Unsupervised Event Abstraction in a Process Mining Context: A Benchmark Study

Greg Van Houdt[1(✉)], Benoît Depaire[1], and Niels Martin[1,2,3]

[1] Research Group Business Informatics, UHasselt - Hasselt University, Hasselt, Belgium
{greg.vanhoudt,benoit.depaire}@uhasselt.be
[2] Research Foundation Flanders (FWO), Brussels, Belgium
[3] Data Analytics Laboratory, Vrije Universiteit Brussel, Brussels, Belgium
niels.martin@vub.be

Abstract. Due to the rise of IoT, event data becomes increasingly fine-grained. Faced with such data, process discovery often produces incomprehensible spaghetti-models expressed at a granularity level that doesn't match the mental model of a business user. One approach is to use event abstraction patterns to transform the event log towards a more coarse-grained level and to discover process models from this transformed log. Recent literature has produced various (partial) implementations of this approach, but insights how these techniques compare against each other is still limited.

This paper focuses on the use of Local Process Models and Combination based Behavioural Pattern Mining to discover event abstraction patterns in combination with the approach of Mannhardt et al. [15] to transform the event log. Experiments are conducted to gain insights into the performance of these techniques. Results are very limited with a general decrease in fitness and precision and only a minimal improvement of complexity. Results also show that the combination of the process discovery algorithm and the event abstraction pattern miner matters. In particular, the combination of Local Process Models with Split Miner seems to improve precision.

Keywords: Process mining · Unsupervised learning · Event log · Abstraction

1 Introduction

Process Discovery focuses on the discovery of the process flow from an event log [2], in order to gain insights in the real execution of a business process [1]. However, when event logs are recorded at a fine-grained level, the activities in the discovered process model become increasingly less recognisable to the business users. Furthermore, fine-grained event data often result in incomprehensible spaghetti-models [22].

S. Leemans and H. Leopold (Eds.): ICPM 2020 Workshops, LNBIP 406, pp. 82–93, 2021.
https://doi.org/10.1007/978-3-030-72693-5_7

Against this background, [14] introduced a pattern-based approach to augment low-level events to higher-level activities, resulting in more insightful process models. Their original approach requires domain experts to provide event abstraction patterns which map low-level events to higher-level activities, which are subsequently used to transform the event log.

Mannhardt and Tax [16] studied the use of Local Process Models to learn these event abstraction patterns from data. More recently, [3] introduced a new unsupervised technique to discover event abstraction patterns that are compact and maximal, increasing the options to apply the technique proposed in [16] in an unsupervised manner. This raises the question how well these two options perform with the end goal in mind, i.e. to transform the log such that a process model is discovered which is more comprehensible and remains properly fitting and precise.

This paper describes a benchmark study of Local Process Models (LPMs) and Combination based Behavioral Pattern Mining (COBPAM) in combination with the approach in [16], focused on their capabilities to obtain models of lower complexity without sacrificing fitness and precision. Furthermore, performance differences between these two approaches were explored in order to identify underlying mechanisms at work. This resulted in following contributions:

- In contrast to previous studies, this study also considers the impact of event pattern abstraction on the understandability of the final process models, approximated by a broad set of complexity measures.
- This study provides an empirical comparison between LPM and COBPAM in combination with the method presented in [16], providing initial suggestions which of both event abstraction pattern miners performs best.
- This work provides empirical insights into the interaction between the process discovery algorithm and the event abstraction pattern miner with respect to their conjoint impact on fitness, precision and comprehensibility.

The remainder of this paper is structured as follows. Section 2 gives an outline of related work in the domain. Next, Sect. 3 defines the methodology for the benchmark study, as well as elaborates on the experimental design. Section 4 then gives an overview of the experiment's results before Sect. 5 concludes the paper.

2 Related Work

This paper builds further on the work in [16], which studied the use of LPMs to discover event abstraction abstraction patterns in combination with the approach in [14]. However, their work was slightly different than ours, as they only focused on fitness and precision, whereas we also take process model complexity into account.

LPM [23] can be used to discover event abstraction patterns in an unsupervised manner. It extends frequent pattern mining techniques to more complex patterns and aims to describe frequent behaviour in an event log in local patterns. In [21], LPM is extended with utility functions and constraints to mine more meaningful patterns, while [9] shows how high quality sets of a limited number of LPMs can be constructed. However, both latter approaches require some kind of domain expert interaction, which puts them outside the scope of our study.

