Discovering the Impact of Students' Modeling Behavior on their Final Performance

Galina Deeva(✉), Monique Snoeck, and Jochen De Weerdt

Research Center for Management Informatics, KU Leuven,
Naamsestraat 69, 3000 Leuven, Belgium
galina.deeva@kuleuven.be

Abstract. Conceptual modeling is an important part of Enterprise Modeling, which is a challenging field for both teachers and learners. Creating conceptual models is a so-called 'ill-structured' task, i.e. multiple good solutions are possible, and thus students can follow very distinct modeling processes to achieve successful learning outcomes. Nevertheless, it is possible that some principles of modeling behavior are more typical for high-performing rather than low-performing students, and vice versa. In this study, we aimed to discover those patterns by analyzing logged student modeling behavior with process mining, a set of tools for dealing with event-based data. We analyzed data from two individual conceptual modeling assignments in the JMermaid modeling environment based on the MERODE method. The study identified the presence of behavioral patterns in the modeling process that are indicative for better/worse learning outcomes, and showed what these patterns are. Another important finding is that students' performance in intermediate assignments is as well indicative of their performance in the whole course. Thus, predicting these problems as early as possible can help teachers to support students and change their final outcomes to better ones.

Keywords: Conceptual modeling · Domain modeling
Process mining · Education · Learning analytics

1 Introduction

Recently, learning analytics and educational data mining have provided teachers with new tools to facilitate learning. Some of the important objectives of learning analytics are to understand and predict student performance and behavior, and to improve teaching support. With growing availability and accessibility of learners' data, it became possible to analyze students' behavior, and even provide them with feedback automatically and in real-time.

R. A. Buchmann et al. (Eds.): PoEM 2018, LNBIP 335, pp. 335–350, 2018.
https://doi.org/10.1007/978-3-030-02302-7_21

Nevertheless, performing such behavior analyses is not always a straightforward task, especially for so-called ‘ill-structured’ domains with multiple good solutions. One of such domains, conceptual modeling is challenging for both teachers and learners. While creating a conceptual model, students can follow very different modeling processes to achieve successful learning outcomes. However, some principles of modeling behavior may be more typical for high-performing rather than low-performing students, and vice versa. In this paper, we approach conceptual modeling from a process-oriented perspective and aim to discover behavioral patterns by analyzing logged modeling behavior with process mining.

Process mining enables the creation of process models based on event log data that are captured in an information system [1]. In this research, process mining is used to gain more insight into student behavior in the context of an individual course on conceptual modeling. Specifically, students enrolled in the course of ‘Architecture and Modeling of Information Systems’ are given two individual assignments. These assignments require students to create a conceptual model in the JMermaid modeling environment, which logs all student modeling activities. We analyze these log data to find patterns that are indicative of better or worse learning outcomes, as well as to discover the correlation between the scores on each individual assignment and the final score.

1.1 Research Questions

The main goal of this study is to improve the understanding of how certain sequences of modeling activities correlate with better/worse learning outcomes. As such, we aim to address the following research questions:

1. Is there a correlation between the performance in intermediate assignments and the final score of the course?
2. Are there any recognizable patterns of a modeling process that can be correlated with better or worse learning outcomes?
3. What are these patterns, if they exist?

The paper is organized as follows. In Sect. 2, we present an overview of recent literature on conceptual modeling education and educational process mining. Next, the methodology of the study, including the data collection process, is given in Sect. 3. The results of the analysis are presented in Sect. 4. Subsequently, the main findings and limitations of the study are discussed in Sect. 5. Finally, Sect. 6 summarizes the findings and gives directions for future work.

2 Related Work

2.1 Conceptual Modeling

A conceptual model (also known as domain model) is a complete and holistic view of a system based on conceptual but precise qualitative assumptions about its concepts (entities), their interrelationships, and their behavior [2]. A conceptual model of an information system provides an abstract model of an enterprise and enables the design of an information system [2,3].

Conceptual modeling requires problem analysis and solving, which are by nature inexact skills. As a consequence, teaching such skills to novice modelers is a difficult task: novice modelers produce incomplete, inaccurate, ambiguous and/or incorrect models in their early careers [4]. There are many reasons that make teaching and learning conceptual modeling difficult. First, the quality of a conceptual model depends on a variety of knowledge factors: the knowledge of modeling concepts, of the modeling language and of the domain to be modeled are key factors affecting the quality of a model [5]. These issues can be addressed by providing students with the proper amount of supportive information about required knowledge [6]. Second, different procedural factors also affect the outcome of a modeling effort. Observations of the modeling process of novices indicate that they follow a linear problem-solving pattern, thereby focusing on one task at a time rather than switching between modeling activities [7]. Furthermore, novice modelers show poorly adapted cognitive schemata with regards to the identification of relevant triggers for verifying the quality of models [4], a problem that is exacerbated by the absence of established validation procedures [8]. These factors pertain to the process of modeling, which is why in this study we try to tackle a process-oriented view on conceptual modeling.

