Educational Process Mining for Verifying Student Learning Paths in an Introductory Programming Course

Eduardo Machado Real*†, Edson Pinheiro Pimentel*, Lucas Vieira de Oliveira*,
Juliana Cristina Braga* and Itana Stiubiener*
*Federal University of ABC (UFABC)
Santo André (SP), Brazil
(eduardo.real, edson.pimentel, l.vieira, juliana.braga, itana.stiubiener)@ufabc.edu.br
†State University of MS (UEMS)
Nova Andradina (MS), Brazil
eduardomreal@uems.br

***Abstract*—This full paper of the research-to-practice category addresses the problem of organizing instructional materials and assessment activities in e-learning courses and their effects on learning outcomes. Usually, the teacher organizes the course sequence according to his didactic-pedagogical strategies and expects this help to guide the student through his learning process in the course. However, unless restrictions are imposed, students may choose to follow different paths than those indicated in the material's organization. A question emerges from this context: what are the impacts on the students learning outcomes when they take learning paths other than expected by the teacher? In Virtual Learning Environments, student's interaction with course materials can be stored in the so-called event logs. With the support of Educational Process Mining, it is possible to track the path of how and what specific actions students perform during learning, resulting in process models and historical statistical information. This paper aims to present the application results of PM techniques to verify the students learning paths in an introductory programming course. We used a Moodle event log containing 24605 events collected from 73 undergraduate students. For experiments, we divided this original log file into five other segments of datasets among passed and failed students variations. Techniques to obtain statistical information, Heuristic Miner algorithm to process discovery, and other techniques were applied from the implementations available in ProM Framework and scripts based on PM4Py library. The results showed that overall approved and failed students took different paths and event numbers to perform activities in the course. Besides, we obtained control-flows and frequencies of the activities and connections, thus making it possible to identify the dependencies, which resources started or ended the process, among other things. The analysis of these results provides general and specific information on students' learning paths and can help teachers observe students' behavior patterns and progress.**

***Index Terms*—Process Mining, E-learning, Learning Paths.**

I. INTRODUCTION

The use of e-learning has increasingly been encouraged and disseminated, not only for distance learning courses but for presential and blended-learning courses as well. According to [1], e-learning is a learning process that makes use of information and communication technologies in order to create different courses, distribute learning content, allows communication between students and educators and, of course, manage learning. Those who defend the usage of e-learning mention as main benefits: the possibility of learning anywhere and anytime; the possibility of studies personalization and feedback; the encouragement and respect of the student autonomy in the self-regulation of their learning processes.

Several studies have been showing data that describes the growth of e-learning. For example, in a recent study about open and distance courses in the world [2], shows that the enrollments in distance learning have been growing year after year and have grown faster in emergent economies.

The Learning Management Systems (LMS) have been the main type of software adopted in e-learning and among then stand out Moodle, BlackBoard, EDx, and others. The LMS are composed of several tools that allow the organization of content and assessment activities and also provide synchronous and asynchronous communication between course participants. Several works have studied the characteristics of these LMS in relation to student and teacher engagement [3] and also satisfaction in relation to e-learning [4].

In addition to the LMS functionalities, an important aspect of e-learning is the design of the courses, that is, the way in which the contents and activities are organized. Usually, the teacher organizes the course sequence (content, activities, communication) according to his didactic-pedagogical strategies to guide the student in his learning process in the course. However, unless restrictions are imposed, the LMS allow students to choose to follow different paths than those indicated in the course organization. A question arises from this context: what are the impacts on student's learning outcomes when they follow different learning paths than those expected by the teacher?

In the LMS, the interaction of the student with the course materials can be stored in the so-called event logs. These log data have the potential to reveal in a complete way (start-to-end) the student's behavior based on how the contents and

activities were accessed and performed by him, i.e., the steps taken during the studies in the e-learning. This revelation can be achieved with the usage of Process Mining (PM) techniques whose basic idea is to extract knowledge from the event logs recorded by an information system [5] [6] [7]. Thus, it seeks the confrontation between such event logs (i.e., observed behavior) and process models (elaborated by a specialist or automatically discovered). In the field of Education, the application of PM is conceptualized as Educational Process Mining (EPM).

The objective of this paper is to present the results of the application of PM techniques to verify students learning paths in an Introduction to Programming course. The course was conducted in the Moodle platform, and the data was obtained from its event log. The article is organized as follows: section II describes the fundamental concepts of PM and EPM and also describes some related works; section III describes the methods used to conduct the experiments; section IV presents the obtained results and discussions; and section V sets out the final considerations.

II. BACKGROUND

This section aims to share the main concepts and formalisms necessary to understand this work. In addition, the section presents some related works.

A. Process Mining

PM aims to explore event logs in a meaningful way to provide information, identify bottlenecks, anticipate problems, recommend countermeasures, optimize processes, etc. To do so, it focuses on data that allows tracking the time and causality of activities, obtaining events that are imperceptible to specialists and are not handled by the common machine learning algorithms [5] [6] [7]. Formally, the entry to the PM is the event logs generated in the information systems. Each log record is related to an event and each event corresponds to an activity performed, in such a way that the event log must present a sequential relationship between the events. Also, there may be other additional information in the logs that can also be useful, such as date and time (start and/or end), resource (people or device that performed the activity), among others. The event logs are described and defined by some basic attributes, the first two being the minimum necessary to obtain a model, namely:

- Case – corresponds to the input elements which are defined as central analysis objects;
- Activity Name – corresponds to the activities or actions performed and that generate events in the event log;
- Timestamp – identifies the date and time of events in the event log (start and/or end records);
- Resource – identifies those who perform actions in the event log (can be a person or device, for example).

