Domain-Driven Event Abstraction Framework for Learning Dynamics in MOOCs Sessions*

Luciano Hidalgo[1] and Jorge Munoz-Gama[1]

Department of Computer Science, Pontificia Universidad Católica de Chile
{lhidalgo1,jmun}@uc.cl

Abstract. In conjunction with the rapid expansion of Massive Open Online Courses (MOOCs), academic interest has grown in the analysis of MOOC student study sessions. Education researchers have increasingly regarded process mining as a promising tool with which to answer simple questions, including the order in which resources are completed. However, its application to more complex questions about learning dynamics remains a challenge. For example, do MOOC students genuinely study from a resource or merely skim content to understand what will come next? One common practice is to use the resources directly as activities, resulting in spaghetti process models that subsequently undergo filtering. However, this leads to over-simplified and difficult-to-interpret conclusions. Consequently , an event abstraction becomes necessary, whereby low-level events are combined with high-level activities. A wide range of event abstraction techniques has been presented in process mining literature, primarily in relation to data-driven bottom-up strategies, where patterns are discovered from the data and later mapped to education concepts. Accordingly, this paper proposes a domain-driven top-down framework that allows educators who are less familiar with data and process analytics to more easily search for a set of predefined high-level concepts from their own MOOC data. The framework outlined herein has been successfully tested in a Coursera MOOC, with the objective of understanding the in-session behavioral dynamics of learners who successfully complete their respective courses.

Keywords: Event Abstraction · MOOC · Learning Dynamics

1 Introduction

The use of technology in educational environments has increased the learning alternatives around the world. In this regard, Massive Open Online Courses (MOOCs) are one of the most popular alternatives, since they enable learners to operate through a completely online environment, across a variety of subjects, scaling seamlessly across hundreds or thousands of users [4]. These courses were originally conceived of as opportunities for personal capacity building and have

* This work is partially supported by ANID FONDECYT 1220202, IDeA I+D 2210048 and ANID-Subdirección de Capital Humano/Doctorado Nacional/2022-21220979.

now been integrated into the curricula of numerous educational institutions, in which the line between face-to-face and online learning has become increasingly blurred. This integration has led to an increase in the understanding of how users carry out their tasks and perform on these platforms, and this in itself has become both a topic of interest for all stakeholders and an open area of research [2, 4]. In particular, there is a growing interest in understanding learner dynamics within a session, i.e., during an uninterrupted period of work [2, 4].

Educational managers have considered process mining a promising tool with which to answer their research questions, given its ease of use for users who are not necessarily experts in data and process analytics [14]. A common approach in the literature consists of using fields directly from a database table as activities for process mining algorithms, e.g., the accessed MOOC resource [14]. The conclusions that can be drawn from this approach are limited. Given the number of possible activities and variants, the result may end up as a spaghetti process model. In such cases, a majority of authors opt to heavily filter the number of activities or arcs to achieve a readable albeit partial model and to limit the complexity of the questions that can be answered.

More complex questions necessarily require event abstractions, i.e., low-level events are combined in high-level patterns, creating logs that are better tailored to answering such questions and with less variability, thus improving interpretability. In the literature on process mining, there is a broad variety of event abstraction methods (for a literature review on the topic, readers should see [15]). Most event abstraction approaches are data-driven (bottom up), i.e., domain-agnostic and unsupervised methods to detect frequent patterns in data. In certain cases these frequent patterns are mapped according to the most fitting education concepts, e.g., self-regulated learning profiles. However, the application of these techniques, although possible, is difficult when there is a set of high-level activities that have already been defined and an attempt is made to determine such behaviors in the log in a domain-driven (top-down) manner. For example, in the case of learning dynamics, the same pattern of accessing a MOOC resource may reflect whether the learner is studying from a resource, or simply skimming over it to understand what will come next. Finally, defining an event abstraction can be a highly complex task for educational decision-makers who are not experts in process mining, since it requires a solid understanding of concepts such as case ID, activity ID and event ID. That is why it is necessary to define frameworks (or easy-to-follow recipes) in interdisciplinary scenarios, such as education, in order to apply process mining.

