Analysis of Learning Behavior in a Programming Course using Process Mining and Sequential Pattern Mining

Eduardo Machado Real
Federal University of ABC (UFABC)
Santo André, Brazil
State University of MS (UEMS)
Nova Andradina, Brazil
eduardo.real@ufabc.edu.br

Edson Pinheiro Pimentel
Federal University of ABC (UFABC)
Santo André, Brazil
edson.pimentel@ufabc.edu.br

Juliana Cristina Braga
Federal University of ABC (UFABC)
Santo André, Brazil
juliana.braga@ufabc.edu.br

Abstract*—This full paper of the research-to-practice category addresses the problem of discovering patterns in students' learning paths in an introductory programming course. Student interactions in courses delivered through learning management systems (LMS) are stored in event logs. Each event corresponds to some action performed by students on course content. Process Mining (PM) techniques allow extracting knowledge from the event logs generating the trace, that is, the ordered set of events of a given student in a course. PM can also obtain the process models generated from the traces. Process models show where students started and ended their paths, enabling them to characterize possible behaviours. It shows that each student can behave differently, generating different sequences. Applying Sequential Pattern Mining (SPM) techniques make it possible to extract sequential patterns and discover sequences of similar events. Thus, this article aims to present the results of applying PM and SPM techniques to verify and analyze the students' learning paths in an introductory programming course. For this, a log of Moodle events was collected, which went through data selection and cleaning, and segmentation. The Moodle event log stores data that allows exploring information on three levels: (1) activity-type, (2) activity, and (3) action in the activity. Thus, the study aimed to investigate which activity-types, activities, and actions were performed and in what order. Among the applied techniques are those to obtain: event log statistics, process models with the Directly-Follows Graph algorithm and event sequences with the Generalized Sequential Patterns algorithm. The results showed that the students had distinct behaviors when accessing and performed the activities. Overall, we highlight the following two main findings that enabled a better understanding of the different behaviors: (i) verifying process models of Levels 2 and 3 attributes allows to deepen the analysis in the flow of the events of the activities; (ii) The subsequences of interest extracted via SPM can complement the analysis of the students' behaviors.

***Index Terms*—Educational Process Mining, Sequential Patterns Mining, Learning Behaviour, Learning Analysis**

I. Introduction

Learning Management Systems (LMS) have been the leading platform used to mediate courses via the Internet and are composed of several tools to provide content, study and evaluation activities, and communication between peers.

In these environments, the teacher organizes the course sequence according to his didactic-pedagogical strategies to guide the student in his learning process. On the other hand, each student navigates and interacts with the contents and activities of the course, characterizing their learning path. Considering that each student can make distinct paths, the set of paths of all students corresponds to the learning profile in the course.

The analysis and understanding of this learning profile in the course can help diagnose if the students' learning behaviors are adherent to the teacher's pedagogical intentions and how and which components of the courses impact these behaviors.

This analysis is possible because, generally, the LMS record the students' interactions in the course in a file called the event log. Each event recorded in the log corresponds to some action (view, edit, update, submit, evaluate, etc.), of some student, on some component of the course, either of study or evaluation.

Techniques from artificial intelligence can be used to support the processing of these data (logs) in order to facilitate human analysis, such as:

1) Process Mining (PM), in which models of processes are obtained based on the students' learning paths and statistical information about these paths; and
2) Sequential Pattern Mining (SPM), in which it is possible to discover sequences of common actions in the students' paths, and this discovery can support and complement the analysis of the learning paths obtained by the MP.

The basic idea of the PM is to extract knowledge from the event logs recorded by an information system. Thus, it seeks to confront such event logs, i.e., observed behavior, and process models, elaborated by specialists or automatically discovered ([1] [2] [3]) . This extracted behavior is represented and detailed by historical statistical information and process models obtained by PM techniques.

In the field of Education, the application of PM is conceptualized as Educational Process Mining (EPM), whose main objective is to provide information that allows the analysis of educational processes represented by activity event flows carried out by students.

The PM techniques are based on the so-called Traces, being that each Trace is an ordered set of events of a particular student and can be considered a standard student behavior. A trace can indicate, for example, that a student has read a text and then performed an evaluative activity, or vice versa.

The fact that Trace to have the sorting characteristic, that is, indicating the sequence in which the events occurred, makes it possible to apply the SPM, which has been a technique used to discover frequent patterns in sequentially ordered data and was proposed in [4] as the mining problem (and discovery) of subsequences of interest in a set of sequences.

