Understanding automated feedback in learning processes by mining local patterns

Galina Deeva*[1], Jochen De Weerdt[1]

KU Leuven, Faculty of Economics and Business, Department of Decision Sciences and Information Management

Abstract. Process mining, and in particular process discovery, provides useful tools for extracting process models from event-based data. Nevertheless, certain types of processes are too complex and unstructured to be able to be represented with a start-to-end process model. For such cases, instead of extracting a model from a complete event log, it is interesting to zoom in on some parts of the data and explore behavioral patterns on a local level. Recently, local process model mining has been introduced, which is a technique in-between sequential pattern mining and process discovery. Other process mining methods can also used for mining local patterns, if combined with certain data preprocessing. In this paper, we explore discovery of local patterns in the data representing learning processes. We exploit real-life event logs from JMermaid, a Smart Learning Environment for teaching Information System modeling with built-in feedback functionality. We focus on a specific instance of feedback provided in JMermaid, which is a reminder to simulate the model, and locally explore how students react to this feedback. Additionally, we discuss how to tailor local process model mining to a certain case, in order to avoid the computationally expensive task of discovering all available patterns, by combining it with other techniques for dealing with unstructured data, such as trace clustering and window-based data preprocessing.

Keywords: process discovery, local process models, automated feedback, trace clustering

1 Introduction

Nowadays, most educational institutions use a variety of information systems to support educational processes. In most cases, these information systems have a logging functionality that allows for monitoring and analyzing the process it supports. These data can be analyzed from a variety of different perspectives, showing different aspects of learning. Traditional data mining techniques have been used to build predictive models, acquire better understanding of learning processes, or give recommendations to students and educators. However, the majority of traditional data mining techniques do not have an objective to analyze, discover and visually represent a complete educational process. Process mining

* Corresponding author.

does have this objective, as it aims to extract process-related knowledge from event logs stored by information systems [1].

Process mining provides useful tools for extracting knowledge from event-based data [2]. One of the most insightful tasks of process mining is process discovery, i.e. extracting a process model that represents the event log from start to end. However, while providing useful results in many cases, process discovery is not always able to represent complex and unstructured processes. In learning analytics, the data oftentimes contains a large number of activity types, making it hard to be represented by a single process model and resulting in so-called spaghetti models or flower models. An example of this can be found in our previous study [3], in which we discussed that discovering a process model from behavioral data in a Massive Open Online Course (MOOC) in a global way is a rather challenging task, while methods that work on a more local level, such as sequence mining, yield more insightful results.

Recently introduced in [4], Local Process Model (LPM) mining can be positioned in-between of process and sequence mining. As such, LPM mining can cope with unstructured processes on a local level, avoiding the difficulty of representing the process as a whole. In addition, LPM mining is also capable to grasp concepts that are hard to represent with most sequential pattern mining approaches, such as concurrency, choice, loop and sequential composition [4]. The goal of local process model discovery is to find patterns that occur in the event log on a local level. Such local models can be very insightful, especially if the task of describing the behavior in the complete event log is too complex, but also in cases when it is interesting to focus on the local behavior. Another advantage of LPM discovery is that it is capable to grasp relations between more than 3 items, which is often not possible with most of sequence mining techniques.

In this work, we explore discovery of local patterns from event-based data from JMermaid[1], a Smart Learning Environment for teaching Information System (conceptual) modeling, enriched with a feedback mechanism that provides students with real-time automated feedback. Our goal is to employ LPM mining and other techniques in order to study the behavior of students after they receive feedback. We focus on a certain instance of feedback, which is a reminder to simulate the created conceptual model, and then locally explore the patterns that follow this feedback on real-life datasets from JMermaid.

As discussed in [5], it is a computationally difficult task to discover local patterns in the event log with too many activity types. Due to that, in this paper we discuss and compare possible ways of discovering LPM that are tailored to a specific problem, as to reduce the computational complexity of discovering all available local models. We explore combinations of different techniques for dealing with unstructured data, such as trace clustering and window-based data preprocessing.