This paper proposes a domain-driven event abstraction framework specifically to analyze learning dynamics in MOOC sessions. The framework is simple enough to be replicated by educational managers and defines the following: 1) a minimal data model that can be adapted to most platforms (Coursera, FutureLearn, EdX); 2) the definition of a low-level event log, including the definition of case ID and session; and 3) the definition of seven high-level learning dynamics and their corresponding high-level log. In addition, this paper validates and illustrates the application of the framework by means of a case study: to deter-

mine the learning dynamics of the sessions of students who successfully complete the MOOC "Introduction to Programming in Python" on the Coursera platform. The remainder of the paper is structured as follows: Section 2 describes the most relevant research undertaken in the area; Section 3 presents the framework and its three core elements; Section 4 illustrates the application of the framework in the selected case study in order to validate the feasibility thereof; and Section 5 concludes the paper and outlines potential future work.

2 Related Work

Process Mining and MOOCs: Although MOOC systems generate a significant amount of data, their research using process mining techniques is just starting [14]. However, several authors have attempted to describe or explore student processes from this data. For example, [12] investigate the differences in the process between three different sets of students depending on whether they have completed all, some or none of the MOOC activities. On the other hand, by combining clustering techniques with process mining, [3] identify four sets of students, ranging from those who drop out at the very beginning of the course to those who successfully complete it. Their research shows how students who composed the cluster of individuals who successfully completed the course tended to watch videos in successive batches. In one of the most relevant works in this subject, [9] study the event logs of three MOOC Coursera courses and discovered six patterns of interaction among students. These patterns were also grouped into three clusters, identified as sampling learners, comprehensive learners and targeting learners, according to the behavior described. Furthermore, this work has incorporated the concept of "session" as a unit of analysis. [4] explore in greater depth the behavior of students in work sessions according to eight different possible interactions, segmenting them according to those who complete and those who do not complete the course. The aforementioned paper finds that students who complete the course are those who show more dedicated behavior and carry out a greater number of sessions.
Process Mining and Event Abstraction: Despite the utility of process mining techniques for understanding how organizations function, the systems that generate this data are not necessarily capable of handling the appropriate level of detail. Therefore, techniques that allow the abstraction of high-level activities from granular data are vital for the correct application of process mining techniques [15]. Currently, there are several strategies with which to address this problem. One family of techniques uses unsupervised machine learning by grouping events according to different dimensions, such as: the semantics of activity names [13], the physical proximity in which events occur [11], events that occur frequently together [10], and sub-sequences of activities that are repeated [5], among others. Additional authors have proposed less automated strategies, such as [7], who group elements according to the relationships between entities (ontologies) in order to abstract events using domain knowledge. This latter research was successfully applied in the medical domain. Similarly, [1] proposes

a four-stage method based on the prior identification of process activities, a granular matching between activities and events according to their type, and certain context-sensitive rules. Indeed, this method proposes the grouping of different events into activities. Furthermore, in a combination of supervised and unsupervised methods, [8] propose a method for event abstraction in diffuse environments. As such, the aforementioned approach is based on the separation of events into sessions according to activity periods, prior to the generation of clusters of events, which are manually reviewed in a heat map in order to subsequently map them to high-level activities.

3 Domain-Driven Event Abstraction Framework

This section presents the domain-driven event abstraction framework to aid educational managers in building a high-level event log with which to analyze the learning dynamics of students during their MOOC work sessions. The framework is composed of three stages: 1) a minimal data model capable of being mapped to any MOOC system; 2) the definition of a low-level log from the minimal data model; and 3) the definition of a high-level log derived from the low-level log.

S1: Minimal Data Model The first stage that defines the framework is the minimal data model. This is a data model with the minimum information necessary to build the low-level event log, which serves as a *lingua franca* among different MOOC systems, including Coursera, FutureLearn and edX, among others. Figure 1 shows the minimal data model, which is filled with information each time a user interacts with a MOOC resource. The model contemplates the identification of three main elements: the resource interacted with, the user who performs the interaction, and the time at which the interaction is made. Resources and users are identified with a unique identifier, present in all MOOC systems. In addition, the model determines that each resource adheres to an associated order within the MOOC. Utilizing this approach, it can be determined whether the user is interacting in a sequential or disorderly manner via the resources. The model also defines the type of resource in question. In this proposal, two generic types are defined: content resources (video-lectures, presentations, etc.) and assessments (quizzes, exams, etc.). However, the framework can be easily extended to include other types of resources, such as project or bibliographical resources. Finally, each interaction with a resource has an associated state (start or complete) in the event log. This makes it possible to identify whether the learning dynamics of students correspond to exploratory or in-depth work patterns. The majority of MOOC systems contain the necessary information to be able to determine status. In some cases, such as Coursera, state is explicitly recorded as two different interactions (one is "Started" and the other is "Completed") in its Course Progress table. In other systems, status can be determined from two timestamps ("Start" and "End") that are associated with the same interaction.
S2:Low-level Log The second stage of the framework describes how to build the low-level log, based on the information contained in the minimal data model, as defined in the previous stage. Each interaction with a MOOC resource recorded

