Decomposed Process Discovery and Conformance Checking

Josep Carmona

Definition

Decomposed Process Discovery and *Decomposed Conformance Checking* are the corresponding variants of the two monolithic fundamental problems in process mining van der Aalst (2011): *automated process discovery*, which considers the problem of discovering a process model from an event log Leemans (2009), and *conformance checking*, which addresses the problem of analyzing the adequacy of a process model with respect to observed behavior Munoz-Gama (2009), respectively.

The term *decomposed* in the two definitions is mainly describing the way the two problems are tackled operationally, to face their computational complexity by splitting the initial problem into smaller problems, that can be solved individually and often more efficiently.

Josep Carmona
Universitat Politècnica de Catalunya, e-mail: jcarmona@cs.upc.edu

Overview

The input for process discovery is an *event log* Mendling and Dumas (2009), from which a process model (typically a Petri net Murata (1989)) needs to be produced. The input for conformance checking is an event log and a process model (again, typically a Petri net), from which a conformance artefact or a conformance diagnosis will be produced.

General View of Decomposed Process Discovery. The general view of decomposed process discovery is shown in Figure 1. Initially, the log may be mapped to a different representation R, which is well suited to apply decomposition. Among the different alternatives, a common one is to do the decomposition on the log itself (so, $R = L$). An alternative, is to derive a state-based representation, e.g., using the techniques from van der Aalst et al (2010). Then, instead of directly deriving a process model from the representation of the event log (R), it is first partitioned, deriving a set of representations $R_1, \dots R_n$. In case the event log is used for decomposition, these will be *sublogs* $L_1, \dots L_n$ of L.

An important distinction should be done: the decomposition that is performed for decomposed process discovery and conformance checking is different from the decomposition that is done by *trace clustering* techniques. Trace clustering techniques partition the traces in the event log into multiple sets of traces, such that each trace in the original log can be found in one of the sublogs. In general, each sublog generated by trace clustering is a set of "similar" traces, and it corresponds to a "variant" of the process. For the purpose

Carmona, J. Decomposed process discovery and conformance checking. A: "Encyclopedia of big data technologies 2019". Berlín: Springer, 2018, p. 1-7.
The final authenticated version is available online at hhttps://doi.org/10.1007/978-3-319-63962-8_95-1

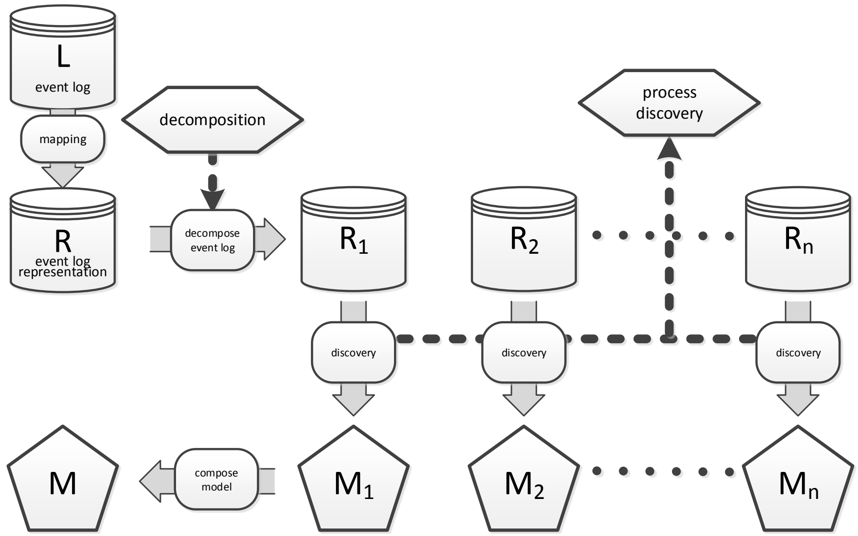

Fig. 1 General view of decomposed process discovery. Figure adapted from van der Aalst (2013).

of decomposed process discovery, the decomposition does not consider dividing the traces in the log into multiple sets, but instead by splitting each trace into a set of *subtraces*. In other words, one trace is cut into multiple ones, denoted subtraces. After each trace has been split in this way, each set of subtraces is put together into one sublog. Intuitively, the idea is that each sublog produced in this way corresponds to a stage or subprocess (or more generally a "fragment") of the original process.