2.2 Educational Process Mining

Many studies applied process mining within the field of education, in which case it is often referred to as educational process mining (EPM). Recently, there was an increasing number of studies that exploited EPM in different real-life scenarios. For example, Weerapong et al. [9] analyzed the control flow perspective of student registration at the university. Juhaňák et al. [10] studied students' quiz-taking behavior patterns in a learning management system Moodle.

A common goal in EPM is to find behavioral patterns that are typical for certain groups of learners. For example, van der Aalst et al. [11] compared different student groups with comparative process mining using process cubes, discriminating between the learning behavior of successful vs. unsuccessful, male and female, local and foreign subgroups, as well as the behavior of students within different chapters of the course. Similarly, Papamitsiou and Economides [12] 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.

In the field of business process models, there is a recent research stream that studies the process of process modeling (PPM). For instance, Pinggera et al. [13] performed a cluster analysis on the log data from large-scale modeling sessions and identified three distinct styles of modeling. Claes et al. [14] introduced a way to visualize different steps that modelers conduct to create a process model. These and similar studies on PPM give useful examples of insights into business process modeling process that can be obtained with process mining. The main difference of our study is its focus on the process of conceptual modeling with the MERODE method, since typical behavioral patterns of modelers in different domains and different modeling languages may as well differ.

Previous research involving the JMermaid learning environment can be found in [15,16], where process mining was used for revealing modeling behavior patterns that can be related to certain learning outcomes. Event log data captured in JMermaid was used to analyze student performance in a group assignment. The main difference between the current study and [16] is that we analyze student behavior and performance at the individual level instead of a group level, and thus aim to provide recommendations for improving modeling skills of the individual learners, as well as investigate how the scores obtained in individual assignments are correlated with the final scores of the course.

3 Methodology

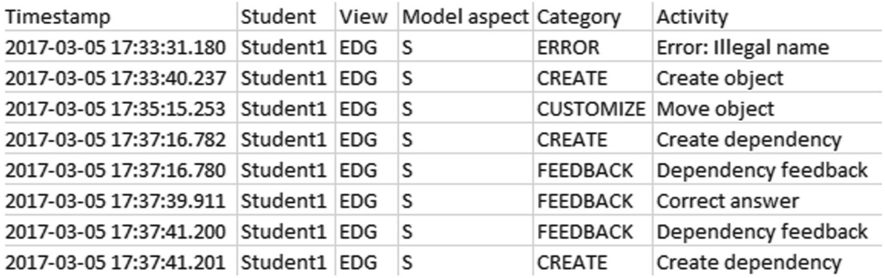
3.1 The JMermaid Modeling Environment

We analyze behavioral data from the JMermaid modeling 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 [17], and used in the Architecture and Modeling of Management Information Systems (AMMIS) course[1]. The main objective of the AMMIS course is to introduce the learners to the latest techniques for object-oriented analysis and enterprise information system modeling. Students have to learn how to create an information system's conceptual model, which includes three modeling perspectives: the structural properties (domain object types and their associations) are captured by means of a class diagram (called Existence Dependency Graph (EDG)), the behavioral aspects of domain object types are described by means of Finite State Machines (FSM), and the interactions between domain objects are captured by means of an object-event table. The JMermaid tool allows drawing these different types of diagrams, and offers the students support for the verification and simulation of their models.

3.2 Logging Functionality in JMermaid

JMermaid is capable to log student activities in the format shown in Fig. 1. The log file contains each activity that a student conducted or triggered in the system, timestamped to milliseconds. There is a total of 60 possible *Activities*, and they are further abstracted into eight *Categories*, which can be seen as higher level activities. The *View* indicates which of the three parts of the model, i.e. EDG, OET or FSM, is being currently worked on, and *Model aspect* can be structural (S, i.e. working on the class diagram) or behavioral (B, i.e. working on the FSMs or OET).