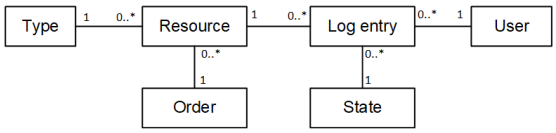

Fig. 1: Minimal data model sugested.

in the minimal data model represents an event in the low-level log. The transformation of the information in the minimal data model to the low-level log is straightforward, with the exception of two elements: the session and case ID.

This framework is designed to analyze the learning dynamics in student work sessions, i.e., an uninterrupted period of work. Therefore, it is necessary to define to which session each interaction with a resource pertains. Certain MOOC systems have their own built-in session definition and identification. However, in most MOOC systems this definition is not explicitly available, although it can be determined. For example, two consecutive interactions pertain to different sessions if, between their timestamps, a certain threshold of time has passed in which no interaction with the MOOC has been carried out. Different thresholds and the implications thereof have been reviewed in the literature [6]. Once the session has been determined, the framework defines the case ID of the low-level log as the pair (user ID, session ID), i.e., different sessions of the same student correspond to different cases in the log.

S3: High-level Log The low-level event log obtained in stage two resembles the analysis input that a non-expert user in process mining would normally use directly in a tool such as Disco or ProM. However, the large amount of resources and variants that result from this type of log make it difficult to obtain process-driven answers. Therefore, stage three of the framework defines seven high-level activities, with each one representing a different learning dynamic which reflects learner behavior, regardless of the resources consulted. In particular, this includes four dynamics associated with content consumption and three related to interaction with assessments.

- *Progressing*: this represents the learning dynamic of a student who consumes a resource and then continues, in the correct order, with the next resource in the course.
- *Exploring*: this represents the learning dynamic of a student who interacts in a superficial manner with new content, simply in order to know what to expect, for example, to determine the time needed to consume that content.
- *Echoing*: this represents the learning dynamic of a student who consumes a resource, and then continues on to the next resource in the correct order, but with resources that have already been previously completed. A good example is a learner who decides to review content prior to sitting an exam.
- *Fetching*: this represents the learning dynamic of a student who interacts with a previously completed resource, with or without completing it, and in no particular order. A good example is a student who, after failing an

assessment question, re-watches (partially or totally) a specific video in order to identify the answer.
- *Assessing*: this represents the learning dynamic of a student who interacts with and completes an assessment-type resource that has not been previously completed. In the case of a block of several Assessing dynamics in a row, regardless of their order, these are collapsed into a single dynamic.
- *Skimming*: this represents the learning dynamic of a student who initiates but does not complete interactions with assessments. For example, the student could be reviewing the questions before taking an assessment seriously or could be reviewing an assessment beforehand in order to understand where he/she went wrong.
- *Retaking*: this represents the learning dynamic of a student who initiates and completes assessments that have been previously completed. For example, a user who did not obtain a satisfactory score and who decides to retry in order to improve their previous result.

Figure 2 summarizes how each low-level event is associated with a learning dynamic, in terms of a decision tree. It should be noted that low-level consecutive events associated with the same learning dynamic are consolidated within the same activity and the checking if an resource was completed before is measured across all sessions. To determine the duration of the activity, the first timestamp in the sequence is taken as the start of the activity and the final one is taken as the end. This generates a new event log with a significant reduction of activities.

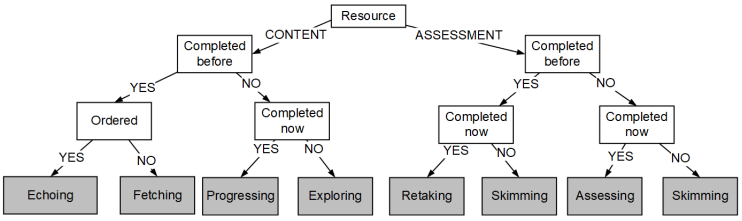

Fig. 2: Criteria to assign to each activity.

