Replay using recomposition

Lee, W.L.J.; Verbeek, H.M.W.; Munoz Gama, J.; van der Aalst, W.M.P.; Sepúlveda, M.

Please check the document version of this publication:

• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

Replay using Recomposition: Alignment-Based Conformance Checking in the Large

Wai Lam Jonathan Lee[2], H.M.W. Verbeek[1], Jorge Munoz-Gama[2],
Wil M.P. van der Aalst[1], and Marcos Sepúlveda[2]

[1] Architecture of Information Systems Group, Department of Mathematics and
Computer Science, Eindhoven University of Technology, (The Netherlands)
`{h.m.w.verbeek,w.m.p.v.d.aalst}`@tue.nl
[2] Department of Computer Science, School of Engineering
Pontificia Universidad Católica de Chile, (Chile)
`{walee,jmun}`@uc.cl - `{marcos}`@ing.puc.cl

Abstract. In the area of process mining, efficient alignment-based conformance checking is a hot topic. Existing approaches for conformance checking are typically *monolithic* and compute exact fitness values. One limitation with monolithic approaches is that it may take a significant amount of computation time in large processes. Alternatively, *decomposition* approaches run much faster but do not always compute an exact fitness value. This paper presents the tool *Replay using Recomposition* which returns the exact fitness value and the resulting alignments using the decomposition approach in an iterative manner. Other than computing the exact fitness value, users can configure the balance between result accuracy and computation time to get a fitness interval within set constraints, e.g., "Give me the best fitness estimation you can find within 5 minutes".

1   Recomposing Conformance for Large Processes

In conformance checking of process mining, alignment-based approaches have become the state-of-the-art technique due to their robustness and detailed view on deviations [1]. Unlike previous approaches, alignment-based conformance checking provides an optimal analysis and can pinpoint deviations between the observed behavior and the modeled behavior at the level of events, e.g., the identification of an executed activity in the process that is not permitted by the process model. However, with the growth in the volume of data and the size of event logs, there is a need to make alignment-based conformance checking more scalable. Case studies have shown that alignment-based conformance checking can take excessive time or even be unfeasible in large and complex processes. Furthermore, no result is provided if the technique is not completely feasible for the entire dataset.

One of the most promising research lines is to decompose the alignment problem. Rather than aligning the overall process and the overall event log, the two are decomposed into subprocesses and sublogs, and alignment is performed on these smaller subcomponents. Experimental results based on large scale applications of decomposition techniques have shown significant improvements in performance, especially in computation time [4, 6]. A recent work has presented a new technique, *Recomposing Conformance* [2], which not only applies decomposition to the alignment problem, but merges results from subcomponents as an overall result. Moreover, this merged result is guaranteed to correspond to the exact overall result that would have been yielded under the overall approach without decomposition. As such, there are three main contributions. First, the application of decomposition techniques can lead to significant reduction in computation time. Second, approximate interval results can be computed even if conformance checking is not completely feasible for the whole dataset. Third, users can configure different criteria such as the overall time threshold to adjust to the desired balance between accuracy and computation time. In this paper we introduce *Replay using Recomposition*, an implementation of the *Recomposing Conformance* technique.
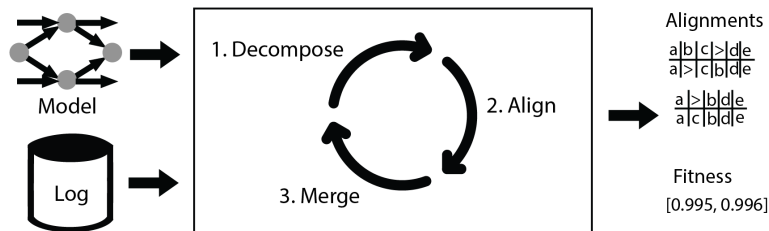
2 Recomposing Conformance: The Tool

Fig. 1. Overview of *RecomposingConformance* framework

Replay using Recomposition has been implemented as a plugin in the *ProM6.7* framework. Figure 1 provides a high level overview of the conformance checking framework. It takes in a process model (an *Accepting Petrinet*, i.e., a Petri net with initial and final markings) and an event log as input and returns a set of alignments and fitness scores as output. *Recomposing Conformance* is an iterative framework composed of three phases. First, a decomposition of the model and log is generated either by an initial decomposition or through a modification of the decomposition from the previous iteration. Second, subcomponents of the decomposition are aligned separately to achieve performance gains over the monolithic approach. Third, alignment results from subcomponents are merged if possible. Alignment for the subset of the log that could not merged in the current iteration is continued in the following iteration. We refer users to [2] for

further details on the formal guarantees, merging conditions, and other details of the technique.

As previously mentioned, the *Recomposing Conformance* framework is highly configurable. Users can configure three aspects to balance the result accuracy and the required computation time. First, users can configure the initial decomposition of the model and log which is required to start the conformance checking process. Second, users can opt between two decomposed replay methods: normal decomposition or Hide&Reduce [5]. Third, the desired level of accuracy and computation time can be adjusted through trace rejection and termination conditions. For trace rejection, users can set the computation time and maximum conflict thresholds for individual traces so that traces that take a long time to align due to conformance issues are rejected to reduce computation time. For termination, users can set the overall computation time and maximum iteration thresholds, the target alignment percentage of the log and the target range for the approximate interval result (level of accuracy) so that the checking process can be terminated once the set time or accuracy is met. As previously mentioned, an approximate interval result is computed if conformance checking is not feasible for the whole dataset under the set constraints. Figure 2 shows a screenshot of a dialog in the plugin where users can configure the mentioned parameters and the resulting alignment that can be used for deviation analysis.