This paper aims to present the results of applying the PM and SPM techniques to verify and analyze the students' learning paths in an introductory course to programming. The course was offered in blended learning and conducted on the Moodle platform, one of the most widely used open-source LMS in the world, and the data was obtained from its event log.

The Moodle event log stores data that allows exploring information at three levels: type-activity (Level 1), activity (Level 2), and action on activity (Level 3). For the application of PM and SPM techniques, the following fundamental questions were asked:

Q1) What activity-types (Level 1) were accessed and in what order?
Q2) What activities (Level 2) were performed and in what order?
Q3) What actions (Level 3) were performed in a specific activity?

The paper is organized as follows: section 2 positions the fundamental concepts of PM, EPM, and SPM and describes some related works; section 3 describes the methods used to conduct the experiments; section 4 presents the results and discussions obtained; section 5 the final considerations are drawn.

II. Background

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

A. Process Mining

PM aims to explore event logs 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 ([1] [2] [3])

Formally, 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_1*, E_2*, E_3*, ..., E_M*), obviously respecting the order of performed events (a chronological records of activities). In addition, it can occur in the Traces repetitions (*), in sequence, of events of the same activity, characterized as a loop.

The PM has as input an event log in which at least three basic attributes must be defined:

- *Case* - elements defined as central objects of analysis;
- *Activity Name* – activities performed and which generate events in the event log;
- *Timestamp* – date and time of recording of log events (start and/or end).

For example, in the following event log, 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*;
- *Traces*: *[<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, *<a, d, c, e, g>* Case 2 and so on.

For the application of PM, there are three main types of sets of techniques that can be used from event logs and models:

- Discovery, which is a technique that has as input an event log (a log file) and produces a process model;
- 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 conforms with the model and vice versa;
- Enhancement, whose main idea is to extend or improve an existing process model using information about the real process recorded in some event log.

Also, there are different perspectives of application that can be considered and adopted [1], such as:

- Control-flow, which focuses on the order of activities;
- 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 [1].

B. Educational Process Mining

EPM, as a method, can open up new ways of analyzing students learning behavior and problem-solving [5]. It can be used, for example, to understand better the educational processes, the students' learning habits, and the factors that influence their performance and to examine specific learning actions performed [6] [7] [8] [9].

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 correctly [10]. Thus, the exploration of historical educational data with appropriate mining techniques allows more detailed analysis of student behaviors [11].

C. Sequential Pattern Mining

SPM techniques can identify patterns in sequences of elements, such as in DNAs, RNAs or proteins (bioinformatics), shopping items (market analysis), sentences/words (text

analysis), and learning paths (learning analysis in technology-mediated education). The patterns found can correspond to complete sequences or, mainly, to fragments of sequences (subsequences).

By definition, SPM is one of the most frequently used methods that study the temporal association between variables and consists of discovering subsequences of interest in a set of sequences. The interest of a subsequence can be measured in terms of several criteria, such as the frequency of occurrence and length [12].

Formally, consider a set of items. A sequence is an ordered list of these items. A sequence $S_a =< a_1, a_2, ..., a_n >$ is said to be contained in another sequence $S_b =< b1, b2, ..., bm >$ if there exists integers $1 \leq i_1 < i_2 < ... < i_n \leq m$ such that $a_1 \subseteq b_i^1, a_2 \subseteq b_i^2, ..., a_n \subseteq b_i^n$ (denoted as $S_a \sqsubseteq S_b$ e S_a is called a subsequence of S_b) [13] [12] [14].

In real applications, a $suporte(S)$ measure is also used. The support (denoted by *minsup*) is a value defined as input to obtain the fraction of the sequences total that contain the extracted subsequences. The support calculation takes the amount that a sequence pattern appears among all the sequences in the data set divided by the total number of sequences [13] [12] [14].

In this context, in many domains, the order of events or elements is important [13]. For example, in the Education, SPM can be applied to discover sequences of common actions conducted in an LMS [12]. Therefore, the common subsequences obtained by the SPM show the common orders of access to the materials and activities of the course.

For example, consider an event log with four Traces $< a, d, c >, < a, c, d, c, b, a >, < b, a, d, c > e < a, c >$. The events a, b, c and d refer to four types of activities that can be accessed in the LMS Moodle. Considering a *minsup* = 0.3, we have, for example, the patterns $< a, c >= 2$ (i.e., it appears in two of the four Traces), $< a, d >= 2$, $< b, a >= 2$ e $< a, d >= 3$.

D. Related work

This section presents and discusses some related work with this study. They are grouped according to their purpose: when exploring the event logs, discovering sequence patterns, and visualizing the behavior and results of the students.