4 Case Study: Successful Student Sessions in Coursera

To illustrate the application of the framework and validate its applicability with real data, a case study was conducted using data from the "Introduction to Programming in Python" course on the Coursera platform. The objective of the case study was to examine the learning dynamics that took place in the sessions of students who successfully completed the course. Specifically, the following two questions are defined: *RQ1: What are the characteristics of the sessions that involve learning dynamics in which a resource is revisited?* and *RQ2:Are there differences in terms of learning dynamics between the first sessions and the final sessions carried out by students?* With that in mind, this section presents the following: first, the descriptive information of the course and the application

of the three levels of the framework related to the case study; second, the results-based answers of the two research questions; and third, a brief discussion of the implications of these results.

Case Study & Framework: This study considers data generated from a Coursera course held during the period June 23, 2017 to April 14, 2018. The course involved a total time commitment of 17 hours and was organized into 6 modules, 1 for each week. In this analysis, consideration was taken of 58 possible resources with which to interact, 35 content resources (video lectures) and 23 assessment resources.

The first step in the application of the framework was to align the minimal data model available to Coursera with the minimal data model proposed herein. The Coursera data model contained more than 75 tables. The most relevant table for this study was the Course Progress table, which recorded the course ID, the resource interacted with, the user who performed the interaction, the status (start/complete) and the timestamp detailing when it occurs. However, as this table only contained IDs, it was necessary to supplement it with the course information tables (Course Item Types, Course Items, Course Lessons, Course Modules, Course Progress State Types) so as to establish the order of the resources within the course and obtain descriptive information.

The second step in the application of the framework was to build the low-level event log, including the concept of session ID. To do so, an activity was considered to occur in the same session as the previous activity if, between them, there was a lapse equal to or less than 60 minutes, which is the maximum limit for time-on-task, as established by [6]. Hence, the case ID for the low-level event log was established as the pair (user ID, session ID). Only users who began and completed the course during the observation period were used for this analysis. The criterion to determine whether a user completed the course was based on whether that learner completed either of the final two course assessments. This yielded 209 user cases for analysis and a total of 320,421 low-level events. Finally, Coursera recorded progress through each question within the assessment as each new event started, e.g., a student completing an assessment with 10 questions results in 11 events started and 1 completed. This duplication was subsequently condensed, resulting in a low-level event log of 39,650 events.

The final step in the application of the framework was the creation of the high-level event log from the definition of the seven high-level activities: *Progressing, Exploring, Echoing, Fetching, Assessing, Skimming and Retaking*. The resulting high-level log contained 18,029 events. This represented a 54.5% reduction of activities compared to the low-level log. As with the low-level log, the case ID for the high-level event log was established as the pair (user ID, session ID). From the 209 users, this generated 7,087 cases which were grouped into 1,237 distinct variants.

RQ1: This study detected differences between the various sessions that involved learning dynamics in which a resource was revisited, i.e., Echoing, Fetching, and Retaking. An exploratory analysis of the sessions showed that Fetching appeared in 13% of cases, and the interaction of this activity seems to be strongly related

to assessment dynamics (i.e., Assessing, Skimming, Retaking); in 54.3% of cases in which this activity was detected, its occurrence was preceded by one of the activities related to assessment; and in 50.9% of cases, Fetching was followed by some form of assessment.

In this case, the Fetching of a content (totally or partially) suggests a specific search-related action, either in preparation for an assessment or in response to a certain element that appeared in an assessment and about which it is worth clarifying a particular doubt. However, when consideration is taken of the sessions that included an Assessing or Retaking activity besides Fetching, the proportion changed, with 25.6% performing the fetch prior to the assessment and 21.5% afterwards. Analysis of the content associated with Fetching showed that the most commonly fetched resources were *2.2.2 Input, 3.1.1 If/Else , 3.2.2 For, 2.1.1. Data Types* (which can be understood as the first different elements for someone with no prior programming knowledge), and *6.1.4 List Functions* (following analysis, it could be seen that this particular content was poorly designed and suggestions were made to re-record the video using a new structure).