The paper is structured as follows. In Section 2, recent studies on process mining in an educational context and LPM mining are reviewed. Next, the JMermaid learning environment, research questions and scenarios for mining local

[1] http://merode.econ.kuleuven.ac.be/mermaid.aspx

patterns in the context of a smart learning environment (SLE) with automated feedback are discussed in Section 3. Subsequently, the data preparation and results are presented in Section 4. Finally, Section 5 outlines our findings and gives directions for future work.

2 Related Work

2.1 Process mining in an educational context

There has been a variety of studies that applied process mining within the field of education, in which cases it has been frequently addressed as Educational Process Mining (EPM). EPM aims to build complete educational process models that are able to reproduce the observed behavior, check if the modeled behavior matches the behavior observed, manage extracted information to make the tacit knowledge explicit and to facilitate a better understanding of the processes [1].

Recently, there has been an increasing number of studies that applied EPM to real-life cases. The objectives of such applications vary widely. For example, Weerapong, Porouhan and Premchaiswadi [6] analyzed the control flow perspective of student registration at the university, with the goal to solve issues that might occur during this process. Vahdat et al. [7] used Fuzzy Miner and a complexity metric to estimate the understandability of process models of engineering laboratory sessions. Trcka and Pechenizkiy [1] explored online-assessment data to investigate how students navigate between multiple choice questions, and whether this process can be improved with automated feedback. More recently, Juhaňák, Zounek and Rohlíková [8] analyzed students' quiz-taking behavior patterns in a learning management system Moodle.

A few studies aimed to explore process mining in a more global setting of Massive Open Online Courses (MOOCs). For example, Mukala et al. [9] applied the dotted chart, process discovery and conformance checking techniques to a Coursera MOOC dataset. In another study, Deeva et al. [3] investigated the applicability of process discovery techniques for dropout prediction with a case study on a MOOC from the EdX platform. Additionally, Maldonado-Mahauad et al. [10] used process mining for exploring frequent interaction sequences in three Coursera MOOCs.

A common goal in EPM research is to find behavioral patterns typical for certain groups of learners, or to compare the behavior of student clusters. For example, Schoor and Bannert [11] aimed to find discriminative process patterns for high and low performing groups in a collaborative learning task, showing that successful students perform regulatory activities with a higher frequency and in a different order than less successful students. Another comparison of different student groups was performed by van der Aalst, Guo and Gorissen [12], where records of watching video lectures are analyzed with comparative process mining using process cubes, which allowed to discriminate between the learning behavior of student subgroups, such as successful vs. unsuccessful, male and female, local and foreign, as well as the behavior within different chapters of the course. Moreover, Papamitsiou and Economides [13] exploited comprehensive process models with concurrency patterns in order to detect and model guessing behavior in computer-based testing, revealing common patterns for students with different goal-orientation levels.

Previous research involving the JMermaid learning environment can be found in [14] and [15], where process mining was used for revealing modeling behavior patterns that can be related to certain learning outcomes. We expect that more insightful patterns can be observed in event logs from JMermaid by mining local models instead of complete process models due to the complexity of underlying processes. Thus, it is anticipated that more sophisticated methods for mining local patterns can facilitate deeper understanding of these data.

More information regarding EPM can be found in the most recent survey of this topic by Bogarín, Cerezo and Romero [16].

2.2 Local Process Model mining

Local process model mining can be positioned in-between process discovery and episode/sequential pattern mining. The concept and the procedure of LPM discovery was introduced by Tax et al. [4], where it was also compared with other techniques for mining local patterns in unstructured event logs, such as process mining algorithms Declare miner, Fuzzy miner and Episode Miner, and the sequential pattern mining algorithm PrefixSpan. The authors showed that LPM discovery is capable of deriving insightful patterns that in some cases cannot be discovered with aforementioned techniques. Additionally, they proposed metrics for assessing the quality of local process models.

The same authors in [5] expanded their findings by introducing heuristic approaches for coping with computational difficulties of discovering LPMs. These approaches are Markov clustering, log entropy and the relative information gain heuristics, which are used to create projections of event logs. The example event log from this study contained 1734 activity types, which is too difficult to deal with with a straightforward approach described in [4], since the computational complexity grows significantly for such substantial amount of variety in the logs. To solve this problem, Tax et al. exploited the idea of discovering local models from the projections of event logs, containing only activities of interest for a particular LPM.

