose projected behavior does not change. No elements are added to C_S, if any bag from T_{set} contains $\top$ or $\bot$. This is for reasons similar the one described in Sub-section 4.4. The second scenario is when none of the bags from T_{set} contain $\top$ or $\bot$, i.e. $\forall_{E \in T_{set}} \top, \bot \notin E$. Let T_L be the set of all the labels represented by the transitions in T_{set}. An activity combination for a subset of activities A_s (s.t. the label of newly added transition is not in A_s) is added to C_S if $A_s \cap T_L = \emptyset$. Consider the model from Figure 3d derived from Figure 3c by adding a new transition labeled d ($T_L = \{b\}$). If the cardinality is chosen to be 2, then combination of activities $\{a, b\}$ and $\{b, c\}$ are not added to C_S. For example, consider the projection of activities $\{b, c\}$ as shown in Figure 5a. In the new net projected (and reduced) on activities $\{b, c\}$, there is a possibility to *skip* the activity b, via τ_d. Hence there is additional behavior introduced corresponding to activity b which was not present in the prior projected net. Hence such activity combinations are not added to C_S, and are candidates for recalculation. As a counter example, it is easy to see that the previous activity pair of $\{a, c\}$ has the same projected behavior, as shown in Figure 5b, after the introduction of t_d, and hence this activity combination is added to C_S.

4.6 Addition of a new transition and a new place

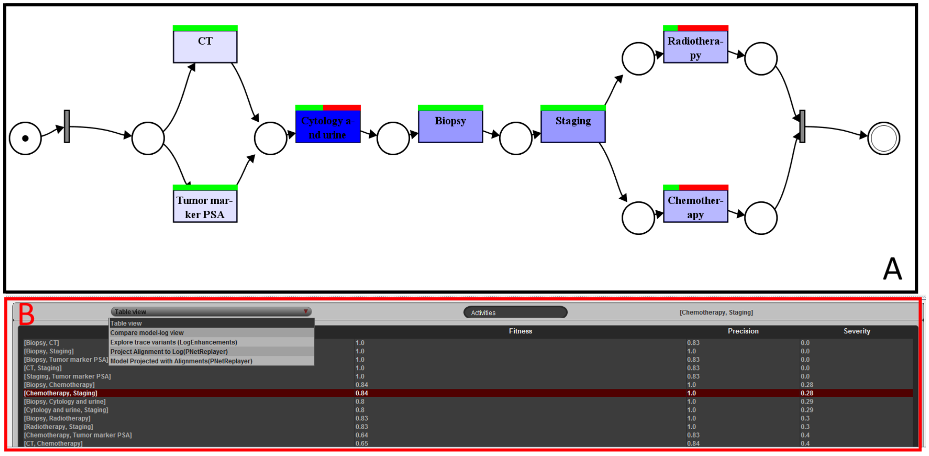
Adding a new transition and a new place using the so-called abstraction rule [8] results in a new sequence in the model. For example, Figure 3e is derived



(a) Projecting (and reducing) the net from Figure 3d onto $\{b, c\}$.

(b) Projecting the net from Figure 3d onto $\{a, c\}$.

Fig. 5: Projected behavior for $k = 2$ corresponing to the new net after adding t_d (Figure 3d).



	Fitness	Precision	Severity
[Biopsy, CT]	1.0	0.83	0.0
[Biopsy, Staging]	1.0	1.0	0.0
[Biopsy, Tumor marker PSA]	1.0	0.83	0.0
[CT, Staging]	1.0	0.83	0.0
[Staging, Tumor marker PSA]	1.0	0.83	0.0
[Biopsy, Chemotherapy]	0.84	1.0	0.28
[Chemotherapy, Staging]	0.84	1.0	0.28
[Biopsy, Cytology and urine]	0.8	1.0	0.29
[Cytology and urine, Staging]	0.8	1.0	0.29
[Biopsy, Radiotherapy]	0.83	1.0	0.3
[Radiotherapy, Staging]	0.83	1.0	0.3
[Chemotherapy, Tumor marker PSA]	0.64	0.83	0.4
[CT, Chemotherapy]	0.65	0.84	0.4

Fig. 6: (A) shows the conformance of the modeled activities according to the event log: the density of blue color indicates the frequency, the red (darker) and green (lighter) bars above the transitions indicate the distribution of fitting and non-fitting events. The tabular view (B) shows the individual scores of the activity sets (k=2 in this case). The user can dig deeper to explore the relationship between any activity set by choosing the desired visualization.

by adding a new transition (labeled e) and a new place (p_6) to the model from Figure 3d. If the newly added transition is labeled with an activity which is not already present in the model, then for any chosen cardinality, all the activity combinations from Figure 3d are behaviorally equivalent in Figure 3e. That is, if the transition for newly added activity e is made silent, then the net would be behaviorally equivalent to the previous net. Hence all the activity combinations from Figure 3d are added to C_S.