The steps of activities performed in each Case create a Trace, which is described by an ordered sequencing of events (activities), considering that each event is unique and it is related to only one activity (Activity Name). In this context, an event log is described by N Cases C (log = C_1, C_2, C_3,..., C_N), in which each one can generate a Trace T with up to M types of events E (T = E_{Type1}*, E_{Type2}*, ..., E_{TypeM}*), obviously respecting a sequence of performed events and according to the Control-Flow generated by the model. In addition, repetitions (*) of events can also occur, if there are loops or return paths. For example, see the supposed event log snippet below, in which there are N Cases and their respective Traces described by up to eight types of events E (a, b, c, d, e, f, g and h):

- Cases: 1, 2, 3, 4, 5, ..., N;
- Trace: [<a, b, c, d, e, h>, <a, d, c, e, g>, <a, c, d, e, f, b, d, e, g>, <a, d, b, e, h>, <a, c, d, e, f, d, c, e, f, c, d, e, h>, ..., <a, b, c, d, e, f, g, h>].

Note that the first Trace, which is <a, b, c, d, e, h>, corresponds to a sequence of activities performed by Case 1, characterized by a » b » c » d » e » h. The same logic can be used to represent the sequence of the other Traces.

For the application of PM, there are three main types of sets of techniques that can be used from event logs and models [5] [8]: (a) Discovery, which is a technique that has as input an event log (a log file) and produces a process model without using any prior information; (b) Conformance, in which an existing process model is compared with an event log belonging to the same process to verify whether the reality, as recorded in the log, it is in conformity with the model and vice versa; (c) Enhancement, whose main idea is to extend or improve an existing process model using information about the real process recorded in some event log.

Moreover, there are different perspectives of application that can be considered and adopted [5], such as:

- Control-flow, which focuses on the order of activities;
- Organizational, which focuses on information about hidden resources in the log;
- Case, which focuses on the properties of the cases;
- Time, which is concerned with the timing and frequency of events.

To evaluate a model obtained by Discovery algorithms, some quality criterion can be used, based on measurements corresponding to four dimensions, or forces, whose values lies between 0 and 1. They are: Fitness, Precision, Generalization and Simplicity [5] [9] [10].

- Fitness: quantifies the extent to which the discovered model can reproduce Traces from the event log;
- Precision: shows the proportion of the behavior represented by the model that is not seen in the event log, i.e., dealing with overly general models (underfitting);
- Generalization: shows how much the model will be able to reproduce the future behavior of the process and can be seen as a measure of confidence in precision;
- Simplicity: captures the complexity of a process model in terms of readability and interpretation of its structure.

B. E-learning and Educational Process Mining

The e-learning systems allow collecting records corresponding to all events, actions and activities of students at different levels of granularity, from low-level events, such as

keystrokes, gestures and mouse clicks, up to higher levels such as sequences of activities carried out, including learning paths based on principles of Self-Regulated Learning, differences in frequencies in events, among others [11] [12] [13].

EPM makes it possible to map students' behavior based on their paths when accessing content and carrying out activities in a course through an LMS. Thus, the EPM can be used by educators to better understand students learning habits, the factors that influence their performance and the skills acquired, creating and analyzing models of educational processes that represent the observed behavior [1] [14] [13] [12].

The models discovered by EPM can be used to better understand the underlying educational processes, to detect learning disabilities early, to generate recommendations for students, to assist students with specific learning disabilities, to provide feedback to students, teachers or researchers etc [15] [16]. In addition, the EPM allows examining which specific actions students have taken and check the category of actions on certain activities [17]. It is also possible to perform conformity analysis procedures, checking if a previously modeled behavior corresponds to the observed behavior [18].

The representation of the EPM on the scenario of an e-learning can be positioned as Fig. 1, in which the elements that participate in an educational domain are shown, which are capable of generating events performed by the interactions of students in the Virtual Learning Environments (VLEs). These log records create the event logs, used as input for any of the three types of PM, i.e., Discovery, Conformance and/or Enhancement (or, in this case, called Extension). Thus, the results obtained from the generated process models can be analyzed.

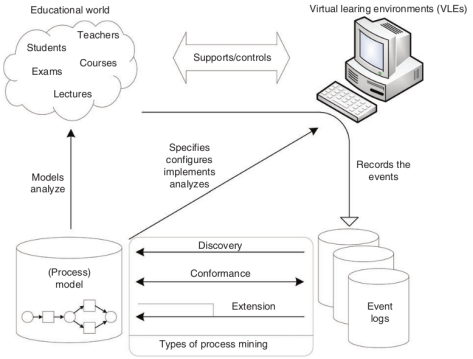

Figure 1. Types and components of PM positioned in the field of Education.

The EPM, as a method, can open up new ways of analyzing students learning behavior and problem-solving [19]. However, the implementation of PM can be considered an interdisciplinary challenge, requiring subject matter experts, item developers, psychometrists and computer scientists to work together to extract, aggregate, model and interpret the data accordingly. In this regard, the authors at [20] correctly emphasize that the comprehensibility of a model is a central objective of education due to the transfer of knowledge that this implies. The interpretability of graphics, models and visual representations by teachers and students makes the results essential for monitoring the learning process and providing feedback.

C. Related Work

This section presents and discusses some works related to this research. They are grouped according to their purposes: how to explore event logs, student behavior and results, PM integration with LMS, and design of LMS and courses.