When comparing with cases in which Echoing appeared (as shown in Figure 3), behavior was seen to have changed, since in the majority of cases this activity was directly related to Progressing, to the extent that in 35.9% of the cases with Echoing, the previous activity or the one that immediately succeeded it was Progressing. This indicates that the extensive repetition of content occurred in sessions in which the student was oriented towards studying content and that during these study sessions doubts arose, which therefore necessitates an in-depth review of previously seen content. This differs to the patterns generated with regards to Fetching, which appears more strongly related to assessment activities. Nonetheless, in this case a relationship also existed with the assessment activities. Yet, they differed in the sequence point in which they appeared, since a repetition of content occurred more frequently prior to the assessment, as opposed to the variants that included Fetching, whereby the content review occurred more frequently after the assessment was accessed.

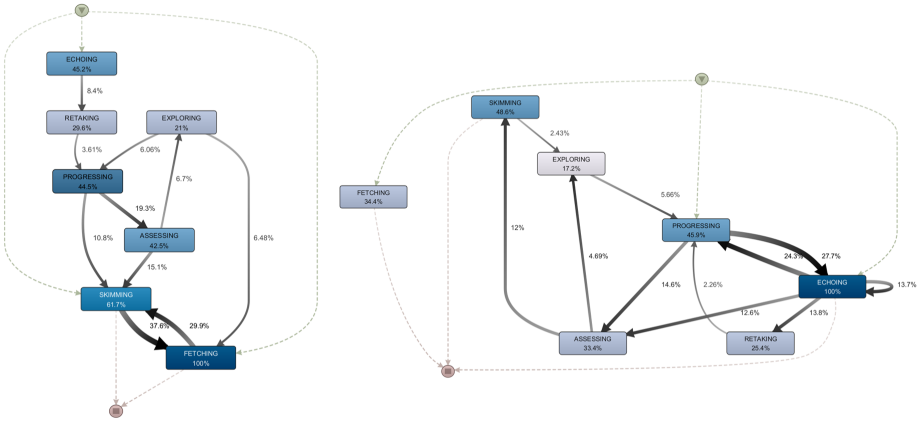

Fig. 3: Comparison between sessions fetching (left) and echoing (right) sessions.

Finally, when reviewing Retaking (Figure 4) it can be assumed that the user entered the session directly with the intention of retaking assessments, since the most common variant (25.9% of cases) only repeated their assessments and then concluded the session. Similarly, the activity with which there was the strongest relationship in this case is Skimming, which indicates a dynamic whereby learners performed a self-evaluation and then reviewed the results, or looked at their previous results which they then attempted to improve. The transition from Retaking to Skimming occurred in 33.6% of cases in which repetition was present, while the reverse occurred in 32.1% of the cases. This implies that either (or both) of these interactions appeared in 43.0% of the cases with Retaking. By reviewing the most commonly repeated Retaking-related assessments, one assessment in particular was noted as having a significantly higher number of Retaking than the rest (597 occurrences out of an average of 285). In consultation with the course designers, this assessment, which measured the topics of variables and input/output, was found to have had a bug in one of the questions. The bug was subsequently corrected after the observation date had been recorded.

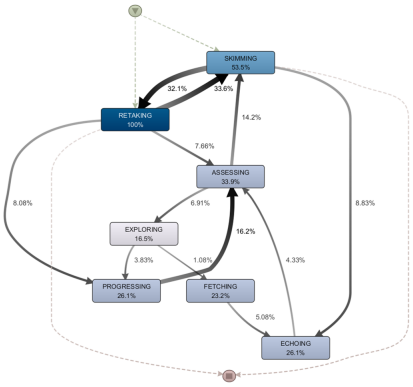

Fig. 4: Learning dynamics in sessions with retaking.

RQ2: The sessions of each student were divided into quintiles by considering the total number of sessions completed by each one. Thus, the sessions of the first and last quintile were compared. This made it possible to verify the existence of differences between behavior at the beginning and end of the course.

In the process model obtained from the analysis of the initial sessions (Figure 5) it can be observed that the most common activities were associated with orderly and comprehensive learning (Progressing 63.9% and Echoing 29.7%). Furthermore, a relatively low commitment to assessment can also be seen at this point in the course, since although the Skimming activity appeared in 36.7% of cases, students were observed undertaking sessions without completing an assessment in 66.3% of cases. This idea is reinforced by the observation that in 5.78% of cases, the Progressing activity involved more than one piece of content being completed in the correct order. This suggests that students preferred not to interrupt their content study progression in order to carry out the interspersed assessments. As the Progressing and Exploring activities refer to the

very first time a piece of content was viewed, these activities were expected to be more frequent at the beginning of the course, showing a decreasing frequency towards the end of the course. However, it is noteworthy that the Echoing activity experienced a high frequency of 29.7% during the initial sessions.