The available techniques to do the partition are enumerated below. Then, from each representation R_i, a process model M_i is obtained through the application of an automated process discovery technique. The output of decomposed process discovery can be the set of process models or fragments derived, or if some composition mechanism is disposable, a unique final model M that encompasses the unified behavior of the n derived process models.

General View of Decomposed Conformance Checking. The general view of decomposed conformance checking is shown in Figure 2. Although both process model and event log are decomposed, the process model decomposition is applied first, and then the event log decomposition is performed so that process model fragments and corresponding sublogs match on their alphabet of activities. Then, conformance checking techniques can be applied locally on each pair of sublog and fragment generated. As conformance checking strongly relies on relating modeled and observed behavior Munoz-Gama (2009), the obtention of a conformance artefact like an *alignment* Adriansyah (2014) between a sublog L_i and a fragment M_i can be materialized. Likewise, conformance diagnosis can already be obtained on these local problems.

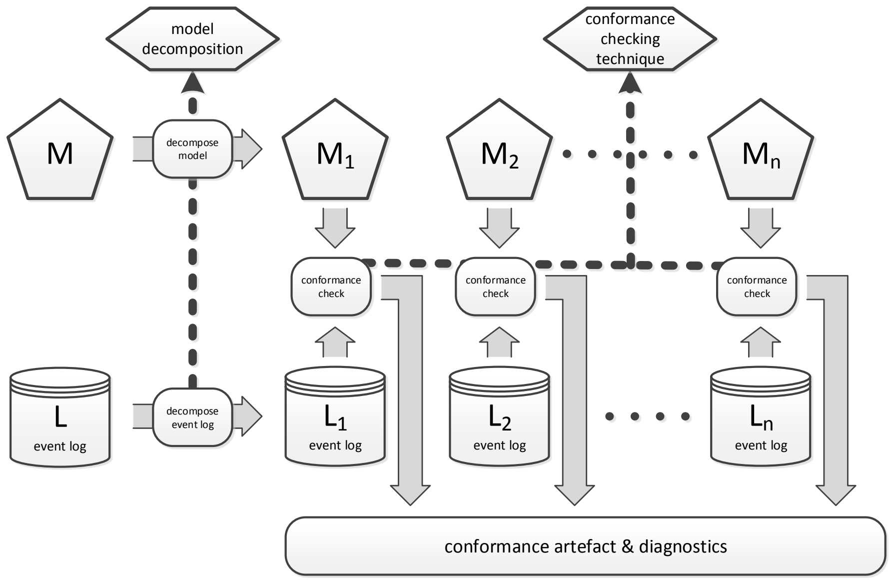

Fig. 2 General view of decomposed conformance checking. Figure adapted from van der Aalst (2013).

The decompositions mentioned above can then be applied at different levels: *log decomposition* and *process model decomposition* van der Aalst (2013). Below a general perspective on the two types of decomposition techniques is provided.

Log decomposition. When the event log is not transformed to a different representation, decomposition can then be applied on top of the event log. Event logs can be either split horizontally, or projected vertically, or both. Figure 3 provides an example of such operations. The horizontal selection of traces can be done in many ways, but *trace clustering* Greco et al (2006); Ferreira et al (2007); Bose and van der Aalst (2009b,c); Weerdt et al (2013); Hompes et al (2015) is usually applied. Notice that in the figure, sublogs arising from clusters can overlap, e.g., trace 3 belongs both to $L1$ and $L2$ in the horizontal selection of traces from L. Clearly, by decomposing a log into smaller sublogs, a linear factor on the complexity reduction can be accomplished.

The vertical projection of log traces starts by selecting a proper subset $\mathscr{A}'$ of the log's event alphabet $\mathscr{A}$. It then projects each trace σ in the log onto activities in $\mathscr{A}'$, i.e., removing from σ those events in $\mathscr{A} - \mathscr{A}'$, resulting in the trace $\sigma|_{\mathscr{A}'}$. The vertical projection can derive subsets $\mathscr{A}_1, \cdots, \mathscr{A}_n$ with $\mathscr{A}_i \subset \mathscr{A}$, deriving the sublogs $L|_{\mathscr{A}_1}, \ldots, L|_{\mathscr{A}_n}$. How to select those sets is the crucial decision; there are techniques based on the *directly-follows* relation from the α-algorithm Carmona (2012), or hierarchical methods that learn patterns which become new labels in the alphabet Bose and van der Aalst (2009a), among others. The vertical projection has in general a bigger impact in the complexity alleviation than the horizontal split, since the complexity of most of the techniques for discovery or conformance checking is dominated by the size of the alphabet.