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



Timestamp	Student	View	Model aspect	Category	Activity
2017-03-05 17:33:31.180	Student1	EDG	S	ERROR	Error: Illegal name
2017-03-05 17:33:40.237	Student1	EDG	S	CREATE	Create object
2017-03-05 17:35:15.253	Student1	EDG	S	CUSTOMIZE	Move object
2017-03-05 17:37:16.782	Student1	EDG	S	CREATE	Create dependency
2017-03-05 17:37:16.780	Student1	EDG	S	FEEDBACK	Dependency feedback
2017-03-05 17:37:39.911	Student1	EDG	S	FEEDBACK	Correct answer
2017-03-05 17:37:41.200	Student1	EDG	S	FEEDBACK	Dependency feedback
2017-03-05 17:37:41.201	Student1	EDG	S	CREATE	Create dependency

Fig. 1. An example of an event log from JMermaid

3.3 Data Collection

During the semester, students enrolled in the course are required to complete two individual take-home assignments. Both assignments include a specification document that states all the requirements. Students have to transform the requirements to a semantically correct conceptual model using the JMermaid modeling environment, which captures student data to event logs.

For the first assignment, students were given a case description on a problem related to a gas station company, for which they were instructed to create a class diagram (EDG). The second assignment included a given class diagram and a description of behavioral aspects, based on which students created FSMs for domain object types with non-default behavior and define interaction aspects in the OET. Population and other data statistics are provided in the next section.

4 Results

4.1 Data Description

We use the data of students who participated in two assignments (referred to as *HW1* and *HW2*) during two academic years (2017 and 2018). The first assignment is focused on modeling structural aspects of the model, while the second one involves modeling the behavioral part for a given class diagram. The models of the students are evaluated on a scale from 1 (fail) to 5 (excellent). Based on these marks, we identified two groups: low-performing students, who received 1 (fail), and high performing students, who received 4 (very good) or 5 (excellent). For this analysis, we don't take into account the students whose assignments were ranked as 2 and 3, since the goal of this study is to find the differences in behavior of worse vs. best scoring students (for the assignment). An overview of the data is given in Table 1, including the number of students and the total number of activities performed in each subgroup.

4.2 Correlation between the Assignment Scores and the Final Score

The distributions of exam scores for each assignment score for the years 2017 and 2018 are shown in Figs. 2 and 3, respectively. The exam scores from 1 to

Table 1. An overview of the data

Dataset	Assignment score	# of students	# of activities	Average # of activities per student
hw1-17-h	4 or 5	13	1609	123.8
hw1-17-l	1	19	2765	145.5
hw1-18-h	4 or 5	7	809	115.6
hw1-18-l	1	12	1386	115.5
hw2-17-h	4 or 5	11	3146	286
hw2-17-l	1	6	1226	204.3
hw2-18-h	4 or 5	3	1067	355.7
hw2-18-l	1	6	1476	246

20 are subdivided into 3 categories: fail (below 10), satisfactory (from 10 to 13) and good (14 and more). As previously explained, the assignments are evaluated with a score from 1 to 5; it is also possible that the student didn't hand in the assignment (shown as "no assignment" in the graphs).

For all the four cases, the students who obtained 4 and 5 for the assignments have performed with distinction (score 14 or more) in the exam. In fact, for the HW1 in 2017 HW2 in 2018, 100% of the students who scored 4 and 5 have obtained good exam scores. Additionally, in 2017, it is easy to see that students who scored at least 2 for both assignments were capable to pass the course with

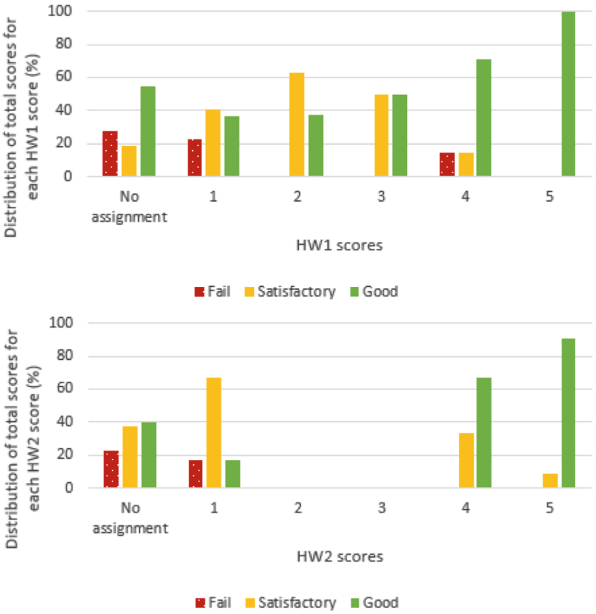


Fig. 2. Distribution of total scores for each HW1 and HW2 scores (2017)