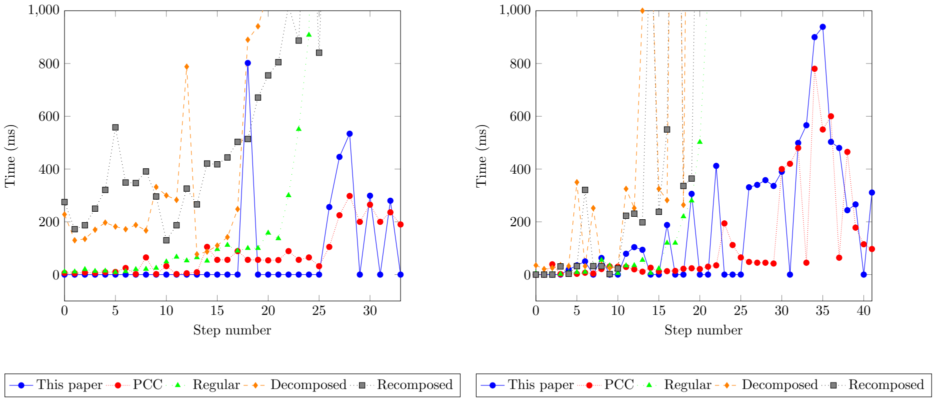
5 Implementation and Visualization

The technique has been implemented in the Interactive Process Mining package of the nightly build version of ProM tool[4]. Figure 6 shows and discusses the visualization of our technique. There are two views, one showing the aggregated visualization of conformance information across all the activity combinations directly on the process model. The other view shows a tabular view of all the activity combinations, along with the corresponding metrics. The user can interact with the activity combinations from the tabular view, and visualize the reduced models containing only those activities present in the selected activity combination. This allows the user to dig deeper to analyze the intricacies of fragmented process models with a certain cardinality, as shown in Figure 8.

6 Evaluation

We evaluate the approach presented in this paper by comparing it with state-of-the-art conformance techniques. The goal of this evaluation is to show the

[4] http://www.promtools.org/doku.php?id=nightly



(a) Performance times for the filtered BPIC 2011 event log.

(b) Performance times for the Sepsis event log.

Fig. 7: The time(s) taken after each interaction (step) for conformance analysis.

effectiveness of our approach in an interactive setting, measured in terms of performance times and correctness of the result. We use two real-life event logs: (i) the Sepsis event log[5] containing the workflow data for roughly 1000 patients suffering from Sepsis in a hospital. A normative process model is already available for this event log at [12]. (ii) the BPIC 2011 event log[6] containing hospital data for cancer patients from a Dutch hospital - filtered to retain only top 40% of the most frequent activities, resulting in a total of around 10000 events. We use the inductive miner [10] to discover a process model from the filtered event log. In order to replicate an interactive scenario, starting with a process model without any activities, we interactively re-construct each process model using the synthesis rules based approach. After each "interaction" (step), conformance is recalculated using following techniques: the decomposed replay technique [20], the recomposed replay technique [19], projected conformance checking [11] (with k=2), the technique presented in this paper (with k=2) and the regular alignments technique [4]. The fitness, precision and time taken for recalculation w.r.t. each technique are recorded.

6.1 Comparing performance times

Figure 7 compares the performance of each approach in terms of time taken after each step. It is quite evident that the traditional approaches, along with decomposed and recomposed approaches can be extremely slow, especially as the size of the process model increases. It was observed that the traditional approach for calculating fitness/precision could take more than 30 mins for both the event logs for the final step. It should however be observed that even though the decomposed approaches are slower, the quality values computed using these techniques were identical to the alignment based conformance technique [4] (which can be considered as a baseline for fitness value). Our approach is within 5% of the

[5] https://data.4tu.nl/repository/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460

[6] http://www.win.tue.nl/bpi/2011/challenge

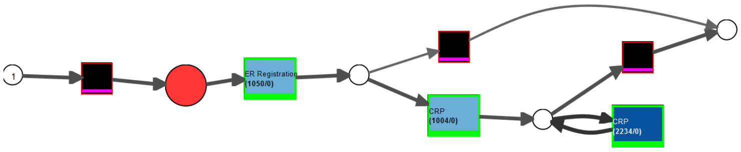


Fig. 8: The view of projected alignment on activities ‘CRP’ and ‘ER Registration’ from the Sepsis event log.