In case the log has been transformed into a different representation *R*, then other forms of decomposition are applicable. For instance, if the log is transformed into a *transition system* Arnold (1994) which encompasses the behavior underlying in *L*, then decomposition techniques like the ones presented in de San Pedro and Cortadella (2016) can be applied.

Process model decomposition. Process model decomposition only makes sense for conformance checking, where apart from the event log, the process model is also an input. The intuitive idea is to break the process model into fragments, and project vertically the log accordingly (see Figure 2). This way, the problem instances tackled in decomposed conformance checking are significantly smaller than in the monolithic version.

Not every decomposition can be used in the scope of conformance checking: a *valid decomposition* van der Aalst (2013) of a Petri net into fragments requires that the places and arcs of the net are partitioned among the fragments, meaning that the decomposition uniquely assigns each place to a single fragment. Transitions in the frontiers of the cuts deriving the fragments are replicated, while transitions inside one fragment cannot, i.e., transitions inside a fragment cannot occur in other fragments. Figure 4 shows an example of a valid decomposition of a process model. Examples of decompositions are the ones based on *passages* van der Aalst (2012), or based on *Single-Entry Single-Exit* (SESE) Munoz-Gama et al (2014).

Key Research Findings

In general, the use of decomposition tends to make automated process discovery and conformance checking problems more tractable. However, current decomposition techniques cannot guarantee optimality in general, or even to provide the same outputs as the ones provided in the monolithic versions of the problems.

Key Research Findings in Decomposed Process Discovery. Techniques for decomposed process discovery have appeared in different forms in the last years Solé and Carmona (2012); Carmona (2012); van der Aalst (2013); van der Aalst and Verbeek (2014); van der Aalst et al (2015); de San Pedro and Cortadella (2016). These approaches not only focus on alleviating the computation of the process discovery task, but also consider other goals. For instance, a discovery approach approach based on van der Aalst (2013) was used in the framework that recently won the *Process Discovery Contest* Carmona et al (2016). Decomposition-based process discovery can also be applied to derive a different modeling notation, like it is presented in Solé and Carmona (2012) for *C- nets* van der Aalst et al (2011). It can also focus on deriving particular Petri-net subclasses, like

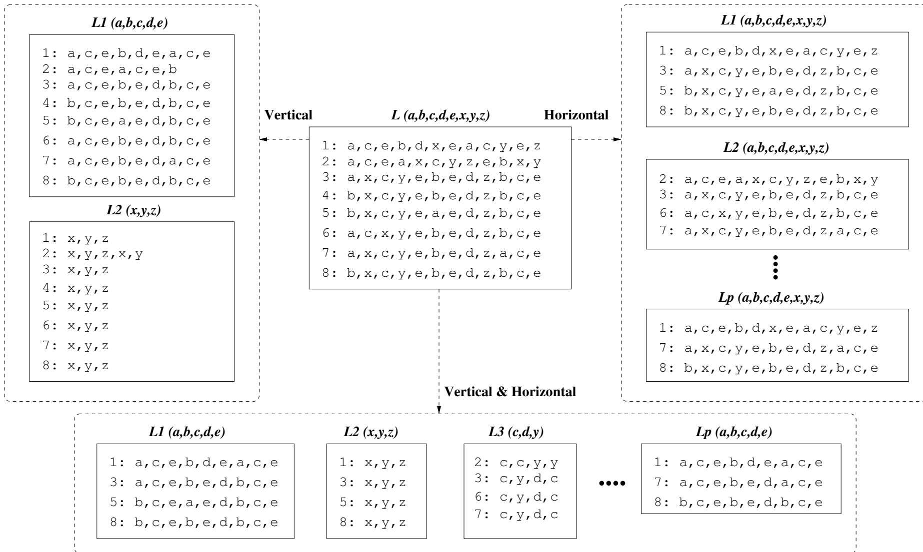

Fig. 3 Log decomposition by either horizontal or vertical selections, or a combination thereof.