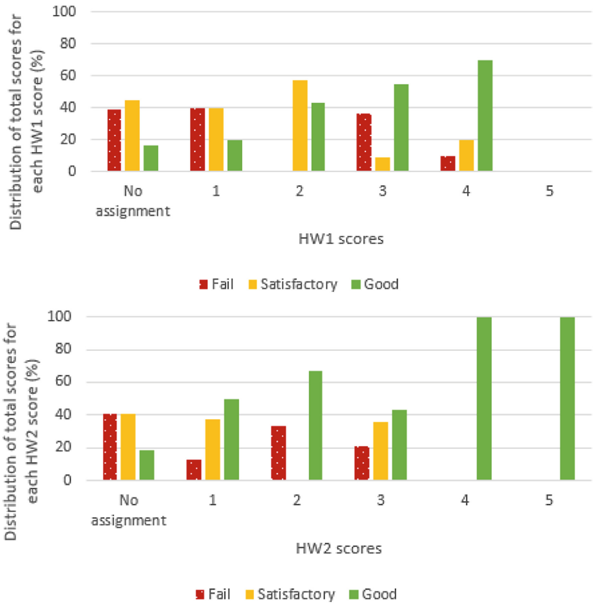


Fig. 3. Distribution of total scores for each HW1 and HW2 scores (2018)

a satisfactory or good mark. In 2018, some students who scored 2 or higher still failed the course, but it was in most cases a minority within the group.

Interestingly, in 2017 no students received marks 2 or 3 for the second assignments. This means that for the second task most students have either improved the quality of their models and received a better score (and have passed the course successfully, as seen in Fig. 2), or this quality decreased and they failed the second assignment, which made it more likely for them to fail the course as well. In 2018 this trend of the second assignment to be more predictive of final performance is not as strong, however, there is a clear tendency for the better scoring students to also perform much better in the exam.

For the students who didn't hand in the assignments, it can be observed that while there is still a chance they will pass the course, their chances to fail the course are the highest from all the groups, and even higher than for the students who made the assignments and failed it. This is especially observed for 2018, in which 40% of the "no assignment" group failed the exam. While the scores of the assignments are found to be predictive for the exam score, it would be interesting to be able to provide students with feedback already while they make their assignment. We therefore analyze the modeling processes in Sects. 4.3 and 4.4.

4.3 Analysis of the Activity Frequencies

Categories of Activities. Before discovering process models, we analyzed the frequency of activities of students with the Disco process mining tool. First, we looked into categories of activities. Figure 4 (HW1) and Fig. 5 (HW2) give an overview of relative activities occurrences (given in percentage) for high and low performing students for both analyzed years. The following patterns can be observed. First of all, there is a tendency to perform some CHECK activity more frequently within the high-scoring students compared to the low-scoring students. This category includes activities for validating the quality of the model, e.g. simulate the model, check the errors, etc. This is an important finding, since it confirms the results from the previous study [16], in which this tendency has been reported in performing a group assignment. This trend can be seen for all the cases, independently of the context of the assignment.

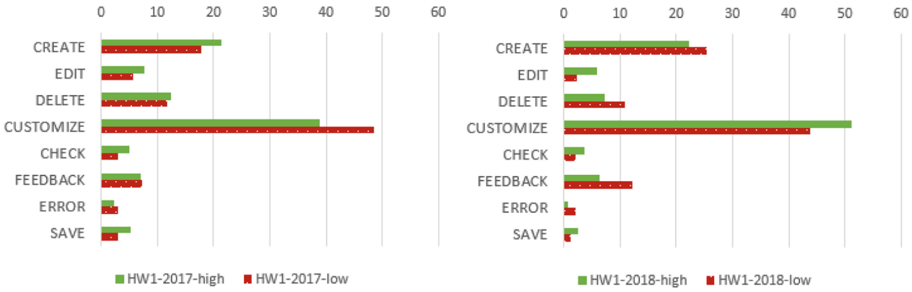


Fig. 4. Frequency of activity categories in HW1 in 2017 and 2018. The values are given as percentage of the total number of activities

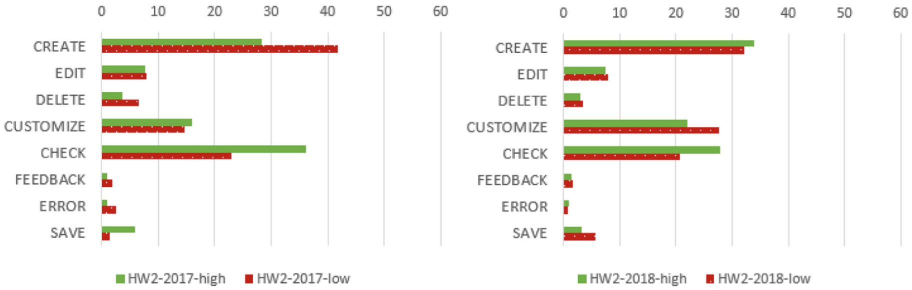


Fig. 5. Frequency of activity categories in HW2 in 2017 and 2018. The values are given as percentage of the total number of activities

Secondly, in three out of four graphs, it is observed that low-scoring students have more ERROR activities than their better scoring peers. This result might seem intuitive, nevertheless it is an important step towards predicting the performance of students using their event-based data. We can assume that low-scoring students make more errors while modeling, and it can be captured by the modeling tool.