[20] organized a tutorial showing how to explore event logs collected in the Moodle LMS and obtain process models containing more or less frequent paths and activities relative to the number of execution occurrences. In this work, the authors brought as a proposal to facilitate the interpretation of results by teachers the segmentation of an event log, given that large data sets result in complex models for a teacher or student to interpret.

In [21], the authors also used group segmentation (course weeks) of an event log from a Massive Open Online Courses (MOOCs) to explore the relationship between learning behavior and progress. The results of the group analysis indicated that students who regularly watched successive videos in batches performed better on learning outcomes. However, the results indicate that this behavior is much more related to the order in which the videos were viewed than to real-time.

[19] present a research that describes how to use event logs to analyze individual behavior in an online educational assessment and track students' skills to solve tasks. In short, this work has shown, for example, how to examine and visualize human-computer interactions based on log paths using directed graphs and also how to group student response forms to find different behavior patterns in problem-solving.

[22] describe a model to capture the temporary nature of student behavior in an Open-ended Learning Environment (OELE). These systems usually include recording mechanisms that track users' activities as temporary sequences, making it possible to analyze the relationships between their behaviors (over time) and performance. Behavior modeling can contribute to the development of early warning systems to predict students at risk while a course is in progress and to personalize e-learning environments [23].

The work described in [17], dealt with an approach to the evaluation of self-regulated learning in groups of students, on Moodle logs. As a result of the application of the EPM, this work obtained that, in general, although the approved students do not exactly follow the guidelines of the teachers (requirements), they follow the logic of a sufficient self-regulated learning process, in opposition to those who failed. The work described in [24] sought to discover and analyze the patterns of behavior performed by students with higher scores in contrast to those with lower scores.

Also applying EPM over Moodle event logs, the authors in [1] have shown that detailed analysis of student behavior is possible through statistics and individual cases of student event sequences. In this sense, patterns of variations in student behavior were obtained by identifying, for example, typical behavior of active or inactive students.

The investigation presented in [25] analyzed the student modeling task based on their operation records using sequential stock mining techniques from a statistical value of the total number of operations performed, specifically based on the order of time (Timestamp) and the correlation between actions.

The approach brought up in [14] was that the analysis of educational trajectories using PM techniques could help explain the relationship between a sequence of academic results and late dropout. These authors created path models in courses with high failure rates, aiming to understand the process that leads to the late abandonment of a given course, through the analysis of students' paths, identifying similarities and differences.

The study presented in [26] showed that using PM from Moodle data requires several stages that are not easily understood directly by teachers, requiring some form of automation. In this sense, they developed an application that successfully performed the pre-processing of the event log and that allowed displaying the results as a Control-flow analysis tool.

In order to facilitate the use of EPM by teachers, the research described in [27] showed an application to integrate the Moodle event log data with PM activities, especially to facilitate pre-processing, also having some specific functionalities in the handling of the event log, in order to help the user to obtain statistical information, as well as to carry out some exploratory data analysis.

Finally, [3] investigated the LMS structure requirements that affect user engagement, focusing on the important design factors that influence this engagement with the LMS e-learning tools.

III. Methods

The educational processes mining performed in this study used data collected from an "Introduction to Programming" course of an undergraduate course in Computing at a public university. The course was conducted at Moodle LMS for 12 weeks, in the blended-learning modality with the participation of 73 students. The students were monitored by 3 teachers and 3 tutors, carried out activities with weekly deliveries, obtaining feedback on all deliveries. In addition to watching video classes and studying the texts, each student should perform quiz-type activities and also programming activities with automatic or manual evaluation (Virtual Programming Laboratory). The assessment procedures included 3 in-person tests.

The log file obtained from Moodle, related to this class, originally had 85759 event logs described by seven attributes, namely: IP Address, Timestamp, Student Name, Event Context, Component, and Event Name. The IP Address attribute was disregarded (deleted). The Timestamp attribute contains the time when the activity was completed. The Student Name identifies the user who performed the action.

The last three attributes, together, characterize which item of the course the student accessed and what was his action on that item. Event Context contains the title of the item accessed, for example, the title of a quiz: "S01-04A-Java-Sequential Structures". Component specifies the item type, for example, Task, Quiz, VPL, or Lesson. Event Name indicates the action performed on the event, for example, viewed, edited, or uploaded. These attributes (Event Context, Component, and Event Name) make it possible to visualize a hierarchy shown in Figure 2: level 1 corresponds to the Component attribute, level 2 indicates the Event Context, and level 3 is related to the Event Name.

This hierarchy is related to a sequence of events when designing the course. First, the teacher selects the type of activity (Component), which is considered level 1. Then the teacher defines the activity with its specific content, giving it a name (Event Context), which is considered level 2. Finally, the student performs the actions in the activity (Event Name), that is, level 3.

Figure 2 shows a schematic representation of the three levels and their relationship to the concept of trace, formed by a sequence of several events carried out by the student.

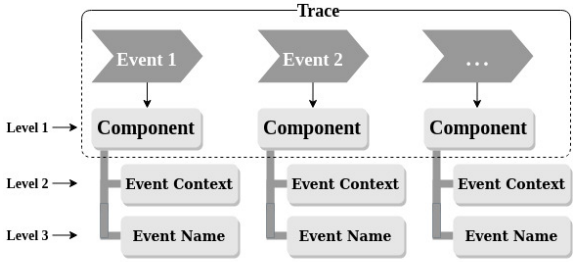

Figure 2. Attribute levels for specific analyzes.