Conversely, by grouping the sessions into quintiles it was possible to evince that sessions at the beginning of the course tended to experience the most changes in terms of learning dynamics. Indeed, despite comprising 19% of the sessions, 23% of all events in the high-level log were found to take place in these initial stages. By conducting the same exercise with each quintile in turn, it can be seen that the number of events grouped together in each one decreased, reaching a mere 15% of events in the final quintile.

It seems that with regards to the final sessions (Fig. 5) these were mainly carried out in relation to assessment activities, since all associated activities (Skimming, Assessing and Retaking) appeared more frequently than those associated with contents. For example, 39.9% of the former performed at least one Assessing or Retaking activity. However, it is striking to find that 40.4% of cases corresponded to students who only undertook Skimming and then finished the session, thus suggesting that a significant number of learners simply logged on to browse the questions without completing the broader assessment. Regarding the dynamics of the content activities, the Progressing activity tended to be the one that initiated the sessions in which it appeared, and was most frequently succeeded by assessment activities, particularly Assessing. This indicates changes from the beginning of the course, whereby the user tended to either continue to study or repeat content more frequently. In addition, from this perspective it should be noted that even at the end of the course the Progressing activity appeared more frequently than Echoing and with a higher average duration (23.5 minutes versus 9.6 minutes, on average). Similarly, the behavior of continuing to study by skipping an assessment drastically reduced its occurrence, accounting for merely 2 cases or 0.1% of these sessions.

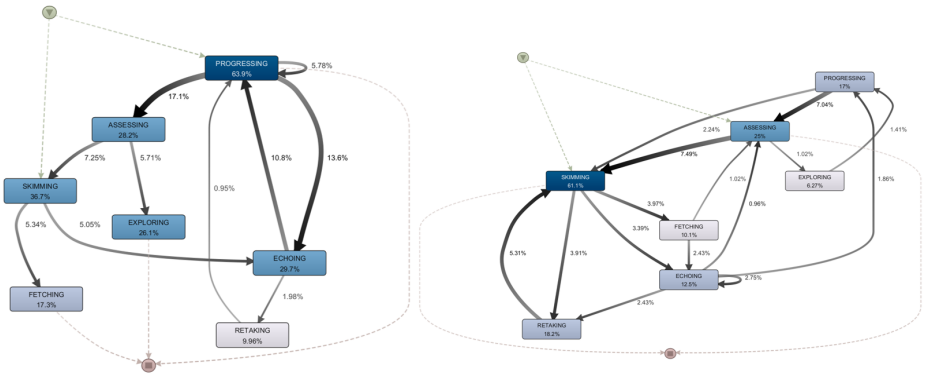

Fig. 5: Difference between initial sessions and ending sessions (30% paths).

Discussion: First, the results show that students varied their session behavior during the duration of the MOOC, given that at the beginning of the course they

were more reluctant to assess themselves and preferred to review content rather that to measure the extent of their overall knowledge. The situation changed as they progressed through the course, as 25% of the total number of events recorded in the log ended up as completed assessments, either for the first time or by repeating an already-completed assessment. This indicates that student commitment to the course increased as they progress through it. By examining the detail of the cases, it was found that the most common variants were of a single activity and that the longest sequences commonly involved activities of the same type (e.g., Progressing and Echoing or Skimming and Retaking). This confirms the findings of [2] who suggest that successful students change the priority of the activities they complete between course sessions. On the other hand, the patterns observed are consistent with experiments that use other techniques or optics on MOOC data, such as machine learning or clustering.

One of the unexpected results in this research was the discovery that the sole, most common variant was the Skimming activity, which accounted for 26.7% of the high-level log. This could point to the need to refine the skimming activity, since the review of an unfinished assessment may be the result of several possible factors, including: that the difficulty of the content due to be assessed is being reviewed in preparation for a serious attempt to complete it; that mistakes made in previous attempts are being reviewed; and that the questions are being used as learning examples, among others.