Similarly, for both assignments in 2017 and for the first assignment in 2018, there is a pattern of performing the SAVE activity, i.e. save the model, more frequently for high-scoring students. A possible explanation could be that high-performing students save more often in view of simulating their model, but also that they are in general more careful about the modeling process.

Next, independently of the context of the task, there is no clear correlation between frequencies of CREATE, DELETE and CUSTOMIZE activities and better performance. There is a tendency for EDIT activity to be more frequent for students who scored well in the first assignment, but it doesn't hold for the second task. CREATE, EDIT and DELETE activities are used to build the model, while the CUSTOMIZE category contains activities which help the modeling process, but don't affect the quality of the model, e.g. show grid in the tool or move the object. An overall conclusion for these categories could be that the "quality" of performed activities matters more than the quantity. Creating more objects, events or FSMs won't necessarily result in a better quality model.

Finally, there is a slight tendency of low-scoring students to receive more feedback (FEEDBACK category) than high-scoring students do. This can be due to the fact that, first, by making more errors or waiting too long before simulating their model, low-scoring students trigger more automated feedback. Second, low-scoring students might feel that they need more help from the system, and thus don't switch off learning dialogs or actively request learning reports. Although currently JMermaid has a limited amount of feedback implemented, this finding might give a direction for further research in this area.

Fine-Granular Level of Activities. Next, we analyze frequencies of occurrence of student activities on a more fine-granular level. Figure 6 (assignment 1) and Fig. 7 (assignment 2) provide an overview of the most frequent activities. Note that the set of activities is different for the two assignments. Similarly to the analysis of the activity categories, we can see that successful students simulate their model significantly more often than less successful students. "Simulate model" is one of the possible actions in the CHECK category, which provides students with the most insights about the quality of their model. Thus, it might be concluded that model simulation could potentially enhance model quality.

For HW1, we observe that the better students switch much more frequently between views than the low-scoring ones. When performing behavioral modeling,

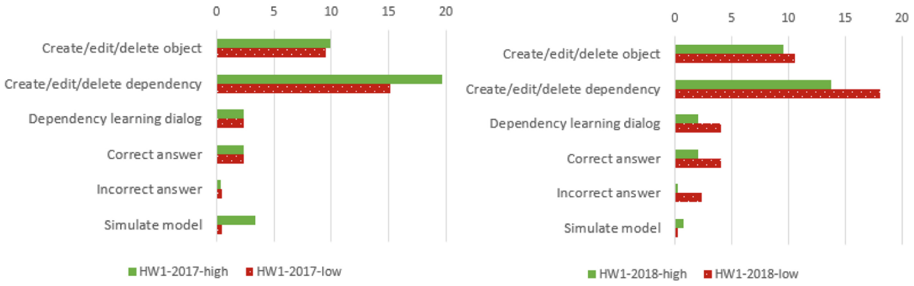


Fig. 6. Most frequent activities in HW1 in 2017 and 2018. The values are given as percentage of the total number of activities

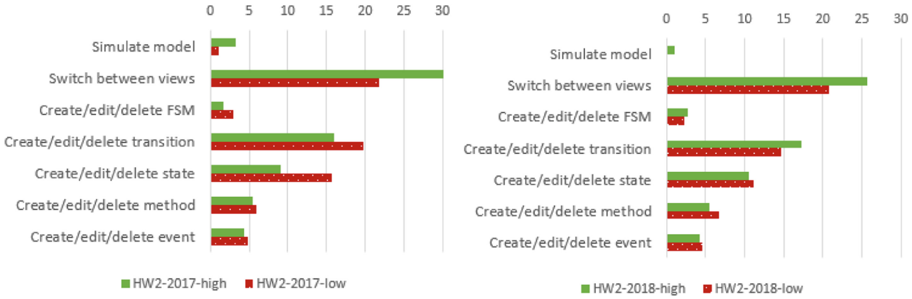


Fig. 7. Most frequent activities in HW2 in 2017 and 2018. The values are given as percentage of the total number of activities

this switch can be considered as a validation activity used to verify the behavioral model against the default behavior implied by the data model [18,19].

Next, for the first assignment, it can be observed that low-scoring students give more incorrect answers to the learning dialogs (Fig. 6). Interestingly, it seems that low-scoring students give more or at least the same number of correct answers compared to the high-scoring students. The reason for this could be that these students are simply asked more questions because of their actions. Nevertheless, the number of incorrect answers can serve as a predictive feature of future problems with the model.