Several occurrences that were not necessary for the study were removed, such as: exclusion of 88 from the 139 elements originally belonging to the Event Context attribute (remaining 51 elements), 9 from the 15 elements of the Component attribute (remaining 6 elements) and 33 from the 51 elements of the Event Name attribute (18 elements remaining). The order of removals started with the Component attribute, followed by Event Context and Event Name. Thus, the original event log has been reduced to 24334 events described by 6 attributes.

The Student Name was defined as the "Case" and the Component as the "Activity Name". The Timestamp attribute is used by the algorithms so that they can determine the order of events.

Inspired by other studies that used segmentation of the data set, the original event log was separated into two other data sets: one containing 44 students who passed the course and the other containing 29 who failed. In addition, three other segments were extracted to support other analyzes: two referring to week 3 of the course (U03), also separating approved and failed students, and one containing the approved students with final grade A (maximum grade). Therefore, five data sets were used for the experiments and analysis of results.

The Fig. 3 presents a summary of the general scheme used, in which are shown the Log_passed (log of approved students events containing 20294 events), the Log_failed (log of failed

students events containing 4040 events), the Logs_passed_U03 (containing 2765 events), the Logs_failed_U03 (containing 667 events) and the Logs _passedA (containing 8285 events). Regarding the complete event log (Log_all), the EPM was applied only to gather some general information as a basis for the analysis of the results obtained for the 5 segments.

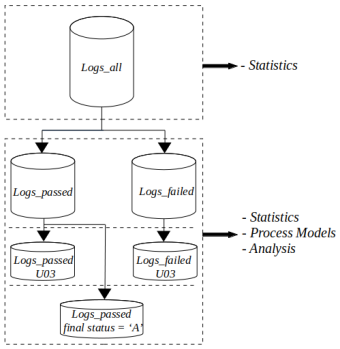

Figure 3. Scheme of the event logs for experiments and analyzes.

The Student Name attribute was defined as the Case attribute and the Component attribute as the Activity Name. The Timestamp attribute is used by the algorithms so that they can determine the order of events.

For this work, the configuration of the event log was defined by the association of the Student attribute as Case and the Component attribute as Activity Name. In addition, the prospects for the application of the EPM occurred on Case, Control-Flow (represented by the process models) and Time (specifically regarding the frequency of activities). In addition, the type of PM executed was Discovery.

The Discovery algorithm used was the Heuristic Miner (HM) [28] [29] [5], using scripts developed in Python with support of the PM4Py library. In addition, techniques were applied to obtain statistical information from the event log (both via the ProM Framework and the developed Python scripts).

IV. Experiments and Analysis of Results

We present below the description of the experiments and respective analyzes on the 5 data sets: log of the approved, log of the failed, log of the unit 3 (separated into approved and failed) and log of approved with grade A.

In general, the events present in the course log are related to 6 activity-types, namely: Virtual Programming Laboratory (VPL), Quiz, Lesson, Upload Files, Tasks and Forum. It is important to highlight that the first two activities were mandatory (assessment) and the others were optional. Although not mandatory, the Lesson activity was highly recommended, as in theory, the student should first study the lesson and then carry out the assessment activities.

A. Full event log

The complete event log (Logs_all) is described by 73 students (Cases attribute) and 24334 events. The first event took place on a Monday, June 10, 2019, at 6:23PM and the last, on a Friday, October 27, 2019, at 11:02PM.

Each student performed an average of 333 events, in which the smallest amount being 2 and the largest being 1003. In addition, the majority of students predominated their paths on average over 4 activity-types, considering that 2 were mandatory and 1 highly recommended.

The Table I presents a summary of the distribution of occurrences of the 6 activity-types.

Table I
Distribution of occurrences of the 6 activity-types.

Activity-Type (Level 1)	Occurrences	Qty Activity (Level 2)	Qty Actions (Level 3)
VPL	13560	25	6
Quiz	8656	15	4
Task	119	3	1
Lesson	1741	7	4
Upload Files	248	3	2
Forum	10	1	1

Among the activity-types, the VPL was the one that most generated occurrences (13560), followed by the Quiz (8656) and after Lesson (1741). The second and third columns show the number of activities (level 2) and actions (level 3). For example, VPL has 25 exercises (level 2) and 6 record options (level 3), such as: edit, evaluate, send, check a submission, check a description and delete. The same reasoning can be done for the other activity-type (level 1).

Understanding this type of scenario (level 1, 2, and 3) is important, as some or many of these level 3 internal activities may not be critical to the analysis. For example, in the case of VPL "evaluate" (level 3) could be considered the most important option, since it indicates that the student required his/her submission to be evaluated. However, for other analyzes, it may be important to check how many times he has used the edit option without necessarily submitting or evaluating. It all depends on the objectives to be established in the analyzes. For this, it is important to have a complete view of the activity-types, activities, quantities, and so on.

B. Event Log subsets

The results of the EPM applied to the event logs separated from the students that were approved (Logs_passed) and failed (Logs_failed) started from historical information according to Table II.

Table II
General data that characterizes the event logs of the approved and failed students.

	Logs_passed	**Logs_failed**
Cases	44	29
Event classes	6	6
First event	Mon, jun-10-2019, 18:23	Mon, jun-10-2019, 19:04
Last event	Sun, oct-27-2019, 23:02	Wed, oct-23-2019, 23:26
Mean events	461	139
Min events	146	2
Max events	1003	391

In the subset of the approved students, it was identified that they performed, on average, 5 of the 6 possible activity-types. This same group has one or more students who performed the largest number of events (= 1003). The lowest number of events performed by those approved was 146 records. Among the group of failed students, the average was 4 activity-types performed and this group also includes the student who performed the least number of events (= 2). Another discrepancy is the average of events performed, whereas the approved had an average of 461, the students who failed had an average of only 139.