Free-choice, *State Machines* or *Marked Graphs* Carmona (2012); de San Pedro and Cortadella (2016).

Key Research Findings in Decomposed Conformance Checking. Techniques for decomposed conformance checking have been proposed in the last years van der Aalst (2013); van der Aalst and Verbeek (2014); Munoz-Gama et al (2014); vanden Broucke et al (2014); de Leoni et al (2014); Verbeek (2017). Apart from the aforementioned techniques for decomposing a process model through a valid decomposition van der Aalst (2013); van der Aalst and Verbeek (2014); Munoz-Gama et al (2014), other approaches that have a different focus have been presented. In vanden Broucke et al (2014), the SESE-based decomposition approach is adapted to boost the computation so that it can be applied online. On a different perspective but again using SESE-based decomposition, the approach in de Leoni et al (2014) shows how data can also be taken into account.

Fitness, i.e., the capability of a model in reproducing a trace, strongly influences the idea of valid decomposition: on a valid decomposition, the fitness of the whole model with respect to the trace can be regarded as the fitness of the individual fragments (see Figure 2). Therefore, if none of the fragments has a fitness problem, then the initial model is fitting van der Aalst (2013). The opposite is not true in general: a valid decomposition where some fragment contains a fitness problem does not necessarily imply that there exist a real fitness problem. Also, by separating the initial problem into pieces, important conformance artifact like *optimal alignments*, i.e., alignments between model and log which contain the minimal number of deviations, cannot be guaranteed. There has been recent work investigating the two aforementioned issues Verbeek and van der Aalst (2016).

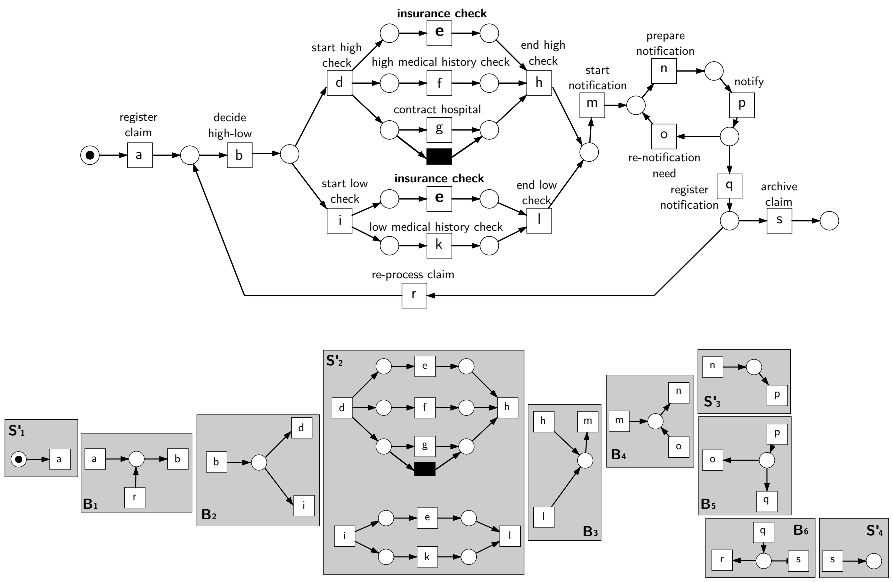

Fig. 4 Process model decomposition based on Single-Entry Single-Exit (SESE) components Munoz-Gama et al (2014).

Examples of Application

All areas where process mining techniques have been applied are amenable for the application of decomposed process discovery and conformance checking. It is suitable for large problem instances, e.g., in healthcare Mans et al (2015), software repositories, finances, among many others.