This time we look into CREATE, EDIT and DELETE activities from another angle. Instead of looking at the number of CREATE actions, independently of the created entity, we compare possible activities for each distinct entity, such as object, dependency, FSM, and so on. In general, the conclusion is similar to the one previously obtained: it seems there is no strong correlation between the quantity of building model activities, but it is quality that matters. This finding generally holds for both assignments, except for create/edit/delete actions on methods, events and states. These activities (which all belong to OET or FSM view) are being performed slightly more frequently by the low-scoring students. This correlation might indicate that low-scoring students might be less sure while creating behavioral aspects of the model, and thus delete and edit these types of elements more often. These is confirmed by Fig. 5, in which indeed low-performing students delete and edit more often than their better scoring peers. This pattern, however, can only be observed for the behavioral aspects of the model, while for the structural ones there are no indications of the quantity of the building actions being indicative of a better/worse score.

4.4 Analysis of Process Models

For the sake of brevity, we only provide process models for the second assignment for the high level of activity abstraction (category of activity). The reasons for this choice are that, first, as described in Sect. 4.3, HW2 seems to be more predictive of the final score. Second, HW2 is slightly bigger, and as such, the log files contain more student actions on average (see Table 1). However, similar patterns are observed in the process models for the first assignment as well.

The process models are given in Figs. 8 (high-scoring students, 2017), Fig. 9 (low-scoring students, 2017), Fig. 10 (high-scoring students, 2018) and Fig. 11 (low-scoring students, 2018). As modeling is a complex task, there is inherently a very large variation of possible process paths. The visual inspection of the models seems to indicate the absence of clearly dominant patterns for good or

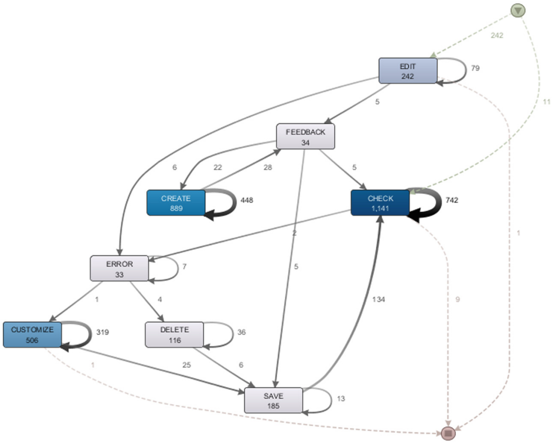


Fig. 8. Process model created in Disco for HW2, high-scoring students, 2017

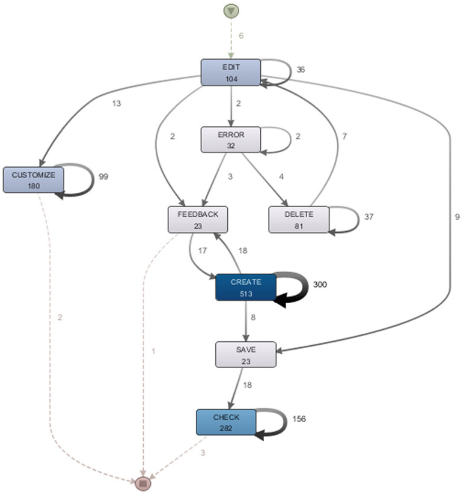


Fig. 9. Process model created in Disco for HW2, low-scoring students, 2017

bad processes for modeling a single perspective. It is nevertheless interesting to see the reaction of students to FEEDBACK events. Low-performing students tend to react to feedback with CREATE (2017) or CUSTOMIZE (2018) events, while better scoring students often CHECK their model after receiving feedback.