Complete process models can be obtained through the HM algorithm, in which it is possible to visualize a Control-flow of all activities, connections and number of occurrences, such as those presented by Figures 4 and 5. In this algorithm, some parameters can be configured indicating threshold values that imply, for example, the cuts of the generated models. Depending on the values used, activity-types and events can be cut from the model. In this study, the models were generated with threshold values that allowed to obtain the complete set containing all the activities and connections of the process (parameters="dependency_thresh": -1, "and_measure_thresh":1, "dfg_pre_cleaning_noise_thresh":0.0).

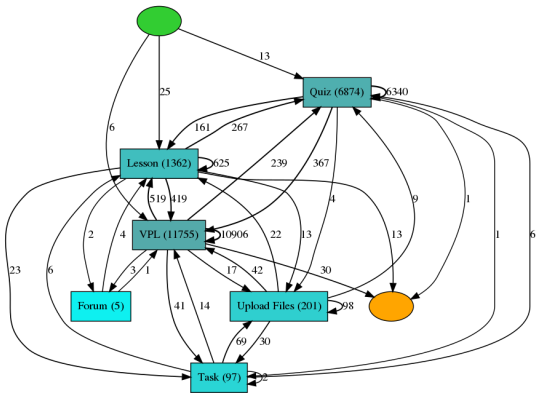

Figure 4. Process model of Logs_passed generated by HM.

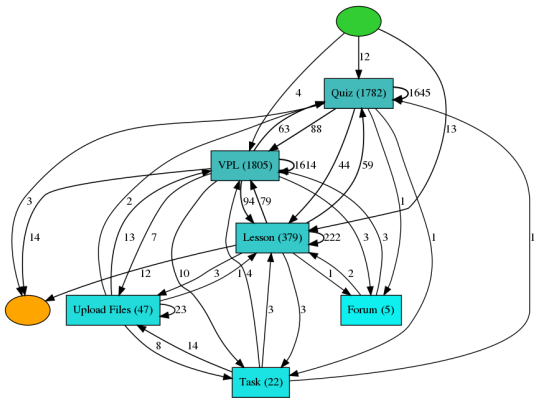

Figure 5. Process model of Logs_failed generated by HM.

These process models are formed based on the set of Traces of all students in each group. Each Control-Flow generated shows the possible activities and connections described by the respective event log. Graphically there are two nodes created artificially by HM, one called the source (on the top - green), which represents the distribution of the beginning of the students' paths by the activities, and another called the sink (bottom - pink), which represents the distribution of the end of the paths. For example, in Figure 4 the source node directs three connections to the activity-types performed initially: VPL (6), Lesson (25) and Quiz (13). The values contained in the edges indicate how many students started for a given activity. Thus, in the group of approved, 25 students (57%), out of a total of 44, started their actions by accessing the Lesson activity-type.

In Figure 5, the source node indicates that students from the group of failing students also started their actions by the same activity-types: VPL (4), Lesson (13) and Quiz (12). In this group, 13 students (45%), out of a total of 29, started their actions by accessing the Lesson activity-type. Both in Figure 4 and in Figure 5 it is also possible to see the connections between activity-types nodes (Activity Name attribute) by means of directed arrows between them, in addition to the values that express the number of occurrences. For example, in Figure 4, starting from the node Lesson, 419 events (occurrences) went to VPL and 625 continued to another Lesson.

The first step of the HM algorithm includes calculations of causal dependencies between pairs of activities. For this, the HM considers only the order of events in a Case (student), and not among the Cases, using the Timestamp of the activities to calculate the ordering.

Therefore, the values of causal dependencies measure how much a dependency relationship exists between pairs of activities X and Y (denoted by $X =>_W Y$), based on the frequency in which the activities occur in sequence. In this case, if one activity is always followed by another, it is likely that there is a probable dependency relationship between the two activities. For this, it occurs by building a dependency graph represented by a matrix, whose values are calculated according to the Equation 1 ([28] [29]), where $-1 <= (X =>_W Y) <= 1$ e $X >_W Y$ is the number of occurrences of connection X and Y.

$$X =>_W Y = \frac{(|X >_W Y| - |Y >_W X|)}{(|X >_W Y| + |Y >_W X| + 1)} \quad (1)$$

The analysis of this matrix can show some student habits and eventually indicate to teachers reformulations in their didactic sequences. The results obtained from the application of the Equation 1 vary between -1 and 1. The closer to 1, the greater the causal dependence. Thus, a high value of causal dependence (positive value close to 1) is directly related to a greater distance between $X >_W Y$ e $Y >_W X$ (in reverse). That is, if in the opposite direction the value is close to zero, the result will be a greater causal dependence in the direction $X >_W Y$.

The Table III shows the casual dependence values of the Control-flow of students in the group of approved ones (see

Figure 4). In the table the three highest values are: (a) 0.625 –Task » Quiz; (b) 0.5666 - Lesson » Task; (c) 0.4821 - VPL » Task. It should be noted, however, that 0.625 was obtained from a low number of occurrences: 6 occurrences of Task » Quiz). As there was only 1 occurrence in the opposite direction (Quiz » Task), the result was relatively high (in the direction of the value 1): 0.625. Obviously, it is up to the teacher to interpret this meaning (high or low) from his expectations in relation to didactic strategies. The second highest value (0.5666) was obtained from 23 occurrences (Lesson » Task) and 6 occurrences in the opposite direction (Task » Lesson): ((23 - 6) / (23 + (6 + 1)) = 0.5666.