Future Directions for Research

There exist several challenges to face, so that decomposed techniques can be applied in more situations. Below we provide some key challenges that may be faced in the next years:

- *Decomposition in the large*: the techniques available so far do not always guarantee a bound in the complexity of the operations performed. Fundamental research needs to be developed so that lightweight approaches are proposed that guarantee this bound, opening the door to real-time or online approaches.
- *Recomposing the results*: both in Figures 1 and 2 one can see that the result of the decomposition may not always be a unique object, but several. To deploy a unique final result from these may be a challenge. For instance, in the case of decomposed conformance checking, to compute an alignment, it is well-known that there are situations where only a best-effort is possible Verbeek and van der Aalst (2016). Therefore, research on how to recompose the results will be needed.
- *Decomposition beyond Petri nets*: most of the techniques available for

decomposition rely strongly on the Petri net semantics. However, other modeling notations are also used, and in particular the industry adoption of Petri nets is low. Therefore, other notations like BPMN, will be considered as input for decomposition techniques.

- *Decomposition beyond control flow*: very few approaches have been presented that consider not only the control-flow of the process, but also other data attributes de Leoni et al (2014). However, the aforementioned techniques decompose only on the basis of the control-flow. Techniques that apply decomposition on different perspectives will enable a different perspective for decomposition.
- *Alternative theories for decomposed conformance checking*: currently, only the notion of valid decomposition has been explored as a viable strategy to decompose a process model. In the future, totally different alternatives to a valid decomposition may be studied.

Cross-References

Automated Process Discovery
Conformance checking
Event Log
Trace Clustering

References

van der Aalst WMP (2011) Process Mining - Discovery, Conformance and Enhancement of Business Processes. Springer

van der Aalst WMP (2012) Decomposing process mining problems using passages. In: Application and Theory of Petri Nets - 33rd International Conference, PETRI NETS 2012, Hamburg, Germany, June 25-29, 2012. Proceedings, pp 72–91, DOI 10.1007/978-3-642-31131-4_5, URL https://doi.org/10.1007/978-3-642-31131-4_5

van der Aalst WMP (2013) Decomposing petri nets for process mining: A generic approach. Distributed and Parallel Databases 31(4):471–507, DOI 10.1007/s10619-013-7127-5, URL https://doi.org/10.1007/s10619-013-7127-5

van der Aalst WMP, Verbeek HMW (2014) Process discovery and conformance checking using passages. Fundam Inform 131(1):103–138

van der Aalst WMP, Rubin VA, Verbeek HMW, van Dongen BF, Kindler E, Günther CW (2010) Process mining: a two-step approach to balance between underfitting and overfitting. Software and System Modeling 9(1):87–111, DOI 10.1007/s10270-008-0106-z, URL https://doi.org/10.1007/s10270-008-0106-z

van der Aalst WMP, Adriansyah A, van Dongen BF (2011) Causal nets: A modeling language tailored towards process discovery. In: CONCUR 2011 - Concurrency Theory - 22nd International Conference, CONCUR 2011, Aachen, Germany, September 6-9, 2011. Proceedings, pp 28–42

van der Aalst WMP, Kalenkova AA, Rubin VA, Verbeek E (2015) Process discovery using localized events. In: Application and Theory of Petri Nets and Concurrency - 36th International Conference, PETRI NETS 2015, Brussels, Belgium, June 21-26, 2015, Proceedings, pp 287–308

Adriansyah A (2014) Aligning observed and modeled behavior. PhD thesis, Technische Universiteit Eindhoven

Arnold A (1994) Finite Transition Systems. Prentice Hall

Bose RPJC, van der Aalst WMP (2009a) Abstractions in process mining: A taxonomy of patterns. In: Business Process Management, 7th International Conference, BPM 2009, Ulm, Germany, September 8-10, 2009. Proceedings, pp 159–175, DOI 10.1007/978-3-642-03848-8_12, URL https://doi.org/10.1007/978-3-642-03848-8_12

Bose RPJC, van der Aalst WMP (2009b) Context aware trace clustering: Towards improving process mining results. In: Proceedings of the SIAM International Conference on Data Mining, SDM 2009, April 30 - May 2, 2009, Sparks, Nevada, USA, pp 401–412

Bose RPJC, van der Aalst WMP (2009c) Trace clustering based on conserved patterns: Towards achieving better process models. In: Business Process Management Workshops, BPM 2009 International Workshops, Ulm, Germany, September 7, 2009. Revised Papers, pp 170–181

vanden Broucke SKLM, Munoz-Gama J, Carmona J, Baesens B, Vanthienen J (2014) Event-based real-time decomposed conformance analysis. In: On the Move to Meaningful Internet Systems: OTM 2014 Conferences - Confederated International Conferences: CoopIS, and ODBASE 2014, Amantea, Italy, October 27-31, 2014, Proceedings, pp 345–363, DOI 10.1007/978-3-662-45563-0_20, URL https://doi.org/10.1007/978-3-662-45563-0_20