baseline and is much faster than the traditional approaches as we exploit the inherent rules used during process composition. [11] is typically faster than most of the approaches. However, [11] provides very limited diagnostic information about the transitions in the FC WF net, especially when there are duplicate activities present in the model. Our approach is more robust and can deal with duplicate occurrences of activities in the FC WF net. Moreover, by projecting the model on a subset of activities, we can find the relationship between activities which may not be directly connected in the overall model, for e.g. Figure 8. Also, during many steps in the process construction, the response time of our approach is almost zero, or close to zero. These are the changes in the process model wherein the projected behavior of all the activity combinations was the same in the iterated model.

7 Conclusion and Future work

In this paper, we presented a framework to enable fast conformance checking in an interactive process discovery setting. We instantiated this frameowrk to exploit the underlying principles used in interactive process discovery to calculate fast conformance by incrementally studying the change in the structure of a model. By using two real-life event logs we were able to show that the approach suggested in this paper is faster compared to many state-of-the-art conformance checking techniques. Furthermore, even though the actual fitness (and precision) scores are only approximated, they are still very close to the original values as computed using the traditional alignment-based conformance checking approach. In the future, we would like to extend the technique presented in this paper to combine it with other conformance checking techniques such as the one in [11].

References

[1] van der Aalst, W.M.P.: Decomposing petri nets for process mining: A generic approach. Distributed and Parallel Databases 31(4), 471–507 (Dec 2013)

[2] van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)

[3] Adriansyah, A., van Dongen, B.F., van der Aalst, W.M.P.: Conformance checking using cost-based fitness analysis. In: Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International. pp. 55–64. IEEE (2011)

[4] Adriansyah, A., van Dongen, B.F., van der Aalst, W.M.P.: Towards robust conformance checking. In: Business Process Management Workshops, Lecture Notes in Business Information Processing, vol. 66, pp. 122–133. Springer Berlin Heidelberg (2011)

[5] vanden Broucke, S.K.L.M., Munoz-Gama, J., Carmona, J., Baesens, B., Vanthienen, J.: Event-Based Real-Time Decomposed Conformance Analysis, pp. 345–363. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)

[6] Carmona, J.: The Alignment of Formal, Structured and Unstructured Process Descriptions, pp. 3–11. Springer International Publishing, Cham (2017)

[7] De Leoni, M., Maggi, F.M., van der Aalst, W.M.P.: Aligning event logs and declarative process models for conformance checking. In: Business Process Management, pp. 82–97. Springer (2012)

[8] Desel, J., Esparza, J.: Free choice Petri nets, vol. 40. Cambridge university press (2005)

[9] Esparza, J.: Synthesis rules for Petri nets, and how they lead to new results. In: International Conference on Concurrency Theory. pp. 182–198. Springer (1990)

[10] Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs containing infrequent behaviour. In: Business Process Management Workshops. pp. 66–78. Springer (2014)

[11] Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Scalable process discovery and conformance checking. Software & Systems Modeling (Jul 2016)

[12] Mannhardt, F., Blinde, D.: Analyzing the trajectories of patients with sepsis using process mining. RADAR+EMISA 1859, 72–80 (Jun 2017)

[13] Munoz-Gama, J., Carmona, J., van der Aalst, W.M.P.: Single-Entry Single-Exit decomposed conformance checking. Information Systems 46, 102–122 (2014)

[14] Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE 77(4), 541–580 (Apr 1989)

[15] Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Information Systems 33(1), 64 – 95 (2008)

[16] Taymouri, F., Carmona, J.: Model and event log reductions to boost the computation of alignments (2016)

[17] Taymouri, F., Carmona, J.: A Recursive Paradigm for Aligning Observed Behavior of Large Structured Process Models, pp. 197–214. Springer International Publishing, Cham (2016)

[18] Vázquez-Barreiros, B., van Zelst, S.J., Buijs, J.C.A.M., Lama, M., Mucientes, M.: Repairing Alignments: Striking the Right Nerve, pp. 266–281. Springer International Publishing, Cham (2016)

[19] Verbeek, H.M.W.: Decomposed Replay Using Hiding and Reduction as Abstraction, pp. 166–186. Springer Berlin Heidelberg, Berlin, Heidelberg (2017)

[20] Verbeek, H.M.W., van der Aalst, W.M.P., Munoz-Gama, J.: Divide and conquer: A tool framework for supporting decomposed discovery in process mining. The Computer Journal 60(11), 1649–1674 (2017)