In the context of this experiment and considering the established didactic sequence, there was an expectation of high causal dependence between Lesson » VPL, that is, the student studies Lesson and then performs the VPL activity (writing and program evaluation). According to Table III the value for Lesson » VPL is 0.1064, that is, a value far from the limit 1 obtained by: (519 - 419) / (519 + (419 + 1)) = 0.1064. In fact, it is noted that the distance between the two directions is only 100 occurrences (519-419) and it is not possible to identify a dependency. Thus, a deeper dive into the data (level 2 and level 3) is necessary for a better analysis, that is, just analyzing the dependency table based on level 1 seems to be insufficient.

Thus, it does not seem possible to state, based on these results, that sequences of pairs influenced a successful trajectory of students. For example, based on the values presented, what can be observed is that the student follows the activity-type "Task" to "Quiz" (0.625), and from "Lesson" to "Task" (0.5666) and so on.

Table III
CAUSAL DEPENDENCY MATRIX OF APPROVED STUDENTS.

	Quiz	**Lesson**	**VPL**	**UpFile**	**Task**	**Forum**
Quiz	0.0	-0.2470	0.2108	-0.3571	-0.625	-
Lesson	0.2470	0.0	-0.1064	-0.25	0.5666	-0.2857
VPL	-0.2108	0.1064	0.0	-0.4166	0.4821	0.4
UpFile	0.3571	0.25	0.4166	0.0	-0.39	-
Task	0.625	-0.5666	-0.4821	0.39	0.0	-
Forum	-	0.2857	-0.4	-	-	-

The symbol '-' indicates that there is no connection.

To stipulate a sequence, which must include more than one pair of activities, an order can be created by aggregating the pairs. The first step is to define which activity to start with, but there is no defined rule for this. For example, you can choose an activity at random or by choosing one of the activities that were used to start the process. Note that through the Control-Flow of Figure 4, it is possible to start the sequence by the "Lesson" activity. The next step is to follow the highest values of causal dependence between the pairs. In addition, it is necessary to define a size limit for this sequence to be stipulated, otherwise an infinite loop occurs.

Thus, limiting the size of the sequence to five and starting from "Lesson" we have: the highest causal value is "Task" (0.5666); followed by "Task" the highest value is "Quiz" (0.625); from "Quiz" the highest value is "VPL" (0.2108); and from "VPL" the highest is "Task" (0.4821). At this point, we have a return to the starting point (Task). This is one of the possible sequences, starting with "Lesson", which in theory should be the beginning of all students: "Lesson » Task » Quiz » VPL » Task" .

Some activity-types that appear in the sequence may not be in accordance with the expected didactic sequence. Thus, it should be considered in the analysis that a pair of activities with relatively low frequency could be discarded. For example, in this sequence "Lesson (23) » Task (6) » Quiz (367) » VPL (41) » Task, the values 6, 23 and 41 can be considered low, when compared to 367 (Quiz » VPL). Therefore, this five-fold sequence may not be the main one. These values are also low when checking the number of occurrences in the pairs Lesson (419) » VPL and Lesson (267) » Quiz.

Thus, a strategy could be to consider values belonging to a segment that contains the main activities over a period or a more specific group, whether they are characterized by the high value of occurrences and/or by the degree of importance, defined by the teacher, for each activity-type, in the didactic sequence of the course.

In order to dive a little deeper into the data and better understand the results, we processed and analyzed a subset containing only the data from one unit (U03). The U03 addresses the content of "Repetition Structures" and occurs at an intermediate point in the course, at a time when possible dropouts probably already occurred. In addition, this content is covered two weeks before an in-person exam. It is worth remembering that the activities of each unit of the course had weekly deadlines.

The Table IV presents a summary of the actions of students who passed and failed in the scope of U03. These values emerged from 2765 events generated by students who passed and 667 of those who failed. Note that 100% of those approved registered events in that period, with an average of 63 events performed, with a minimum of 13 and a maximum of 220. As for those who failed, only 65.5% performed U03 activities, with an average of only 13 registered events, with a minimum of 1 and a maximum of 90.

Table IV
GENERAL HISTORICAL STATISTICS OF U03

	Logs_passed	**Logs_failed**
Cases	44	19
Event classes	4	4
Mean events	63	35
Min events	13	1
Max events	220	90

Obviously, during the U03, it was not yet defined which students would pass or fail. However, with this type of analysis, a projection could have been made on those students who would not be performing the activities, as well as which path they would be taking between the activities. As for finished

classes, as in the case of the event log under study, it is possible to verify the characteristics of the trajectory performed by each group, either through Control-flows and/or by Traces. For example, through Traces, you can select one or more students with good final performances and thus compare them with those with lower performances. Figures 6 and 7 show the process models obtained via HM for each student subgroup (pass and fail) during the U03.

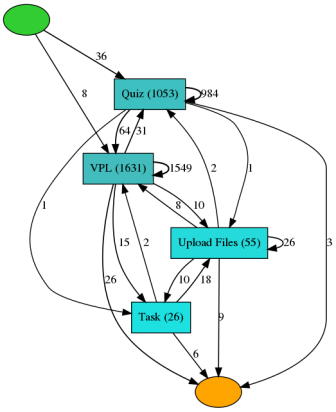

Figure 6. Process model extracted via HM of the 3rd week of the approved students.