Carmona J (2012) Projection approaches to process mining using region-based techniques. Data Min Knowl Discov 24(1):218–246, DOI 10.1007/s10618-011-0226-x, URL https://doi.org/10.1007/s10618-011-0226-x

Carmona J, de Leoni M, Depaire B, Jouck T (2016) Summary of the process discovery contest 2016. In: BPM 2016 Workshops, LNBIP 281 proceedings, pp 7–10

Ferreira DR, Zacarias M, Malheiros M, Ferreira P (2007) Approaching process mining with sequence clustering: Experiments and findings. In: Business Process Management, 5th International Conference, BPM 2007, Brisbane, Australia, September 24-28, 2007, Proceedings, pp 360–374

Greco G, Guzzo A, Pontieri L, Saccà D (2006) Discovering expressive process models by clustering log traces. IEEE Trans Knowl Data Eng 18(8):1010–1027, DOI 10.1109/TKDE.2006.123, URL http://dx.doi.org/10.1109/TKDE.2006.123

Hompes B, Buijs J, van der Aalst W, Dixit P, Buurman H (2015) Discovering deviating cases and process variants using trace clustering. In: Proceedings of the 27th Benelux Conference on Artificial Intelligence (BNAIC)

Leemans S (2009) Automated process discovery. In: Sakr and Zomaya (2017), pp 288–289

de Leoni M, Munoz-Gama J, Carmona J, van der Aalst WMP (2014) Decomposing alignment-based conformance checking of data-aware process models. In: On the Move to Meaningful Internet Systems: OTM 2014 Conferences - Confederated International Conferences: CoopIS, and ODBASE 2014, Amantea, Italy, October 27-31, 2014, Proceedings, pp 3–20, DOI 10.1007/978-3-662-45563-0_1, URL https://doi.org/10.1007/978-3-662-45563-0_1

Mans R, van der Aalst WMP, Vanwersch RJB (2015) Process Mining in Healthcare - Evaluating and Exploiting Operational Healthcare Processes. Springer Briefs in Business Process Management, Springer, DOI 10.1007/978-3-319-16071-9, URL https://doi.org/10.1007/978-3-319-16071-9

Mendling J, Dumas M (2009) Business process event logs and visualization. In: Sakr and Zomaya (2017), pp 288–289

Munoz-Gama J (2009) Conformance checking. In: Sakr and Zomaya (2017), pp 288–289

Munoz-Gama J, Carmona J, van der Aalst WMP (2014) Single-entry single-exit decomposed conformance checking. Inf Syst 46:102–122, DOI 10.1016/j.is.2014.04.003, URL https://doi.org/10.1016/j.is.2014.04.003

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

Sakr S, Zomaya A (eds) (2017) Encyclopedia of Big Data Technologies. Springer US

de San Pedro J, Cortadella J (2016) Mining structured Petri nets for the visualization of process behavior. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy, April 4-8, 2016, pp 839–846, DOI 10.1145/2851613.2851645, URL http://doi.acm.org/10.1145/2851613.2851645

Solé M, Carmona J (2012) A high-level strategy for c-net discovery. In: 12th International Conference on Application of Concurrency to System Design, ACSD 2012, Hamburg, Germany, June 27-29, 2012, pp 102–111

Verbeek HMW (2017) Decomposed replay using hiding and reduction as abstraction. T Petri Nets and Other Models of Concurrency 12:166–186

Verbeek HMW, van der Aalst WMP (2016) Merging alignments for decomposed replay. In: Application and Theory of Petri Nets and Concurrency - 37th International Conference, PETRI NETS 2016, Toruń, Poland, June 19-24, 2016. Proceedings, pp 219–239

Weerdt JD, vanden Broucke SKLM, Vanthienen J, Baesens B (2013) Active trace clustering for improved process discovery. IEEE Trans Knowl Data Eng 25(12):2708–2720, DOI 10.1109/TKDE.2013.64, URL http://dx.doi.org/10.1109/TKDE.2013.64