Inspired by LPM, Acheli et al. [3] designed the Combination based Behavioural Pattern Mining (COBPAM) approach. It exploits a partial order on potential patterns to discover only those that are compact and maximal, i.e. least redundant.

The recently conducted review study of van Zelst et al. [25] proposes a taxonomy of supervised and unsupervised techniques for event abstraction. Some interesting unsupervised ones are global trace segmentation [11], HLPM-Mine [10], Bose et al. [8], Alharbi et al. [6], RefMod-Miner [18] and the work of Sánchez-Charles et al. [19]. This study focuses on LPM and COBPAM as the setup under consideration employs the approach in [16], which requires a defined process pattern between the low-level events in the event abstraction pattern. Another approach producing compatible patterns for this setup would be the RefMod miner [18]. Unfortunately, as no public implementation was available for this technique, it was not considered in this study.

It is worth noting that the combination of event abstraction pattern miners and the technique in [16] is not the only possibility to discover higher-level process models from low-level event data. For example, [20] presents a framework designed to transform location sensor data to an event log via interaction mining that business users can understand.

3 Methodology and Experimental Setup

This study takes the following algorithmic problem class as a starting point:

> A process model discovered from an event log is too complex to understand because the event log is too fine-grained. What is needed, is a technique which augments the event log to a higher abstraction level such that this transformed event log results in less complex discovered process models which are still correctly representing the underlying process.

This paper considers an algorithmic design to tackle this problem based on the approach in [14] in combination with two unsupervised abstract pattern discovery techniques, i.e. LPM [23] and COPBAM [3].

The quality of the algorithm design is defined on three criteria. Firstly, we want the process model discovered from the transformed log to be more comprehensible. The second and third criteria state that the model discovered from the transformed log should remain fitting and precise with respect to the original data.

3.1 Evaluation Method

To evaluate the comprehensibility of the model discovered from the transformed log, we use complexity as a proxy, which has been shown to be inversely related to the model understandability [17]. In total, ten complexity metrics were used, covering the four complexity dimensions identified in [13][1]. These complexity metrics are computed for the process models discovered from the transformed event logs.

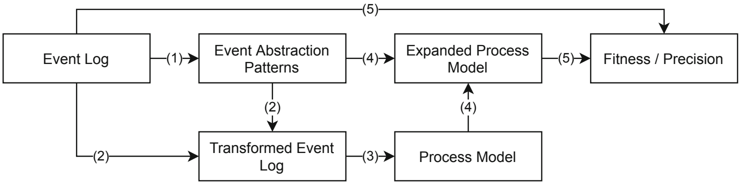

Fig. 1. Method to evaluate fitness and precision: (1) event abstraction patterns are discovered from the original event log, (2) the event log is transformed to higher abstraction level, (3) a process model is discovered, (4) the process model is expanded with the event abstraction patterns to the original granularity level, (5) precision and fitness are computed.

To evaluate the fitness and precision of the model discovered from the transformed event log with respect to the original event log, the same approach as in [16] was used. First, event abstraction patterns are discovered from the original event log. These event abstraction patterns map a local process pattern, defined at the original granularity level, to a higher level activity. Second, these patterns are used to transform the event log. Third, a process model is discovered from this transformed event log. Fourth, the event abstraction patterns are used to expand the process model into an expanded process model which is at the abstraction level of the original event log. This is done by replacing the higher-level activities by its corresponding local pattern. Fifth, the original event log is compared against the expanded process model to calculate fitness and precision values. Fitness is measured by the alignment-based fitness measure [4] and precision is measured with the alignment-based ETC precision measure [5].

3.2 Data

Six publicly available real-life event logs[2] with different characteristics are used in this study. Table 1 illustrates the variation among the logs in terms of number of events, number of activities, number of cases and number of distinct traces.

Regarding the BPI challenge 2019 log, similar to [7], a sample of the event log was taken for performance reasons. We preserved case variants containing at least 50 cases, leaving us with 71% of the events.

[1] This is done via the R package `understandBPMN` [13].
[2] The event logs were extracted from the 4TU Centre for Research Data in May 2020.

Table 1. Event log characteristics.