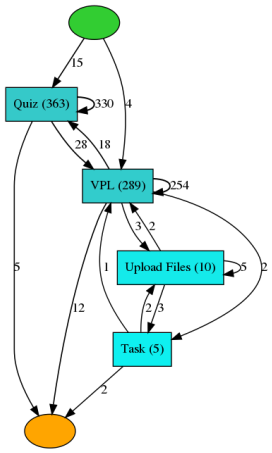

Figure 7. Process model extracted via HM of the 3rd week of the failed students.

It is valid to take into account, for the purpose of proportional frequency comparisons, that during U03 the students had as internal activities (level 2) 3 Quizzes, 3 VPL, and 1 Task. In view of this scenario, most of the students of the two groups started their processes through the Quiz activity and ended in VPL. However, those who passed were more frequent in the VPL activity (59%) and those who failed were more frequent in the Quiz activity (54.4%).

Although VPL and Quiz activities were highlighted in relation to their frequency values, the Task activity was also important throughout the trajectory of students in the discipline process during U03. In this case, based on the fact that a high value of $X =>_W Y$ results from a large gap between the values of $X >_W Y$ and $Y >_W X$ (Equation 1), and observing the obtained process models (Figures 6 and 7), it is possible to verify that for the approved students the connection of the pair VPL » Task has a value that may be relevant in a causal dependency observation ($VPL >_W Task = 0.7222$). For those who failed, despite a positive value ($VPL >_W Task = 0.25$) the dependency was much lower.

Note that among those who were approved, 26 (59%) performed the Task activity-type, while the same activity was performed by only 5 (26%) among those who failed. In addition, the occurrences of Task can be associated exclusively with those of Upload Files, since this happens only for Task. Thus, the pair Task + Upload Files for those who passed were represented with 81 occurrences and for those who failed with just 15. It is also worth noting that for each Task, only 1 record is counted for each student in the event log, while for Upload Files it can be more than one.

The trajectory of students at U03 can be verified by the occurrence values in each internal activity of level 2. Each one corresponds to the 3 exercises proposed for VPL (L1, L2 and L3), the 3 for Quiz (Q1, Q2 and Q3) and 1 for Task + Upload Files (TU). For the group of approved students, L1, L2 and L3 were the most accessed, with 728 (26.66%), 467 (16.9%) and 436 (15.77%), respectively. Still among those approved, the other activities account for Q1 with 328 (11.68% of occurrences), Q2 with 416 (15.04%), Q3 with 309 (11.17%) and TU with 81 (2.92%). For those who failed, the most accessed were L1 with 134 (20.09%) of occurrences and Q1 with 133 (19.04%), followed by Q2 and Q3, both with 115 (17.24%) and TU with 15 (2.25%). These values show some differences that may represent the focus of students during U03 and consequently analyze a projection for the course of the discipline. Note that the approved students dedicated themselves to the three VPL activities, which are limited to the practice of programming using language. Those who failed dedicated themselves only to the first exercise (L1), focusing on the internal Quiz activities (Q1, Q2 and Q3).

Obviously, it may be feasible to consider internal activities of level 3. For example, in VPL, there are six possibilities of operation for each activity of level 2 (L1, L2 and L3), namely: submission deleted, submission edited, submission evaluated, submission uploaded, submission viewed, and description viewed. This means that, if the interest is, for example, to analyze the operations that really indicate a reality of the practice of the exercises, then it would be worth highlighting those related to editing (edited), submitting (uploaded) and evaluating (evaluated). Thus, from this perspective, 61% of those approved and 49% of those who failed dedicated themselves to this practical cycle for L1, for L2 56% of those approved and 44% of those who failed and for L3 61% of those approved and 52% of those who failed.

It is also possible to design a process model to be considered ideal, based on student segments with optimal performance results, for example, considering all units and all students approved with final grade A, which supposedly made a satisfactory path. In this case, all (= 16) performed 5 of the 6 possible activities. In all, 8285 events were performed by

the students, with an average of 518 per student. The lowest number of events performed by any student was 327 and the highest 1003. This means that they were very active, based on Table II. Regarding activities, 56.25% of them started the process for Lesson and 61.38% focused on VPL.

V. Final Considerations

This study aimed to apply the EPM to verify the learning paths made by students in an Introduction to Programming course. For this, we used a course log file, obtained from the Moodle platform with 85759 events that were reduced to 24334 events described by six attributes. From this, five subsets were generated and four process models were obtained using the Heuristic Miner algorithm. Then, these models were analyzed, together with of other statistical informations. The results showed that the students follow different paths from those planned by the teacher.

We concluded that generating each process model considering the perspectives of Control-Flow, Case, and Time, together, enabled a better understanding of the different paths revealed. This understanding was expanded with the use of the hierarchy technique, advancing in more profound levels of the log (levels 1, 2, and 3). The analysis of a specific period also proved to be promising. It should be noted that in traditional PM activities are treated as ordered tasks based on established business models. However, in the MPE, the actions of each student in the course can be quite different, even if the teacher has previously established an ideal order.

As limitations, we highlight that this work did not consider the students' prior knowledge in the analyzes. We understand that this can be an essential element of learning difficulties and can affect students' trajectories.

As future work, this research intends to design an architecture to support the teacher's work in the analysis of these learning paths to better plan or redesign their course.

References

[1] R. Dolak, "Using process mining techniques to discover student's activities, navigation paths, and behavior in lms moodle," in *Innovative Technologies and Learning*, L. Rønningsbakk, T.-T. Wu, F. E. Sandnes, and Y.-M. Huang, Eds. Springer International Publishing, 2019, pp. 129–138.