The works described in [6], [15], and [8] also applied PM using segmentation to specify students groups and to facilitate the analysis and added external information (final grades). In [6] the PM was applied to analyze self-regulated learning in groups of students (passed and failed). In the study presented in [15] were verified the behavior patterns of students based on higher and lower scores. Also, based on the results the finals, the authors in [8] researched the students' behavior and late abandonment. In [7] the data was used to identify, for example, the behavior patterns of active or inactive students.

In [16] [17], the authors applied PM to explore different student learning paths, specific actions learning and their impacts on results. In this cases, they explored several possibilities to analyze, such as groups of students, specific period of course, external information, and specific attributes how main activities. In additional, [16] verified overall behaviors, but [17] focused to understand student failures.

In [18] a learning analysis to detect learning tactics and strategies was done. The work used three event logs collected from courses with different course designs and delivery modalities (different learning contexts): three different delivery modalities (blended learning, flipped classroom, and MOOC) and two different course designs (problem-solving and practice-based). As a result, they demonstrated the association between the learning strategies used and academic performance in different learning contexts.

About the SPM, the authors in [12] and [14] discovery frequent navigation patterns in event logs from LMS-Moodle. In [12], was demonstrated of how the presentation of patterns through hierarchical grouping and appropriate visualization can facilitate the interpretation of patterns. In [14], an application method considered more suitable for mining navigation patterns was proposed. In addition, three sequence modeling strategies with educational implications have been proposed.

From data collected from other learning platforms, the authors in [19] describe an approach of the SPM to identify the main body sequences in oral presentations of students during a given course, using seven attributes correspond to hand movements.

In [20] a model was presented to understand and learn the most difficult topics, in which a method is used to analyze the relevance between the processes and the learning situations used through mobile technologies. For this, data from students' video activities were collected.

In [21], the study analyzed the students' behavior in a remote laboratory environment from interaction data to identify factors that predict academic success (correlation between strategies). For this, the authors investigated the relationships between activities during practical sessions and the students' performance in the final test. An analysis of sequence patterns was applied to identify significant sequences of actions during the activities.

III. METHODS

The study considered an event log of an Introduction to Programming course conducted in the LMS Moodle for 12 weeks in semi-presential modality. Initially, this log contains 85759 event records described by six attributes: IP Address, Timestamp (date and time), Student (student name), Component (it specifies the type of activity accessed), Event Context (it contains the title of the event activity) and Event Name (it indicates the action performed on the activity-type).

There is a hierarchical relationship between the attributes Component, Event Context, and Event Name, which can be considered in the exploration of each event:

- Component (level 1) contains the activity-types accessed, that is, Lesson, Task, Virtual Programming Laboratory (VPL) and Quiz;

- Event Context (level 2) contains the specific item (activity) accessed in activity-type, e.g. "S01-04A- sequence structures (ex1)";
- Event Name (level 3) contains the action performed in the specific activity, e.g., submitted, evaluated, etc.

A. Data selection and preparation

For the study described in this paper, a data cleaning step was performed, in which the IP Address attribute was removed as well as several log occurrences that were not necessary, reducing the log to 24334 events and five attributes. More information on data cleaning is described in [16].

In order to facilitate the analysis and understanding of the results, Unit 3 (U03) was defined as the focus of the study in this paper, which contains 3711 events and approaches the topic 'Structures of Repetition'. It is about a topic that is not either extremely introductory or advanced in the context of the discipline, studied just before the middle of the course (a third week out of a total of twelve).

The following datasets were prepared (according to Fig. 1):

1) events of students who passed the finals (2985 events);
2) events of students who failed the finals (726 events);
3) events of one of the VPL activities (728 events) from students dataset who passed the finals.

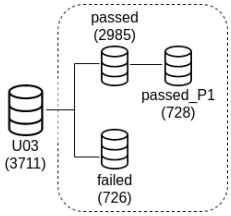

Figure 1. Segmentation of the datasets.

Table II shows an overview of the activity-types (level 1) and specif activities (level 2) present in U03 bellow:

- The first column (Event Context) shows the title of each activity (Level 2);
- The second column (Label) shows a nickname (short name) used to refer to the Event Context (first column) in the analysis and results section since the Event Context description is a long name.

Table I
EVENT CONTEXT ELEMENTS OF U03 AND THE RESPECTIVE CAPTION OF EACH ONE.