	# events	# activities	# cases	# distinct traces
Road traffic fine management	561.470	11	150.370	231
Hospital billing	451.359	18	100.000	1.020
Sepsis case	15.214	16	1.050	846
BPI 2019	1.135.258	27	224.768	192
BPI 2020 - request for payment	36.796	19	6.886	89
BPI 2020 - domestic declarations	56.437	17	10.500	99

3.3 Experimental Design

Every experiment consists of controlled variables that are of interest to the study. In our setup, the controlled variables are the event abstraction pattern miners and the process discovery algorithms used.

Two event abstraction pattern miners were considered in this study, i.e. LPM and COBPAM. For LPM the ProM implementation [24] was used with default parameter settings, except for the maximum number of transitions (5) and the number of patterns to discover (10). From the 10 patterns discovered, the top 3 according to the model ranking were selected, ignoring patterns subsumed by other better-scoring patterns. For COBPAM, the ProM implementation [24] is used with support threshold, language fit threshold and maximum depth set to respectively 0.7, 0.7 and 2 in accordance to the original work [3]. Patterns are sorted by support value and the top 3 patterns which are not subsumed by other patterns are selected.

Furthermore, two discovery algorithms were used, i.e. the split miner [7] and inductive miner infrequent [12]. Both are configured with their default values. This means there is a conversion step from BPMN to Petri net in the case of split miner[3]. The split miner is implemented as stand-alone Java application, while the inductive miner is accessed via the ProM framework [24].

For all 4 combinations of the 2 event abstraction pattern miners and 2 discovery algorithms, the following experiment was performed. For each event log, an initial process model was discovered and corresponding complexity, fitness and precision values were computed. These values serve as the baseline. Next, event abstraction patterns were mined from the event log and the top three were used to transform the event log to a higher abstraction level using the approach in [14]. The ProM implementation was used [24] and low-level events that were not mapped to higher-level events were kept in the transformed log. All other parameters were set at their default values. Finally, a new process model was discovered from the transformed event log and complexity, precision and fitness

[3] Done via the *convert BPMN diagram to Petri Net (Control Flow) plug-in in ProM.*

values were computed as described in Sect. 3.1. These measures can be compared against the baseline values to evaluate the impact of a specific event abstraction pattern miner for a given event log and discovery technique.

4 Empirical Results

This section will explain the results of our experiment per quality dimension. In total, the six event logs, three abstraction levels, two miners and 12 metrics for each model, resulted in 432 metrics. The raw result set is available online[4].

4.1 The Effect of Abstraction on Model Complexity

Complexity is measured by ten metrics in total. These are cognitive weight, token split, connector heterogeneity, control flow complexity, sequentiality, cyclicity, diameter, depth, density and coefficient of network connectivity [13]. From these ten, token split, connector heterogeneity, control flow complexity, sequentiality, cyclicity and the coefficient of network connectivity did not improve on average due to event abstraction for any activity pattern miner, regardless of the process model miner. The results concerning the remaining four are not uniform, however, as is shown in Table 2. The table describes, for each combination of miner and complexity metric, the number of event logs for which an improvement was observed and the average change. Note that a negative delta is considered an improvement, i.e. a reduction in complexity.

Table 2. Abstraction impact on cognitive weight, depth, density and diameter.

		Split miner		Inductive miner	
		# Improvements	Delta	# Improvements	Delta
Cognitive weight	LPM	3/6	−3.37%	2/6	2.54%
	COBPAM	4/6	−2.88%	2/6	3.45%
Depth	LPM	2/6	−7.14%	0/6	33.33%
	COBPAM	4/6	−35.71%	1/6	22.22%
Density	LPM	1/6	12.18%	3/6	−1.52%
	COBPAM	1/6	16.35%	3/6	−0.42%
Diameter	LPM	1/6	4.92%	3/6	−5.41%
	COBPAM	2/6	−2.19%	2/6	2.70%

Cognitive weight, which is the weighted sum of gateways and activities, seems to improve for both LPM and COBPAM when paired with the split miner. Depth, the amount of split minus join gateways, behaves in a similar fashion. The inductive miner has a tendency to generate more (parallel) gateways than

[4] https://github.com/gregvanhoudt/UnsupervisedEventAbstraction.

the split miner, and this effect is still present after abstraction. However, it is important to note that the baseline values for the inductive miner are already lower than the split miner's.