Dalmas, Tax and Norre [21] introduced the heuristics for high-utility LPM discovery. The authors aimed to reduce computational complexity of the LPM mining task by specifying a utility function based on business insights. It was concluded, however, that the search space of LPMs cannot be reduced without loss. Similarly, Tax et al. [22] presented goal-driven discovery of LPMs based on utility functions and constraints for addressing particular business questions.

For fine-granular event logs it is useful to combine events to a higher level of abstraction, which is typically done with clustering techniques. In the study by Mannhardt and Tax [23], local process models are used for automated event abstractions, resulting in overall process models with more balanced precision and fitness scores.

3 Mining local patterns in a smart learning environment

3.1 Automated feedback in JMermaid

In this work, we analyze event-based data from the JMermaid learning environment, developed in our Management Informatics Research Group at the Faculty of Business and Economics, KU Leuven for teaching Information Systems modeling. It is based on MERODE, a method for Enterprise Systems development [24], and used in the Architecture and Modeling of Management Information Systems (AMMIS) course[2].

JMermaid is enriched with a feedback mechanism that provides personalized immediate feedback in an automated way. Based on the findings of a previous study of JMermaid [15], which indicated that frequent simulation of a conceptual model is strictly correlated with the successful learning outcome of a student, we implemented a learning dialog that reminds students to simulate their model after a certain number of actions is conducted in the tool (Figure 1).

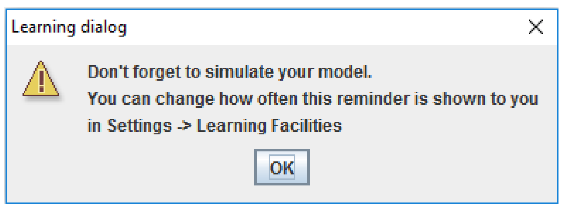

Fig. 1: Reminder to simulate the model provided as feedback to students in the JMermaid tool

We analyze local patterns that involve this instance of feedback, which is addressed below as the *simulation reminder* (SR). The study aims to tackle the following research questions:

1) How to discover typical patterns of student reactions to automated feedback in a smart learning environment?

2) What are the optimal ways to discover those patterns, which are both insightful and computationally efficient?

3) How to grasp differences in reaction to feedback between low and high-performing students? Is there any correlation between students' reactions to feedback and their final scores?

3.2 Window-based preprocessing for detecting automated feedback

To see the immediate student reactions to automated feedback, we focus on events that directly follow SR. We choose 10 following events; however, this number is not restrictive and can be adjusted in the future analyses. Next, we disregard a few outlier traces that contain less than 10 events after SR,

[2] http://onderwijsaanbod.kuleuven.be/syllabi/e/D0I71AE.htm

since it most likely means that a student stopped working in the tool instead of reacting on feedback. Thus, SR is always acting as a start event in a set of traces containing 11 events (SR and 10 following events). Subsequently, an artificial end event is added to each trace. As a result of such window-based preprocessing, the obtained data consists of traces with 12 events, which are $< SR, a, b, c, d, e, f, g, h, i, j, End >$. The data in this format is further addressed as the *filtered data*.

3.3 Scenarios for detecting automated feedback

Five scenarios for extracting local patters containing SR are discussed.

1) LPM mining applied to the complete event logs. The first approach is to apply the LPM mining algorithm to the complete event logs without window-based preprocessing, as to extract all possible LPMs, and subsequently filter them focusing on the models with the simulation reminder. As discussed above, this brute-force approach is expected to be computationally expensive, and might benefit from further optimization. Nevertheless, it is also possible that reducing the search space will cause an information loss [21].

2) LPM mining applied to the filtered data. The second scenario is to apply LPM discovery to the filtered event logs to investigate if data preprocessing can reduce computational complexity and facilitate LPM discovery.

3) LPM mining combined with trace clustering and the filtered data. Similarly to LPM mining, trace clustering techniques aim to resolve the issue of overly unstructured process models that are discovered from event logs with, e.g., a large number of activity types [26], [27]. In trace clustering, similar traces are grouped together so as to focus on similar behavioral scenarios within an event log. Trace clustering techniques could potentially work well on the data representing learning processes, in which different groups of students might follow several distinct learning paths. Nevertheless, not all event data can be potentially clustered.