Event Context	Label
Lesson: repetition and data validation structures	L
Quiz: 12 - while	Q1
Quiz: 13 - for, do-while and validation	Q2
Quiz: 14B - To solve	Q3
VPL: 15A - JAVA-repetition	P1
VPL: 16A - JAVA-repetition and validation	P2
VPL: 17A - JAVA-repetition and validation	P3
Task: 18 - trace table (verification test)	T

Table II presents a list of possible actions on the VPL activity-type to be analyzed in this study. The first column (Event Name) shows the description of the action, and the second column (Label) shows a nickname (short name) used to refer to the Event Name (first column) in the analysis and results section since the description of the Event Name is a long name.

Table II
EVENT NAME ELEMENTS (ACTIONS) OF VPL AND THE RESPECTIVE CAPTION OF EACH ONE.

Event Name	Caption
mod_vpl: vpl description viewed	A1dvi
mod_vpl: submission uploaded	A2sup
mod_vpl: submission edited	A3sed
mod_vpl: submission evaluated	A4sev
mod_vpl: submission viewed	A5svi
mod_vpl: submission deleted	A6sde

B. Selection of the PM and SPM techniques

In the study described in this paper, we applied PM and SPM, in a complementary way, for the student behaviors analysis.

Fig. 2 presents the overview scheme used for the application of PM and SPM. This scheme has as input an event log (a .CSV file converted to .XES file). From this event log, the PM extracts the student's Traces and uses them as input for their techniques, obtaining process models and statistical information. For SPM, this same set of Traces is used as an input to the GSP algorithm, in which a dataset of sequential patterns is obtained. In this scenario, the results obtained by the PM and SPM can provide insights to the teacher.

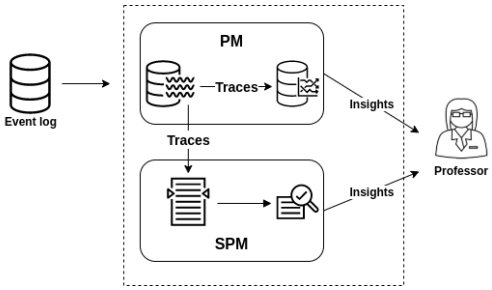

Figure 2. Scheme of application of the PM and SPM.

For PM, an algorithm of the type *Discovery* called *Directly-Follows Graph* (DFG) was used to obtain process models, in which the nodes represent the activity-types and the directed edges (or connections) the control-flow of events between these activity-types (an edge is present between the nodes if there is at least one event in any of the Traces). Both the nodes and the directed edges contain the number of occurrences as a metric to be used as well. In addition, PM techniques were applied to obtain statistical information. The implementations were developed in Pyhton based on the library PM4Py [22].

For the SPM, the Generalized Sequential Patterns (GSP) algorithm was used proposed in [23] and which is based on the Apriori algorithm technique [24]. GSP generates possibly

frequent subsequences (called candidates), calculate support, and prune candidates. For this, in each iteration on the Traces, frequent events and candidate events are formed as common subsequences (details of their operation can be seen in [13]). In this paper, the GSP was implemented in Python by using the GSP-Py [25].

IV. Experiments and Analysis of Results

This section aims to describe the experiments performed and the analysis of results.

A. Summary and description of data characteristics

Table III shows an overview of the event logs from the students who passed and failed the finals.

- 100% of the 44 from the students who passed the finals registered events in the U03 with an average of 68 events held per student, a minimum of 20 and a maximum of 228 events;
- 65.5% (23 of the 29) from the students who failed the finals performed activities in the U03 with an average of 32 events per student, a minimum of 1, and a maximum of 94 events.

Table III
General Historical Statistics of U03.

	Logs_passed	**Logs_failed**
Students	44	23
Events	2785	726
Cases	44	23
Event classes	4	4
Mean events	68	32
Min events	20	1
Max events	228	94

Table IV shows the detailed information for the two datasets (passed and failed) in the U03, considering level 1 (Component or type-activity) on the left side and level 2 (Event Context or activity) on the right side. About the left side (Level 1) of Table IV, the following interpretations can be extracted:

- Among the passed students, the highest percentages occurred in the VPL activity-type with 1631 events (54.6%), that is, the activities that involve program writing, submission, and evaluation;
- Among the failed students, there were more events in the Quiz activity-type with 363 events (50%), which involves manual simulation of program parts, identification of data outputs, and the choice of multiple-choice alternatives.
- The passed and failed students had accesses proportionally close in the Lesson activity-type (8.3% and 9.5%, respectively). However, the passed had an access average almost twice higher (249 occurrences / 44 students = 5.66) than failed (69 occurrences / 23 students = 3), indicating that the passed had more access to study material in the U03.
- In the Task activity-type the passed also had a hits average hits (52 occurrences / 44 students = 1.18) almost six times higher than failed (5 occurrences / 23 students = 0.21). Since that this average was below 1 (0.21), this means that only 20% of the failed accessed or sent this activity related to manual simulation and program tests (step by step).