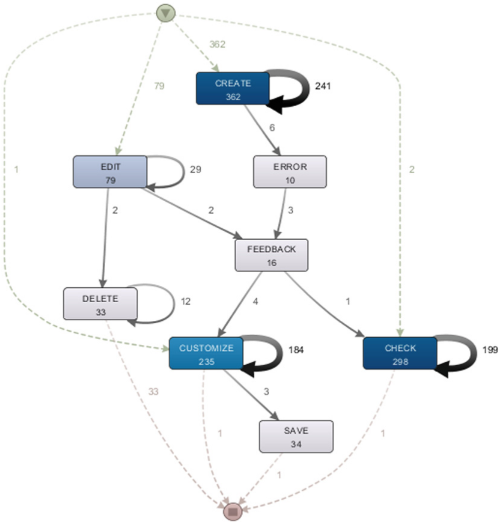


Fig. 10. Process model created in Disco for HW2, high-scoring students, 2018

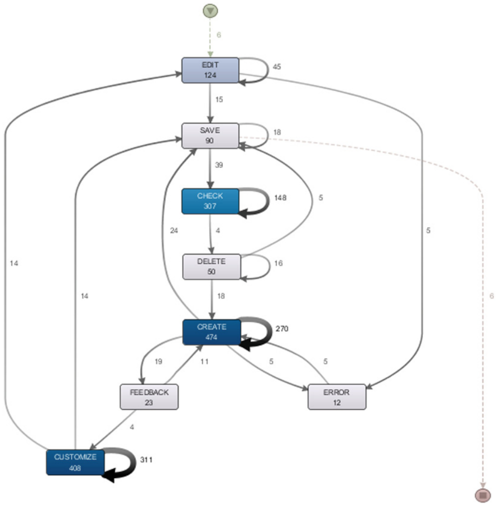


Fig. 11. Process model created in Disco for HW2, low-scoring students, 2018

5 Discussion

This study addresses the question of how the modeling process can be correlated with learning outcomes. In particular, we investigated the modeling process for "part-tasks" where students address a single perspective of a modeling task: data modeling only, or behavioral modeling only for a given data model. The analysis of the scores clearly indicates that the outcome of the process of these part-tasks are indicative for the final achievement of the course, yet the goal of the research is to find features of the modeling process that are indicative for the quality of the outcome, thus allowing to give process-oriented feedback, rather than outcome feedback only. The seemingly absence of dominant patterns indicative for good or bad results in the process models shown in Sect. 4.4, can easily be explained by the large variety of possible paths a student can follow when elaborating models, and the fact that in this case we investigated only part-task modeling behavior for fairly simple assignments and for a small sample. Previous research investigated modeling behavior for a large and complex whole-task assignment. There we more clearly witnessed a series of dominant patterns, such as the iterative modeling as opposed to linear modeling, a pattern also revealed in [7].

Yet the analysis of the frequency of the activities in Sect. 4.3 also revealed that better students switch views much more frequently than their low-scoring peers. This confirms the superiority of the iterative modeling, also at the part-task level. Furthermore, novices' inability to identify triggers for verifying the quality of models identified in [4] is also confirmed as being experienced more by low-scoring students than by high-scoring students as evidenced by their lower number of 'check' activities.

In general, the results of this research illustrate that there are some patterns that can influence the model quality. These patterns are summarized below.

1. Better performing students validate their model more often while modeling. More specifically, simulating the model and cross-checking with the data model when doing behavioral modeling can significantly improve its quality.
2. Low-scoring students tend to make more errors, such as entering illegal name or connecting wrong types of objects. This could be attributed to a better knowledge background of higher scoring students. Most importantly, this can be captured by the modeling tool and used as a feature in a predictive algorithm.
3. In general, execution of more CREATE, EDIT or DELETE activities does not lead to a better conceptual model. Nevertheless, for behavioral aspects the low-scoring students execute more EDIT and DELETE activities, probably due to the fact of struggling with complex parts of the model.
4. Better students tended to save their model much more frequently than worse-scoring students did.
5. High-scoring students tended to respond to feedback with validating model activities, while low-scoring students often perform creating or customizing activities instead.
6. The scores of intermediate assignments are indicative of the final score.

It is interesting to see that pattern 1 indicates that the pattern observed in group work for complete models [16] also holds at the level of part-tasks. Despite the positive results, there are certain limitations to the study. One of the limitations is the limited sample size. Since the assignments were not graded, not all the students made them, and some of the students might not have put a sufficient effort into making the tasks. This could mean that some of the observed behavior might not fully represent the modeling ability of the person. Furthermore, collecting data across academic years induces the limitation that the conditions under which the tasks have been performed as well as their grading may be subject to slight variations. Yet at the same time, the research clearly shows that findings from a single year cannot be easily generalized: the pattern of worse students creating and deleting substantially more than better students in 2017 for HW2 is not fully present for students in 2018 for the same homework. The collection of data in consecutive years thus allows to identify persistent patterns that are more likely to be generalizable. Finally, working with the JMermaid tool has certain limitations as well. For example, some of the log files have been lost or corrupted because some students used the old version of the tool.

6 Conclusion and Future Work

Creating conceptual models is a challenging task to acquire, especially for novices, due to its ‘ill-structured’ nature. Building better models requires not only a better knowledge background, but also a certain order of actions in which such model is created. Given this, in this study we employed a process-oriented view on modeling to explore potential behavioral patterns and indicative features correlated with better learning outcomes. We exploit process mining, as well as descriptive statistics and activity counts, and show behavioral patterns that occur for the students with different performance in the assignments. These patterns are listed in previous sections; most importantly, we show that they exist and could be implemented as features in a predictive algorithm. As such, potential problems in the performance of the students can be spotted in advance, providing an opportunity to help those students and provide them with needed feedback in an automated way [20]. Another important finding is that problems in the intermediate assignments are indicative of the performance in the whole course. Thus, predicting these problems as early as possible can help teachers to support the students and change their final outcomes to better ones.