The third approach is to apply a trace clustering technique k-gram [28], available in Guide Tree Miner plugin in ProM, to the filtered data, and subsequently apply LPM mining, to investigate whether trace clustering can facilitate data exploration in the context of unstructured learning processes.

4) Process discovery applied to the filtered data. The fourth approach is to apply process discovery techniques Inductive Miner and Alpha miner on the filtered data. We want to investigate whether plain process discovery is more or at least equally capable of extracting meaningful local patterns than LPM mining, if being combined with intelligent subsetting of the data.

5) Process discovery combined with trace clustering and the filtered data. Similarly to 3), the last approach is to cluster the traces and subsequently apply process discovery.

4 Experimental evaluation

4.1 Data description and preparation

We analyze event logs of the students performing 23 distinct modeling tasks during in-class exercise sessions. An overview of the data is given in Table 1. To compare the behavioral patterns of low and high-performing students, the students are divided into two groups according to their performance. For this we apply k-means clustering with their final scores for the course and the grades for two intermediate assignments (which are not part of the final score) as features, and obtain two clusters of 43 and 21 students. Dataset 1 (D1) and Dataset 2 (D2) are the complete event logs for low-performing (Group 1) and high-performing (Group 2) students, respectively. Filtered Dataset 1 (FD1) and Filtered Dataset2 (FD2) are the event logs preprocessed as described in Section 3.2. Note that in case of D1 and D2 we use *User_id* as a case id, thus analyzing data from the student perspective, and in case of FD1 and FD2 we use each trace with the simulation reminder as a separate case, thus analyzing each particular reaction to this feedback.

Table 1: An overview of the datasets used in the experimental evaluation

Dataset	Performance	# of students	# of activity types (L)	# of activity types (H)	# of events	# of SR
D1	Low	43	63	16	24296	276
FD1	Low	40	52	16	3076	276
D2	High	21	64	16	21789	232
FD2	High	21	49	16	2684	232

An example of an event log from JMermaid is shown in Figure 2. *ActivityL* represents activities on a more fine-granular level, i.e. at the low level of abstraction, which has around 60 (for D1 and D2) or 50 (for FD1 and FD2) activity types. The JMermaid tool also logs some aspects of the modeling process, such as "view" and ''category" (structural (S) or behavioral (B)), as well as a type of performed action (Feedback, Create, Delete, Edit, Customize, Error, Check, Save). A combination of the ''type" of action with structural or behavioral aspect gives 16 variations, including the simulation reminder. This less fine-granular view, i.e. high level of activity abstraction, is referred in the logs as *ActivityH*.

Timestamp	User_id	View	Category	Type	ActivityL	ActivityH
2017-03-03 14:28:23.645	User1	EDG	S	FEEDBACK	Simulation reminder	Simulation reminder
2017-03-03 14:28:23.661	User1	EDG	S	FEEDBACK	Dependency feedback	FEEDBACK+S
2017-03-03 14:28:23.661	User1	EDG	S	CREATE	Create dependency	CREATE+S
2017-03-03 14:28:38.654	User1	EDG	S	CUSTOMIZE	Move object	CUSTOMIZE+S
2017-03-03 14:28:42.773	User1	EDG	S	CUSTOMIZE	Move object	CUSTOMIZE+S

Fig. 2: An example of an event log from JMermaid

4.2 Results

The scenarios described in Section 3.3 are applied to the data with low (L) and high (H) levels of activity abstraction for Group 1 and Group 2. The results are summarized in Table 2.