Table IV
General information of the attributes Component (Level 1) and Event Context (Level 2) (passed and failed datasets).

	Passed		**Failed**			**Passed**		**Failed**	
Level 1	#	%	#	%	Level 2	#	%	#	%
Lesson	249	8.3	69	9.5	Q1	328	11	133	18.3
Quiz	1053	35.3	363	50.0	Q2	416	13.9	115	15.8
Task	52	1.7	5	0.7	Q3	309	10.4	115	15.8
VPL	1631	54.6	289	39.8	P1	728	24.4	134	18.5
					P2	467	15.6	87	12.0
					P3	436	14.6	68	9.4
					T	52	1.7	5	0.7
					L	249	8.3	69	9.5

= Number of occurrences and % = Percentage of occurrences.

On the right side (Level 2) of the Table IV the following interpretations can be extracted:

- Among the passed students, most occurrences were in the P1 activity (24.4%), which was a basic exercise of repetition structure. After then, P2 and P3 had the highest occurrences values, respectively 15.6% and 14.6%;
- Among the failed students, the activities P1 and Q1 were the most accessed (18.5% and 18.3%, respectively), i.e., a VPL activity-type (program writing) and a Quiz activity-type (manual program simulation and the choose of an answer option);
- Considering the sum of the events of P1, P2 and P3, the actions average of the passed were four times higher (728 + 467 + 436 = 1631 events / 44 students = 37.06) than failed (134 + 87 + 68 = 289/23 = 9.96) indicating that the passed ones focused on the three practical laboratory exercises in search of solutions in an attempt to obtain total success (score 10).

B. Generation of the Process Model

In order to answer the first question ('Q1) What activity-types were accessed and in what order (Level 1)?') two process models were generated using the DFG algorithm, setting the Component attribute as *Activity Name*. The two models are shown in Fig. Fig. 3, on the left referring to passed students and the right referring to failed.

These process models summarize the students' navigation control-flow between the activity-types. The nodes are the activity-types (in this case, elements of the Component attribute), and the directed edges correspond to flows between the activity-types. The values present in each node represent the number of events in the activity-type. The values present in each edges connecting nodes X and Y (in the direction of X to Y) represent the number of events performed in Y after an event was performed in X by the same student.

For example, Fig. 3 (left side) shows that in the VPL activity-type, there are 1631 events. On the directed edge

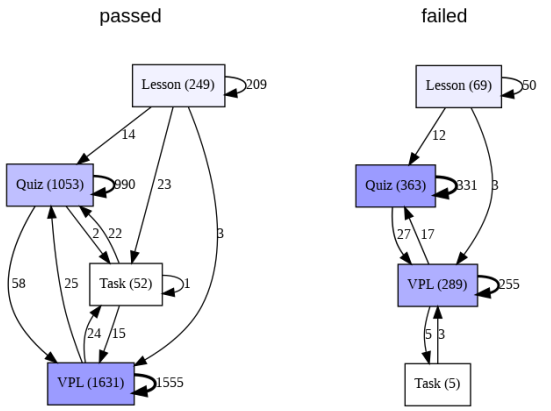

Figure 3. Process models of the activity-types of the passed and failed datasets in the U03.

from VPL (X) to VPL (Y), there is a value of 1555 events, indicating that 95% (1555/1631 = 0.95) of the events in VPL were preceded by an action on the VPL itself, that is, a loop. This is justified because the VPL activity-type has actions such as editing and evaluating, and the same program can be edited and evaluated several times until the expected result is obtained. Still in Fig. 3 (left side), on the directed edge from VPL (X) to Quiz (Y), there is a value of 25 events, that is, 25 actions occurred in Quiz after a student having performed an action in an VPL activity-type. Similar readings can be done on the other nodes and edges.

In order to analyze the starting and ending points of students in the U03, a list of the initial and final activities was obtained, using a PM method to filter the Traces based on the first and last event of each one. Table V shows these activities of levels 1 and 2 and their respective occurrence values for each subset: passed (top) and failed (bottom).

For example, on the top of the Table V (passed), it is possible to observe that the Lesson activity-type (Level 1) was the starting point for 40 of the 44 passed students (90.91%). Also, LPV activities were the endpoints for 27 of the 44 students. Of these 27, 2 ended in the first problem (P1), 6 in the second (P2), and 19 (70.37%) in the third (P3). Thus, the first example shows that most passed students started their journey in the Lesson activity-type as expected. That is, first, the student would study the content and then perform the evaluation activities.