[2] O. Zawacki-Richter and A. Qayyum, *Open and Distance Education in Asia, Africa and the Middle East*, 1st ed. Springer Singapore, 2019.

[3] N. Zanjani, "The important elements of lms design that affect user engagement with e-learning tools within lmss in the higher education sector," *Australasian Journal of Educational Technology*, vol. 33, no. 1, pp. 19–31, 2017.

[4] H. Al-Samarraie, B. K. Teng, A. I. Alzahrani, and N. Alalwan, "E-learning continuance satisfaction in higher education: a unified perspective from instructors and students," *Studies in Higher Education*, vol. 43, no. 11, pp. 2003–2019, 2018.

[5] W. van der Aalst, *Process Mining: Data Science in Action*, 2nd ed. Springer-Verlag Berlin Heidelberg, 2016.

[6] ——, "Process mining: Overview and opportunities," *ACM Trans. Manage. Inf. Syst.*, vol. 3, no. 2, pp. 1–16, 2012.

[7] ——, "Process mining: Making knowledge discovery process centric," *SIGKDD Explor. Newsl.*, vol. 13, no. 2, p. 45–49, 2012.

[8] R. Sayiam and O. Sahingoz, "A process mining approach in software development and testing process: A case study," *Lecture Notes in Engineering and Computer Science*, vol. 1, pp. 407–411, 01 2014.

[9] F. R. Blum, "Metrics in process discovery," 2015.

[10] A. Rozinat, A. Medeiros, C. Günther, A. Weijters, and W. Aalst, "Towards an evaluation framework for process mining algorithms," *Reactivity of Solids*, pp. 1–20, 01 2007.

[11] A. Bogarín, R. Cerezo, and C. Romero, "A survey on educational process mining," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 1–17, 2018.

[12] ——, "Discovering learning processes using inductive miner: A case study with learning management systems (lmss)," *Psicothema*, pp. 322–329, 08 2018.

[13] M. A. Ghazal, O. Ibrahim, and M. A. Salama, "Educational process mining: A systematic literature review," in *2017 European Conference on Electrical Engineering and Computer Science (EECS)*, 2017, pp. 198–203.

[14] J. Salazar-Fernandez, M. Sepulveda, and J. Munoz-Gama, "Influence of student diversity on educational trajectories in engineering high-failure rate courses that lead to late dropout," 04 2019, pp. 607–616.

[15] A. Bogarín, R. Cerezo, and C. Romero, "Discovering students' navigation paths in moodle," *International Conference on Educational Data Mining, Madrid*, pp. 556–557, 2015.

[16] A. Bogarín, C. Romero, and R. Cerezo, "Applying data mining to discover common learning routes in moodle," *Revista Edmetic*, pp. 73–92, 2016.

[17] R. Cerezo, A. Bogarín, and C. Romero, "Process mining for self-regulated learning assessment in e-learning," *Journal of Computing in Higher Education, Springer Science+Business Media, LLC, part of Springer Nature*, pp. 1–15, 2019.

[18] F. Silva, T. R. Silva, and E. Aranha, "Mineração de processo educacional - uma revisão sistemática da literatura," *Brazilian Symposium on Computers in Education*, vol. 30, pp. 1036–1045, 2016.

[19] K. Tóth, F. Rolke, Heiko; Goldhammer, and I. Barkow, *Educational process mining. New possibilities for understanding students' problem-solving skills.* The Nature of Problem Solving: Using Research to Inspire 21st Century Learning, OCDE Publishing, 2017, ch. 12.

[20] C. Romero, R. Cerezo, A. Bogarín, and M. Sánchez-Santillán, "Educational process mining: A tutorial and case study using moodle data sets," *Data Mining And Learning Analytics: Applications in Educational Research*, no. 1, pp. 3–27, 2016.

[21] A. van den Beemt, J. Buijs, and W. van der Aalst, "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 Distance Learning*, vol. 19, no. 5, pp. 38–60, 2018.

[22] R. Rajendran, A. Munshi, M. Emara, and G. Biswas, "A temporal model of learner behaviors in oeles using process mining," 11 2018, pp. 276–285.

[23] S. Shirazi, D. Gasevic, and M. Hatala, "A process mining approach to linking the study of aptitude and event facets of self-regulated learning," 03 2015, pp. 265–269.

[24] D. Etinger, *Discovering and Mapping LMS Course Usage Patterns to Learning Outcomes*, 01 2020, pp. 486–491.

[25] Y. Wang, T. Li, C. Geng, and Y. Wang, "Recognizing patterns of student's modeling behaviour patterns via process mining," *Smart Learning Environments*, vol. 6, pp. 1–16, 12 2019.

[26] D. Aulia and I. Waspada, "The design of exploratory application and preprocessing of event log data in lms moodle-based online learning activities for process mining," *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, vol. 5, pp. 124–133, 12 2019.

[27] P. Nafasa, W. Indra, N. Bahtiar, and A. Wibowo, "Implementation of alpha miner algorithm in process mining application development for online learning activities based on moodle event log data," 10 2019, pp. 1–6.

[28] A. Weijters, W. Aalst, and A. Medeiros, *Process Mining with the Heuristics Miner-algorithm*, 01 2006, vol. 166.

[29] S. De Cnudde, J. Claes, and G. Poels, "Improving the quality of the heuristics miner in prom 6.2," *Expert Systems with Applications*, vol. 41, pp. 1–26, 12 2014.