Table 2: A summary of the results of the five scenarios

#	Group 1 (L)	Group 2 (L)	Group 1 (H)	Group 1 (H)
1	The algorithm returned no results	The algorithm returned no results	The algorithm returned no results	The algorithm returned no results
2	The algorithm returned no results	The algorithm returned no results	LPMs are obtained in a very long time (more than 1 hour)	The algorithm returned no results
3	The algorithm returned no results	The algorithm returned no results	LPMs are obtained very fast (less than 1 minute)	LPMs are obtained very fast (less than 1 minute)
4	The obtained models are too unstructured and flower-like	The obtained models are too unstructured and flower-like	The models provide insightful patterns	The models provide insightful patterns
5	The models provide insightful patterns	The models provide insightful patterns	The models provide insightful patterns	The models provide insightful patterns

1) LPM mining. For the first scenario, LPM mining applied to a complete event log was not able to discover local patterns for any level of event abstraction. This result is expected and can be explained by very high levels of computational complexity. For low level of abstraction, LPM mining was still not able to discover local models even if combined with data filtering or trace clustering. Since the amount of activity types in this case is close to 50, this result is also explained by a high variety of activity combinations. As discussed in [5], more than 17 activity types might already be too many for the plain LPM mining to handle, requiring further data optimization. Therefore high level of activity abstraction with 16 activity types was expected to be easier to analyze. Nevertheless, in the second scenario, LPM mining was able to discover LPMs for Group 1, but not for

Group 2. As also seen from the results of the other techniques, the event logs for Group 2 might contain more distinct activity combinations, for which it is more challenging to discover local patterns. Finally, LPM mining combined with trace clustering discovered LPMs very fast, which indicated that trace clustering is capable of combining traces of learning behavior to meaningful clusters, making it easier to a discovery technique to deal with such data.

Examples of discovered LPMs are provided in Figure 3. First of all, the local patterns discovered for both Group 1 and 2 are similar, which, given the large amount of the patterns containing SR, can be difficult to interpret in terms of addressing our goal of distinguishing between two groups of students. Second, most of the LPMs contain a choice between SR and other activities, which is not useful for analyzing pattern that follow SR. To conclude, LPM mining is capable to provide interesting patterns in learning processes if combined with other data optimization techniques, but it is not optimal for the purpose of analyzing feedback.

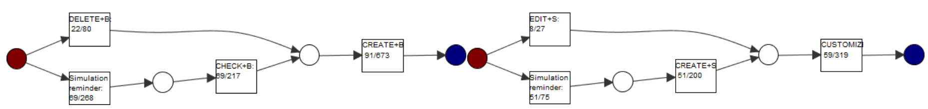

Fig. 3: LPMs discovered in the second (left) and third (right) scenarios

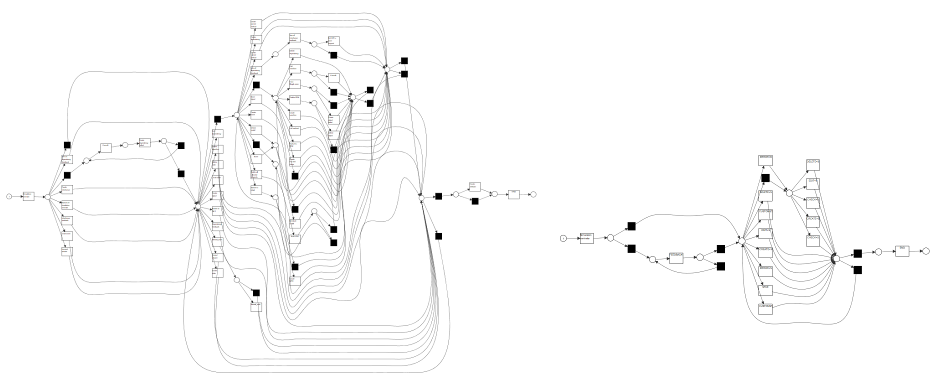

Fig. 4: Process models discovered by Inductive Miner for low (left) and high (right) levels of abstraction in the event logs

2) Process discovery. For experimental evaluation, we apply both Inductive Miner and Alpha++ miner available in ProM process mining toolkit. Since the results are similar, we provide the examples of models discovered by Inductive Miner (Figure 4). The process models discovered from the data with the larger number of activity types are too unstructured and flower-like, even with data preprocessing. On the other hand, the models derived from the logs with higher level of activity abstraction are more structured and can generally provide insights into student reactions to feedback.

3) Trace clustering techniques. The models discovered by Inductive Miner after applying k-gram trace clustering are capable of giving useful insights in case of both high and low levels of activity abstractions (Figure 5).