Among failed students (bottom of the Table V), it is possible to observe that most of them also started in Lesson (19 of the 23 students (82.60%)). In addition, four students finished the Lesson, but 2 of them accessed Lesson only once (the only event of the Trace). Furthermore, the other two generated a short loop of Lessons, and they did not follow to other activity-type. Thus, these four students also belong to those 19 students who started in the Lesson. So, they can be analyzed or not, depends on the professor's criterion.

The process models showed in Fig. 3 refer to Level 1 (activity-types). In order to try understanding better the students' behavior and answer the second question ('Q2) What

Table V
START AND END ACTIVITIES OF LEVELS 1 AND 2.

	Start Activities				End Activities			
	Level 1	#	Level 2	#	Level 1	#	Level 2	#
Passed	Lesson	40	L	40	Quiz	3	Q3	3
	Quiz	2	Q1	2	Task	14	T	14
	Task	2	T	2			P1	2
					VPL	27	P2	6
							P3	19
Failed	Lesson	19	L	19	Lesson	4	L	4
	Quiz	3	Q1	3	Quiz	5	Q2	2
	VPL	1	P1	1			Q3	3
					Task	2	T	2
							P1	2
					VPL	12	P2	4
							P3	6

= Number of occurrences.

activities were performed and in what order (Level 2)?'), we choose to detail the event logs (passed and failed) by analyzing the activities performed, i.e., Level 2. Thus, two other process models were also generated from the DFG algorithm, this time setting the attribute 'Event Context' as *Activity Name*. The models are shown in Figs. 4 (passed) and 5 (failed).

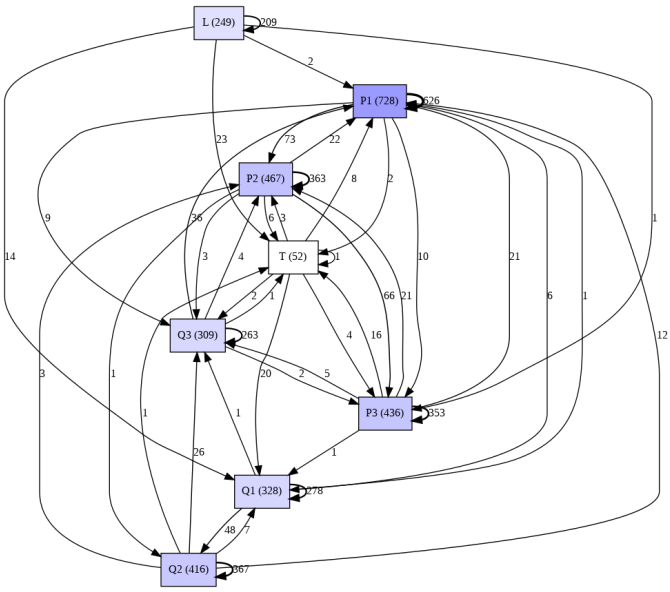

Figure 4. The process model of the passed students, considering the "Event Context" attribute as Activity Name (Level 2).

For example, in Fig. 4 (passed) the P1 activity consists of 728 events. The directed edge from P1 (X) to P1 (Y) contains a value of 626 events, indicating that 86% (626/728 = 0.86) in P1 were preceded by an action in the P1 itself, that is, a loop. The other 102 (728-626) events of P1 originated from other activities, with 36 events (35.6% of 102) being preceded by actions in Q3 activity.

In Fig. 5 (failed) on the directed edge from P1 (X) to P2 (Y) contains 20 events, that is, 20 actions occurred in P2 after that a student performed an action in P1 activity. Considering that the number of occurrences in P2 was 87 (59 of loop and 28 of other activities), P1, therefore, generated 71.43%

(20/28=71.42) of the occurrences in P2. Similar readings can be done on the other nodes and edges.

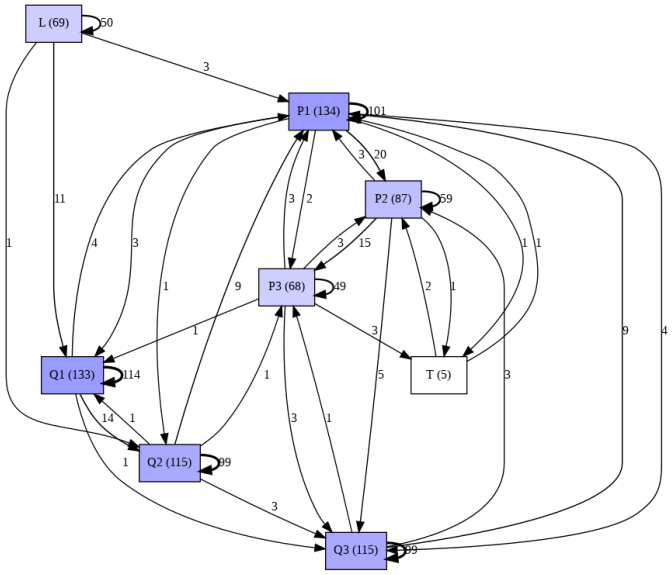

Figure 5. The process model of the failed students, considering the "Event Context" attribute as Activity Name (Level 2).