5 Conclusions

This paper presents a domain-driven event abstraction framework that facilitates the construction of a high-level event log with which to analyze learning dynamics in MOOC sessions. Specifically, the framework is composed of three stages: 1) the minimal data model necessary; 2) the construction of a low-level event log; and 3) the definition of seven high-level activities that can be used to build the high-level event log: Progressing, Exploring, Echoing, Fetching, Assessing, Skimming, Retaking. The application of the framework in a real scenario was validated in a case study in which the learning dynamics of the sessions of students who successfully completed the course were analyzed. Specifically, analysis was undertaken of the behavior in the sessions in which a resource was reviewed and an error found in the course, in addition to the differences in behavior between the first and last sessions of the students.

This research should be considered as exploratory and preliminary in nature, with significant room for improvement in future work. First, the framework attempts to extrapolate the intentions of the students (e.g., progressing vs exploring) from the available data. Nevertheless, such extrapolations could be refined if the framework were complemented with certain additional instruments, for example, surveys and interviews, as has been carried out in other types of MOOC analysis, such as self-regulated learning [2]. Second, domain-driven event abstractions and data-driven event abstractions should not be considered as opposing techniques, but rather as two sides of the same coin that can complement one an-

other. In this regard, rather than a purely domain-driven event abstraction, this investigation could be complemented by one of the data-driven event abstraction techniques outlined in [15], thus creating a hybrid method that combines the two approaches in an iterative manner. Third, it is crucial to test the framework in different courses and MOOCs to ensure its generality and usefulness.

References

1. Baier, T., Mendling, J., Weske, M.: Bridging abstraction layers in process mining. Information Systems **46**, 123–139 (2014)
2. de Barba, P.G., Malekian, D., Oliveira, E.A., Bailey, J., Ryan, T., Kennedy, G.: The importance and meaning of session behaviour in a mooc. Computers & Education **146**, 103772 (2020)
3. Van den Beemt, A., Buijs, J., Van der Aalst, W.: Analysing structured learning behaviour in massive open online courses (moocs): an approach based on process mining and clustering. International Review of Research in Open and Distributed Learning **19**(5) (2018)
4. Bernal, F., Maldonado-Mahauad, J., Villalba-Condori, K., Zúñiga-Prieto, M., Veintimilla-Reyes, J., Mejía, M.: Analyzing students' behavior in a mooc course: A process-oriented approach. In: HCI. pp. 307–325. Springer (2020)
5. Günther, C.W., Rozinat, A., Van Der Aalst, W.M.: Activity mining by global trace segmentation. In: Business Process Management. pp. 128–139. Springer (2009)
6. Kovanović, V., Gašević, D., Dawson, S., Joksimović, S., Baker, R.S., Hatala, M.: Penetrating the black box of time-on-task estimation. In: Proceedings of the fifth international conference on learning analytics and knowledge. pp. 184–193 (2015)
7. Leonardi, G., Striani, M., Quaglini, S., Cavallini, A., Montani, S.: Towards semantic process mining through knowledge-based trace abstraction. In: Int. Symposium on Data-Driven Process Discovery and Analysis. pp. 45–64. Springer (2017)
8. de Leoni, M., Dündar, S.: Event-log abstraction using batch session identification and clustering. In: ACM Symposium on Applied Computing. pp. 36–44 (2020)
9. 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**, 179–196 (2018)
10. Mannhardt, F., Tax, N.: Unsupervised event abstraction using pattern abstraction and local process models. arXiv preprint arXiv:1704.03520 (2017)
11. Rehse, J.R., Fettke, P.: Clustering business process activities for identifying reference model components. In: BPM. pp. 5–17. Springer (2018)
12. Rizvi, S., Rienties, B., Rogaten, J., Kizilcec, R.F.: Investigating variation in learning processes in a futurelearn mooc. J. Computing in Higher Education **32**(1), 162–181 (2020)
13. Sánchez-Charles, D., Carmona, J., Muntés-Mulero, V., Solé, M.: Reducing event variability in logs by clustering of word embeddings. In: BPM. pp. 191–203. Springer (2017)
14. Wambsganss, T., Schmitt, A., Mahnig, T., Ott, A., Soellner, S., Ngo, N.A., Geyer-Klingeberg, J., Naklada, J.: The potential of technology-mediated learning processes: a taxonomy and research agenda for educational process mining. ICIS (2021)
15. van Zelst, S.J., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining: literature review and taxonomy. Granular Computing **6**(3), 719–736 (2021)