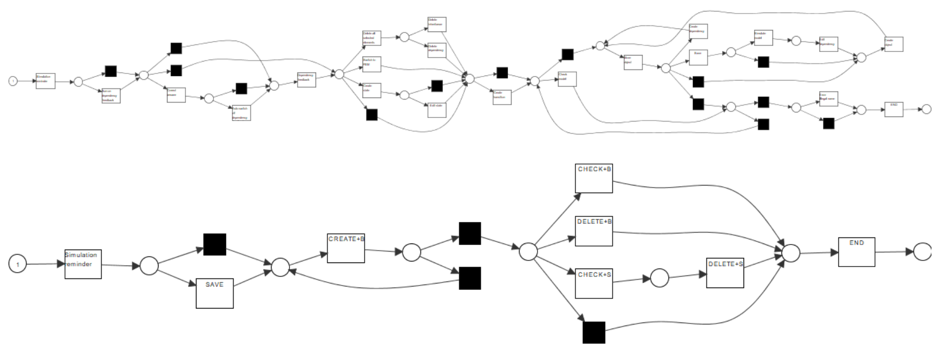

Fig. 5: Process models discovered in the fifth scenario for the low (top) and high (bottom) levels of abstraction in the event logs

5 Conclusion and Future Work

In this paper, the possible ways to discover students reactions to automated feedback in a Smart Learning Environment (SLE) are investigated. We explored Local Process Mining (LPM) discovery and its combinations with other techniques for working with unstructured data, as well as window-based preprocessing of the data. The discussion contained five scenarios for discovering local patterns tailored to a specific case, which included 1) LPM mining on complete event logs, 2) LPM mining on filtered data, 3) LPM mining combined with trace clustering and filtered data, 4) process discovery on filtered data, and 5) trace clustering on filtered data. These scenarios are evaluated on two datasets with log data of high and low-performing students, with the purpose of finding behavioral patterns typical for certain student groups. Two setups with different levels of activity granularity are investigated; one containing 50 activity types and the other with 16 aggregated activity types.

The results reveal that plain LPM discovery is hardly capable to deal with processes with low levels of activity abstraction (50 activity types in our case). In case of less variety in the logs (16 activity types), LPM discovery still requires some adequate data preparation to be able to discover local models. Similarly, process discovery on filtered data is able to achieve meaningful results only in case of less variety in the logs. However, the models discovered with process discovery are more suitable for addressing our research questions, since they give more insights into patterns that follow the feedback. Finally, trace clustering combined with filtered data is capable to achieve meaningful results in case of high as well as low levels of activity granularity. The models discovered on clusters of traces are the most insightful for our task.

This study provides initial steps for exploring reactions to automated feedback in SLEs. Given the limited scope of the paper, we do not focus on a detailed interpretation of the discovered patterns, but rather show possible ways of their discovering. In future work, it will be worthwhile to focus on interpretation of the discovered patterns. Furthermore, other tasks are possible in the context of SLE's data, for which LPM mining might provide more useful results.

References

1. Trcka, N., Pechenizkiy, M.: From local patterns to global models: Towards domain driven educational process mining. In: Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on, IEEE (2009) 1114–1119
2. Van der Aalst, W.M.: Process mining: data science in action. Springer (2016)
3. Deeva, G., De Smedt, J., De Koninck, P., De Weerdt, J.: Dropout prediction in moocs: A comparison between process and sequence mining. In: International Conference on Business Process Management, Springer (2017) 243–255
4. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.: Mining local process models. Journal of Innovation in Digital Ecosystems **3**(2) (2016) 183–196
5. Tax, N., Sidorova, N., van der Aalst, W.M., Haakma, R.: Heuristic approaches for generating local process models through log projections. In: Computational Intelligence (SSCI), 2016 IEEE Symposium Series on, IEEE (2016) 1–8
6. Weerapong, S., Porouhan, P., Premchaiswadi, W.: Process mining using α-algorithm as a tool (a case study of student registration). In: ICT and Knowledge Engineering (ICT & Knowledge Engineering), 2012 10th International Conference on, IEEE (2012) 213–220
7. Vahdat, M., Oneto, L., Anguita, D., Funk, M., Rauterberg, M.: A learning analytics approach to correlate the academic achievements of students with interaction data from an educational simulator. In: Design for Teaching and Learning in a Networked World. Springer (2015) 352–366
8. Juhaňák, L., Zounek, J., Rohlíková, L.: Using process mining to analyze students' quiz-taking behavior patterns in a learning management system. Computers in Human Behavior (2017)
9. Mukala, P., Buijs, J., Van Der Aalst, W.: Exploring students learning behaviour in moocs using process mining techniques. Department of Mathematics and Computer Science, University of Technology, Eindhoven, The Netherlands (2015)
10. Maldonado-Mahauad, J., Pérez-Sanagustín, M., Kizilcec, R.F., Morales, N., Munoz-Gama, J.: Mining theory-based patterns from big data: Identifying self-regulated learning strategies in massive open online courses. Computers in Human Behavior **80** (2018) 179–196
11. Schoor, C., Bannert, M.: Exploring regulatory processes during a computer-supported collaborative learning task using process mining. Computers in Human Behavior **28**(4) (2012) 1321–1331