The main contributions of this work was to provide an empirical approach for studying learners behavior by applying process mining techniques. The goal is to find features that are predictive for better or worse outcome, so that students can be given process-oriented feedback while modeling, rather than only outcome feedback. Further research needs to deepen the current results by repeating the analysis for similar task, in order to confirm the detected patterns. Furthermore, these first results can already be used to expand the tool’s current feedback functionalities. These implemented features can then be used in the future to study the students’ reaction to process-oriented feedback.

References

1. Van der Aalst, W.M.: Process Mining: Data Science in Action. Springer, Heidelberg (2016)
2. Embley, D.W., Thalheim, B.: Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges. Springer, Heidelberg (2012)
3. Wand, Y., Monarchi, D.E., Parsons, J., Woo, C.C.: Theoretical foundations for conceptual modelling in information systems development. Decis. Support Syst. **15**(4), 285–304 (1995)
4. Schenk, K.D., Vitalari, N.P., Davis, K.S.: Differences between novice and expert systems analysts: what do we know and what do we do? J. Manag. Inf. Syst. **15**(1), 9–50 (1998)
5. Nelson, H.J., Poels, G., Genero, M., Piattini, M.: A conceptual modeling quality framework. Softw. Qual. J. **20**(1), 201–228 (2012)
6. Van Merriënboer, J.J., Kirschner, P.A.: Ten Steps to Complex Learning: A Systematic Approach to Four-Component Instructional Design. Routledge (2017)
7. Wang, W., Brooks, R.J.: Empirical investigations of conceptual modeling and the modeling process. In: Proceedings of the 39th Conference on Winter Simulation: 40 Years! The Best is Yet to Come, pp. 762–770. IEEE Press (2007)
8. Shanks, G., Tansley, E., Weber, R.: Using ontology to validate conceptual models. Commun. ACM **46**(10), 85–89 (2003)
9. Weerapong, S., Porouhan, P., Premchaiswadi, W.: Process mining using α-algorithm as a tool (a case study of student registration). In: 2012 10th International Conference on ICT and Knowledge Engineering (ICT & Knowledge Engineering), pp. 213–220. IEEE (2012)
10. Juhaňák, L., Zounek, J., Rohlíková, L.: Using process mining to analyze students' quiz-taking behavior patterns in a learning management system. Comput. Hum. Behav. (2017)
11. van der Aalst, W.M.P., Guo, S., Gorissen, P.: Comparative process mining in education: an approach based on process cubes. In: Ceravolo, P., Accorsi, R., Cudre-Mauroux, P. (eds.) SIMPDA 2013. LNBIP, vol. 203, pp. 110–134. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46436-6_6
12. Papamitsiou, Z., Economides, A.A.: Process mining of interactions during computer-based testing for detecting and modelling guessing behavior. In: Zaphiris, P., Ioannou, A. (eds.) LCT 2016. LNCS, vol. 9753, pp. 437–449. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39483-1_40
13. Pinggera, J., et al.: Styles in business process modeling: an exploration and a model. Softw. Syst. Model. **14**(3), 1055–1080 (2015)
14. Claes, J., Vanderfeesten, I., Pinggera, J., Reijers, H.A., Weber, B., Poels, G.: A visual analysis of the process of process modeling. Inf. Syst. e-Bus. Manag. **13**(1), 147–190 (2015)
15. 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. Comput. Hum. Behav. **41**, 486–503 (2014)
16. 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. Comput. Hum. Behav. **57**, 352–376 (2016)
17. Snoeck, M.: Enterprise Information Systems Engineering: The MERODE Approach. Springer Publishing Company, Incorporated, Switzerland (2014)

18. Haesen, R., Snoeck, M., Lemahieu, W., Poelmans, S.: Existence dependency-based domain modeling for improving stateless process enactment. In: 2009 World Conference on Services-I, pp. 515–521. IEEE (2009)
19. Snoeck, M., Dedene, G.: Existence dependency: the key to semantic integrity between structural and behavioral aspects of object types. IEEE Trans. Softw. Eng. **24**(4), 233–251 (1998)
20. Serral, E., De Weerdt, J., Sedrakyan, G., Snoeck, M.: Automating immediate and personalized feedback taking conceptual modelling education to a next level. In: 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS), pp. 1–6. IEEE (2016)