Density represents the percentage of sequence flows which are present compared with the theoretical maximum number of sequence flows. It shows the inverse behaviour of cognitive weight and depth, improving when paired with the inductive miner. However, the improvement here is much smaller than the deterioration with the split miner. A small density value indicates that the process is more sequential. The models substantiate this, as the split miner has a tendency to loop back to previous gateways to allow for repetitive behaviour, creating more sequence flows. In that regard, it is possible for depth to improve while density worsens.

The diameter metric only seems to improve, on average, for the combinations COBPAM-split miner and LPM-inductive miner. Also, even if diameter improves on average, the value decreases for the majority of the logs. Given that the results do not seem to correlate with a discovery miner or activity pattern miner, specific conclusions cannot be drawn for this metric. Further experimentation is required to obtain more conclusive results.

When considering all complexity measures simultaneously, we count 15 and 19 improvements for LPM and COBPAM, respectively. Although the difference is small, this seems to indicate COBPAM is slightly better in reducing complexity than LPM, independent of the process discovery algorithm. In general, we can conclude that, in our experimental setting, pattern-based event abstraction does not reduce the complexity of newly learnt models. However, caution is advised as we limited ourselves to only three patterns for each abstraction. Inserting additional patterns might have a positive impact on complexity. Keep in mind this will probably also impact fitness, and potentially precision.

4.2 The Effect of Abstraction on Model Fitness and Precision

The second facet of the study, the accuracy of process models, is measured by fitness and precision. Table 3 summarises the outcomes. A positive delta is now considered an improvement. Note that for precision - LPM - Inductive miner, we were unable to compute two values.

Table 3. Abstraction impact on fitness and precision.

		Split miner		Inductive miner	
		# Improvements	Delta	# Improvements	Delta
Fitness	LPM	0/6	−23.86%	1/6	−13.00%
	COBPAM	0/6	−32.04%	0/6	−22.93%
Precision	LPM	4/6	0.67%	1/4	−34.25%
	COBPAM	1/6	−4.78%	1/6	−26.00%

Regarding fitness, the data shows there is a severe negative effect: only one improvement is observed. Performing a t-test at the 5% significance level, the only insignificant difference was for LPM in combination with the inductive miner. Of course, the numbers have to be nuanced as the split miner generated a baseline of nearly perfect fitness, so an increase will be very difficult to accomplish. However, the magnitude of the decrease makes clear automated pattern-based abstraction negatively affects the fitness of new high-level process models.

One possible explanation is the overlap between activity patterns. Recall that fitness can only be calculated after the high-level model is expanded to again include the low-level event classes. This means one event class can now be present at multiple locations in the model. This can result in the obligation of an event class to be executed multiple times according to the high-level process models, which is not the case on the lower level. Also, the overlaps make it unclear which low-level event belongs to which high-level activity [16], generating potential confusion during the abstraction of the event log.

The precision metric also shows clear evolutions, although not as uniform as fitness. In fact, the average precision metric for LPM in combination with the split miner increased. However, the differences were not significant at the 5% significance level. A potential reason for decreases of precision is that we assume parallel relations between activity patterns. Should patterns overlap, it could be more reasonable to state that two patterns cannot co-exist. If two overlapping high-level patterns are present in the model, this is an introduction of additional behaviour. On the other hand, the goal of event log abstraction is hiding/grouping low-level behaviour, which should have a positive influence on precision.

In general, fitness only increased once for LPM without observing any improvements for COBPAM. The precision metric improved 5 and 2 times for LPM and COBPAM respectively, with the majority of improvements located at LPM-split miner in particular. Results suggest that LPM performs better when interested in fitness and precision of abstracted models.

4.3 Discussion

Overall, the study shows that the use of event abstraction pattern miners to transform an event log with the purpose of discovering less complex process model with good fitness and precision, has limited success. On average, it seems that this approach, using either LPM or COPBAM, results in a decrease of fitness and precision, with only limited effect on complexity.

Fitness typically takes a hit when abstracting the event log, which is not completely unexpected as abstraction patterns hide complex behaviour which can no longer be accounted for by the miner. It is also remarkable that the impact on fitness appears to be correlated to the process discovery algorithm. Before abstraction, the split miner produces the best-fitting models. After abstraction, however, this fitness drops heavily, to the extent that the inductive miner produces better-fitting models at that stage.