Based on Table V and passed process model (Fig. 4), we selected P2 activity for analysis, considering the 467 events generated in it (363 are of loop), against 728 events of P1. These values indicate that P2 proved to be an easier activity for students since it had a little more than half of P1's actions. Furthermore, this same observation can be done for P3. Such pieces of evidence can lead the teacher to analyze the complexities of the activities. In addition, the relationships between the activity-pairs P1-P2 and P2-P3 show that the largest events flow between these problems occurred from P1 to P2 and from P2 to P3, even though they could be accessed in any sequence.

Therefore, according to Table V, starting with the Lesson indicates that a student previously studied the theoretical content and then followed the activities and evaluations. For example, Fig. 4 (passed) shows that of the 40 students who started with Lesson, 23 of them followed to Task (T), 14 to Q1, 2 to P1, and 1 to P3. Somehow, by organizing the course, it was possible to get to other activity-types without going through the Lesson.

C. Generation of the Sequential Patterns

In order to answer the third question ('Q3) What actions were taken on a given activity (Level 3) and in what sequence?') the SPM method was applied to reveal temporal associations between variables. In this way, the SPM can point out the most common sequences (patterns) of access to the materials and activities of the courses and make it possible to discover sequences of interest. Obviously, the interpretation of the degree of interest of specific sequences depends on the evaluation of experts in the field of knowledge.

Fig. 6 is an example of a result obtained by the GSP algorithm showing a part of set subsequences extracted from the failed students dataset (level 2). In this execution, the GSP was configured to discover patterns that occurred in at least 50% of Traces, i.e., *minsup* = 0.5. The numbering on each line corresponds to the length of the substrings.

For example, in line 2 (see the number at the beginning of each line in the output file) there are the subsequences of length two (underlined), in line 3 those of length 3 and so on. In addition, each subsequence contains information on the number of Traces in which it appeared. For example, $(P1, P2) : 14$ means that 14 students have as a pattern the subsequence $< P1, P2 >$ (the algorithm does not count if the subsequence appears more than once in the same Trace).

```
1 {('L',): 19, ('Q2',): 16, ('Q1',): 17, ('P1',): 17, ('P2',): 14,
  ('Q3',): 12}
2 {('L', 'L'): 14, ('P1', 'P2'): 14, ('Q1', 'Q1'): 17, ('Q1', 'Q2'): 14,
  ('Q2', 'Q2'): 16, ('Q3', 'Q3'): 12}
3 {('L', 'L', 'L'): 13, ('Q1', 'Q1', 'Q1'): 16, ('Q1', 'Q1', 'Q2'): 14,
  ('Q1', 'Q2', 'Q2'): 14, ('Q2', 'Q2', 'Q2'): 16}
4 {('Q1', 'Q1', 'Q1', 'Q1'): 16, ('Q1', 'Q1', 'Q1', 'Q2'): 14,
  ('Q1', 'Q1', 'Q2', 'Q2'): 14, ('Q1', 'Q2', 'Q2', 'Q2'): 14,
  ('Q2', 'Q2', 'Q2', 'Q2'): 16}
```

Figure 6. Part of the failed students subsequences extracted.

Based on the extracted subsequences patterns, different analysis /types criterion can be defined, such as:

1) Consider only subsequences of length two in which the pair of activities/actions are different and except loops. For example, $< a_x, a_y >$.
2) Consider all subsequences extracted up to an exact length, in which the activities in sequence are different and except loops. For example, $< a_x, a_y, a_z >$.
3) Consider all subsequences up to an exact length, considering one or more different activities and loops.