Fig. 2. Configuration dialog in plugin (left) and resulting alignments (right)

3   A running example

To showcase the tool, we take a running example and show what the tool does with it. Given the fact that the demo needs to be interactive, we can only show a small example here. Of course, for such a small example, the non-decomposed replay will also finish well within time, but if we take a larger example for which this replay does not finish within time, the demo will not be interactive any more. For these larger examples that show that the recomposing replay can be much faster than the non-decomposing replay, we refer the interested reader to [2].

For the small running example, we take the "a12f0n05" event log from [3]. This log contains 12 different activities, and 35 different traces which contain
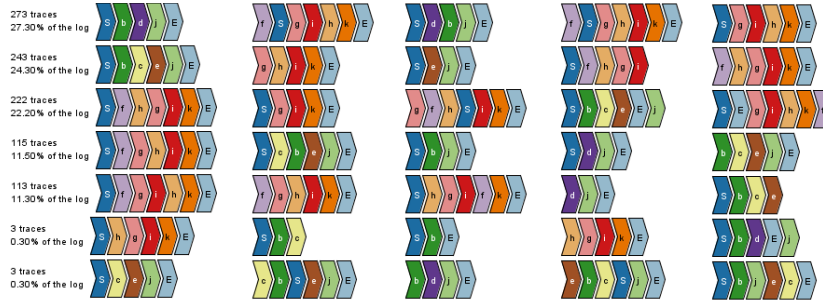
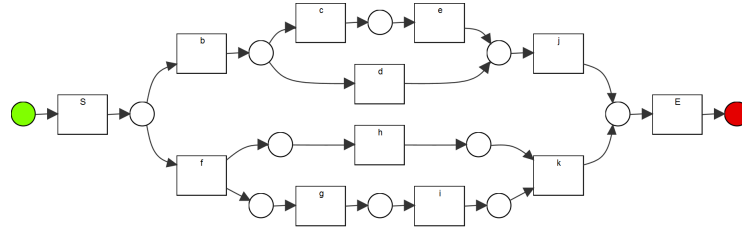**Fig. 3.** Event log for the running example

Fig. 4. Petri net for the running example

some (5%) noise. Figure 3 shows a graphical overview of this event log, showing 35 different traces of which only the 7 different traces in the left-most column occur more than once in the log. Figure 4 shows the corresponding Petri net used to replay this event log.

In the first iteration, the recomposing replay replays all 35 different traces on 12 clusters. For 14 out of these 35 different traces, the first iteration already provides us with a valid alignment. For the remaining 21 different traces, conflicts were found and another iteration is required. For the second iteration, the activity 'f' is selected to be joined on. As a result, the clusters containing 'f' will be merged, yielding only 10 clusters. Only 12 out of the 21 remaining different traces had a conflict on 'f', so only these 12 are replayed on these 10 clusters. From these 12, 5 provide us with a valid alignment, and 7 do not. . Table 1 shows the complete results for the running example. This shows that only a single different trace required a replay on the entire (non-decomposed) net, and that all other 34 different traces could be replayed successfully on some decomposition.

4 Download, Screencast, and Examples

The tool is available as the *Replay using Recomposition* plugin in the *Decompose-dReplayer* package of ProM6.7 (`www.promtools.org`).The plugin is available as a silent version (default configurations) and a visually configurable version. The source code is also available in `https://svn.win.tue.nl/trac/prom/browser/Packages/DecomposedReplayer`. A screencast showing the features of the tool and example datasets are also available at `www.processmininguc.com/tools`.

Table 1. Results on the running example.

Iteration	Join on	#Cluster	#Replay	#Accept	#Reject	#Conflict
1		12	35	14	0	21
2	f	10	12	19	0	16
3	c	9	6	20	0	15
4	e	8	4	21	0	14
5	S	7	3	23	0	12
6	j	6	5	26	0	9
7	b	5	3	29	0	6
8	E	4	2	31	0	4
9	d	4	2	33	0	2
10	g	3	1	34	0	1
11	k	1	1	35	0	0

5 Conclusions

This paper presented *Replay using Recomposition*, an implementation of the *Recomposing Conformance* technique. This alignment-based conformance checking technique applies decomposition to compute an overall result that corresponds to the exact result that would have been yielded under the monolithic approach. Furthermore, users can adjust configurations to balance between result accuracy and computation time. Unlike existing alignment-based conformance checking techniques, an approximate result is given if conformance checking is not feasible for the whole dataset under the set constraints.

References

1. Adriansyah, A.: Aligning Observed and Modeled Behavior. Ph.D. thesis, Technische Universiteit Eindhoven (2014)
2. Lee, W.L.J., Verbeek, H., Munoz-Gama, J., van der Aalst, W.M.P., Sepúlveda, M.: Recomposing Conformance: Closing the Circle on Decomposed Alignment-Based Conformance Checking in Process Mining. (under review) (2017), `processmininguc.com/publications`
3. Maruster, L., Weijters, A.J.M.M., Aalst, W.M.P.v.d., Bosch, A.v.d.: A rule-based approach for process discovery: Dealing with noise and imbalance in process logs. Data Min. Knowl. Disc. 13(1), 67–87 (2006)
4. Munoz-Gama, J.: Conformance Checking and Diagnosis in Process Mining - Comparing Observed and Modeled Processes. Springer (2016)
5. Verbeek, H.M.W.: Decomposed replay using hiding and reduction. In: PNSE 2016 Workshop Proceedings. pp. 233–252 (2016)
6. Verbeek, H.M.W., Munoz-Gama, J., Aalst, W.M.P.v.d.: Divide And Conquer: A Tool Framework for Supporting Decomposed Discovery in Process Mining. The Computer Journal (2017)