Precision also has a tendency to deteriorate, with the exception for the combination of LPM with split miner. For this combination, in the majority of the cases we saw an improvement of precision and the average effect was also positive. It is remarkable that this result is not observed for the combination with COBPAM and that LPM cannot reproduce these effects with inductive miner. This again confirms the pattern that there is some kind of interaction between the process discovery algorithm and the event abstraction pattern miner.

As for complexity, for most of the measures no clear improvement was observed. The only pattern that could be distinguished, which supports the goal of this approach, is the slight improvement of cognitive weight and depth for split miner and the improvement of density for inductive miner. Again, these results hint at an interaction between the discovery miner and abstraction pattern miner.

Overall, we can conclude that this approach combined with LPM and COBPAM has limited results. Future research will be needed to improve these results in order to make them impactfull enough for practical use. Based on our empirical analysis, a potential direction for future research is to delve into the interaction between the discovery algorithm and the abstraction pattern miner. It is clear that there are mechanisms at work and understanding these could open up avenues for improved algorithms. Another path worth investigating is the automatic discovery of how activity patterns interrelate. The approach in [16] has parameters which define which patterns can or cannot co-exist and in what type of interrelation. The current event abstraction pattern miners do not provide this type of information.

Finally, based on these mixed empirical results, one must be careful to draw strong conclusions with respect to the performance of LPM versus COBPAM. One might suggest that both approaches are competitive to each other, with the exception of the combination of split miner with LPM, which actually appears to improve the precision of the models on average. As with respect to reducing complexity, COBPAM seems to have a small edge over LPM, albeit too small to make conclusive statements.

4.4 Limitations

This experiment can be extended in several ways. First of all, a new approach to evaluate LPMs was recently proposed [9], which is not implemented in our work yet. This new evaluation acknowledges the excessive amount of overlapping patterns and disregards confidence and determinism as quality measures. For COBPAM, we only have access to support and language fit scores. A more advanced scoring and selection technique of activity patterns could have improved the experiment, obtaining less overlapping patterns as with the meaningful LPMs [9].

On the other hand, the current study fixes the number of activity patterns that are taken into consideration to three. Mannhardt and Tax [16] concluded that the optimal number of patterns varied per event log. No doubt the same applies to this study.

Next, this experimental setup uses six event logs. Each of them returns six process models: a low-level, a LPM-abstracted and a COBPAM-abstracted model for both the split miner and the inductive miner. Therefore, this study compares 36 process models. To obtain a larger number of observations to draw conclusions from, this number of event logs can easily be increased.

Finally, as discussed in Sect. 2, the RefMod-Miner also satisfies the requirements to take part of this experiment, yet no public implementation is available.

5 Conclusion

In this paper, local process models and the combination based behavioural pattern mining approach are put against each other in unsupervised event log abstraction. The goal was to produce process models at a higher abstraction level with better comprehensibility, while still being well-fitting and properly precise.

However, the experiments show only limited results. While some aspects of complexity show possibilities for improvements, they seem tied to the process model miner. Fitness gets a significant hit overall and precision only tends to improve for the combination of LPM and Split Miner.

Future research is required with respect to the interactions between activity pattern miners and process discovery algorithms. This could allow for more accurate abstraction techniques. Also, discovering meaningful and more precise activity patterns is an interesting research track. But perhaps more importantly, the possibility to discard our assumption about only parallel inter-pattern relations must be explored. Being able to learn this from data could greatly improve the abstraction quality.

References

1. Van der Aalst, W.: Process Mining: Data Science in Action, pp. 3–23. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4_1
2. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: discovering process models from event logs. IEEE Trans. Knowl. Data Eng. **16**(9), 1128–1142 (2004)
3. Acheli, M., Grigori, D., Weidlich, M.: Efficient discovery of compact maximal behavioral patterns from event logs. In: Giorgini, P., Weber, B. (eds.) CAiSE 2019. LNCS, vol. 11483, pp. 579–594. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_36
4. Adriansyah, A., van Dongen, B.F., van der Aalst, W.M.: Conformance checking using cost-based fitness analysis. In: 2011 IEEE 15th International Enterprise Distributed Object Computing Conference, pp. 55–64. IEEE (2011)
5. Adriansyah, A., Munoz-Gama, J., Carmona, J., van Dongen, B.F., van der Aalst, W.M.: Measuring precision of modeled behavior. Inf. Syst. e-bus. Manag. **13**(1), 37–67 (2015)
6. Alharbi, A., Bulpitt, A., Johnson, O.A.: Towards Unsupervised Detection of Process Models in Healthcare. Studies in Health Technology and Informatics, pp. 381–385. IOS Press, Netherlands (2018)

7. Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Polyvyanyy, A.: Split miner: automated discovery of accurate and simple business process models from event logs. Knowl. Inf. Syst. **59**(2), 251–284 (2019)
8. Jagadeesh Chandra Bose, R.P., van der Aalst, W.M.P.: Abstractions in process mining: a taxonomy of patterns. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 159–175. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03848-8_12
9. Brunings, M., Fahland, D., van Dongen, B.: Defining meaningful local process models. In: Proceedings of the International Workshop on Algorithms and Theories for the Analysis of Event Data 2020. CEUR-WS.org (2020). http://ceur-ws.org/Vol-2625/paper-01.pdf
10. Folino, F., Guarascio, M., Pontieri, L.: Mining multi-variant process models from low-level logs. In: Abramowicz, W. (ed.) BIS 2015. LNBIP, vol. 208, pp. 165–177. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19027-3_14
11. Günther, C.W., Rozinat, A., van der Aalst, W.M.P.: Activity mining by global trace segmentation. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 128–139. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_13
12. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Process and deviation exploration with inductive visual miner. In: Limonad, L., Weber, B. (eds.) Business Process Management Demo Sessions (BPMD 2014). CEUR Workshop Proceedings, vol. 1295, pp. 46–50. Eindhoven, The Netherlands. CEUR-WS.org (2014)
13. Lieben, J., Jouck, T., Depaire, B., Jans, M.: An improved way for measuring simplicity during process discovery. In: Pergl, R., Babkin, E., Lock, R., Malyzhenkov, P., Merunka, V. (eds.) EOMAS 2018. LNBIP, vol. 332, pp. 49–62. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00787-4_4
14. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P., Toussaint, P.J.: From low-level events to activities - a pattern-based approach. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 125–141. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_8
15. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M., Toussaint, P.J.: Guided process discovery-a pattern-based approach. Inf. Syst. **76**, 1–18 (2018)
16. Mannhardt, F., Tax, N.: Unsupervised event abstraction using pattern abstraction and local process models. In: Gulden, J., et al. (eds.) CEUR Workshop Proceedings, vol. 1859, pp. 55–63. CEUR-WS.org (2017)
17. Mendling, J.: Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness, vol. 6. Springer Science & Business Media (2008)
18. Rehse, J.-R., Fettke, P.: Clustering business process activities for identifying reference model components. In: Daniel, F., Sheng, Q.Z., Motahari, H. (eds.) BPM 2018. LNBIP, vol. 342, pp. 5–17. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11641-5_1
19. Sánchez-Charles, D., Carmona, J., Muntés-Mulero, V., Solé, M.: Reducing event variability in logs by clustering of word embeddings. In: Teniente, E., Weidlich, M. (eds.) BPM 2017. LNBIP, vol. 308, pp. 191–203. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74030-0_14
20. Senderovich, A., Rogge-Solti, A., Gal, A., Mendling, J., Mandelbaum, A.: The ROAD from sensor data to process instances via interaction mining. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) CAiSE 2016. LNCS, vol. 9694, pp. 257–273. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39696-5_16

21. Tax, N., Dalmas, B., Sidorova, N., van der Aalst, W.M., Norre, S.: Interest-driven discovery of local process models. Inf. Syst. **77**, 105–117 (2018)
22. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.P.: Event abstraction for process mining using supervised learning techniques. In: Bi, Y., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2016. LNNS, vol. 15, pp. 251–269. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-56994-9_18
23. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.: Mining local process models. J. Innov. Digit. Ecosyst. **3**(2), 183–196 (2016)
24. van Dongen, B.F., de Medeiros, A.K.A., Verbeek, H.M.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: The ProM framework: a new era in process mining tool support. In: Ciardo, G., Darondeau, P. (eds.) ICATPN 2005. LNCS, vol. 3536, pp. 444–454. Springer, Heidelberg (2005). https://doi.org/10.1007/11494744_25
25. van Zelst, S.J., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining: literature review and taxonomy. Granular Comput. 1–18 (2020)