12. van der Aalst, W.M., Guo, S., Gorissen, P.: Comparative process mining in education: An approach based on process cubes. In: International Symposium on Data-Driven Process Discovery and Analysis, Springer (2013) 110–134
13. Papamitsiou, Z., Economides, A.A.: Process mining of interactions during computer-based testing for detecting and modelling guessing behavior. In: International Conference on Learning and Collaboration Technologies, Springer (2016) 437–449
14. Sedrakyan, G., Snoeck, M., De Weerdt, J.: Process mining analysis of conceptual modeling behavior of novices–empirical study using jmermaid modeling and experimental logging environment. Computers in Human Behavior **41** (2014) 486–503
15. Sedrakyan, G., De Weerdt, J., Snoeck, M.: Process-mining enabled feedback:tell me what i did wrong vs.tell me how to do it right. Computers in human behavior **57** (2016) 352–376
16. Bogarín, A., Cerezo, R., Romero, C.: A survey on educational process mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **8**(1) (2018)
17. Maggi, F.M., Mooij, A.J., van der Aalst, W.M.: User-guided discovery of declarative process models. In: Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on, IEEE (2011) 192–199
18. Günther, C.W., Van Der Aalst, W.M.: Fuzzy mining–adaptive process simplification based on multi-perspective metrics. In: International Conference on Business Process Management, Springer (2007) 328–343
19. Leemans, M., van der Aalst, W.M.: Discovery of frequent episodes in event logs. In: International Symposium on Data-Driven Process Discovery and Analysis, Springer (2014) 1–31
20. Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.: Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: proceedings of the 17th international conference on data engineering. (2001) 215–224
21. Dalmas, B., Tax, N., Norre, S.: Heuristics for high-utility local process model mining. In: Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data. (2017) 106–121
22. Tax, N., Dalmas, B., Sidorova, N., van der Aalst, W.M., Norre, S.: Interest-driven discovery of local process models. arXiv preprint arXiv:1703.07116 (2017)
23. Mannhardt, F., Tax, N.: Unsupervised event abstraction using pattern abstraction and local process models. arXiv preprint arXiv:1704.03520 (2017)
24. Snoeck, M.: Enterprise Information Systems Engineering: The MERODE Approach. Springer Publishing Company, Incorporated (2014)
25. Serral, E., De Weerdt, J., Sedrakyan, G., Snoeck, M.: Automating immediate and personalized feedback taking conceptual modelling education to a next level. In: Research Challenges in Information Science (RCIS), 2016 IEEE Tenth International Conference on, IEEE (2016) 1–6
26. Song, M., Günther, C.W., Van der Aalst, W.M.: Trace clustering in process mining. In: International Conference on Business Process Management, Springer (2008) 109–120
27. De Weerdt, J., vanden Broucke, S., Vanthienen, J., Baesens, B.: Active trace clustering for improved process discovery. IEEE Transactions on Knowledge and Data Engineering **25**(12) (2013) 2708–2720
28. Bose, R.J.C., Van der Aalst, W.M.: Context aware trace clustering: Towards improving process mining results. In: Proceedings of the 2009 SIAM International Conference on Data Mining, SIAM (2009) 401–412