Considering the subsequences of different activities (without loop) and with length 2, the following observations can be recorded:

- Among those who failed, only two subsequences were extracted: $(Q1, Q2) : 14$ and $(P1, P2) : 14$, i.e., these subsequences are present in 14 (60.86% of Traces) of the 23 students (Fig . 6). However, according to the respective process model of Fig. 5, it is noted that there were 14 occurrences from Q1 to Q2. That is, each student performed this subsequence only once (14 occurrences and 14 subsequences). Still, in Fig. 5 it is noted that there were occurrences from P1 to P2. There were at least six repetitions of this subsequence among the 14 Traces in which it is present (20 - 14 = 6).
- Among the passed students, six subsequences were extracted: $(Q1, Q2) : 41$, $(Q2, Q3) : 26$, $(Q3, P1) : 33$, $(P1, P2) : 40$, $(P2, P3) : 40$ and $(L, T) : 23$. In this case, $< Q1, Q2 >$ is present in 41 of the 44 students (i.e., in 93.18% of Traces) and $< P1, P2 >$ and $< P2, P3 >$ in 90.90% of Traces. To identify which actions were taken in each pair, it is necessary to explore level 3. A specific case of level 3 will be described later.

- It is concluded that the passed students presented a greater variety of pairs of activities concerning those who failed (6 to 2), and these patterns show that, in a certain way, they followed a expected order to perform the activities, that is, the sequence in that were in the organization of the course.

In order to identify which actions were taken in a specific activity, it is necessary to explore level 3. To exemplify, we selected the activity P1 that stood out for being the activity with more events among the approved students (728 occurrences), according to Fig. 4. For P1 activity (of the VPL type) there were six possibilities of actions ($A1dvi$, $A2sup$, ..., $A6sde$), as described in Table II. It is noted that 42 of the 44 passed students performed at least some action in P1.

Thus, a process model referring to the events of activity P1, only from approved students, was generated, applying the DFG algorithm and considering level 3 (the actions in P1), i.e., the 'Event Name' attribute was used as *Activity Name*. In addition, the activities that mark the starting and ending point have also been recovered. According to the process model presented in Fig. 7 all students (42) started with the action $A1dvi$ (description view) and 21 (50%) of them ended with $A5svi$ (submission view), 13 in $A1dvi$, 6 in $A4sev$ and 2 in $A3sed$.

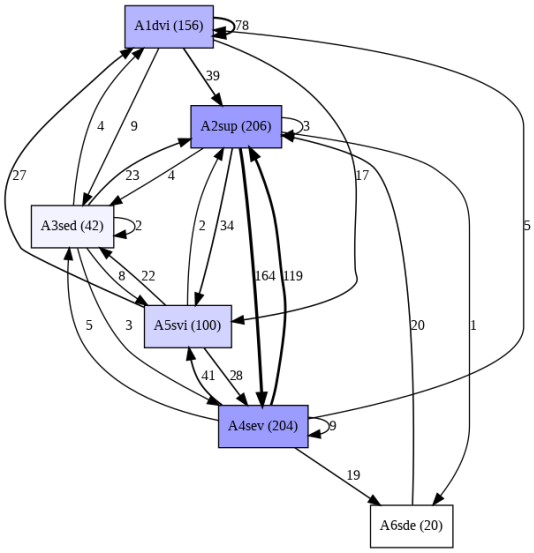

Figure 7. Process model of the P1 activity of the students passed, considering the "Event Name" attribute as Activity Name (Level 3).

According to the process model showed, there is a flow of events of greater occurrence (164), which is from $A2sup$ to $A4sev$, showing that, in general, these students focused a lot on making the submission and then the evaluation of the P1 activity.

Fig. 8 shows the subsequences between the six possible actions in the P1 activity, which were generated using the GSP algorithm to discover patterns that occurred in at least 50% of Traces, i.e., *minsup* = 0.5.

```
1 {('A1dvi',): 42, ('A5svi',): 32, ('A2sup',): 36, ('A4sev',): 35}
2 {('A1dvi', 'A1dvi'): 34, ('A1dvi', 'A2sup'): 36, ('A2sup', 'A4sev'): 28,
  ('A2sup', 'A5svi'): 21, ('A4sev', 'A2sup'): 22, ('A4sev', 'A5svi'): 25}
3 {('A1dvi', 'A1dvi', 'A1dvi'): 23, ('A1dvi', 'A1dvi', 'A2sup'): 30,
  ('A1dvi', 'A2sup', 'A4sev'): 24, ('A2sup', 'A4sev', 'A2sup'): 21}
```

Figure 8. Subsequences of the actions in the activity P1(level 3) of the students passed.

For example, based on the subsequences of length 1, it is possible to verify that 35 students in some time evaluated P1 ($A4sev$) and 36 submitted ($A2sup$). In addition, considering the subsequences of length 2, 28 students performed $< A2sup, A4sev >$, 22 $< A4sev, A2sup >$, 25 $< A4sev, A5svi >$ and so on. There are also subsequences of length 3. For example, 24 students performed $< A1dvi, A2sup, A4sev >$, that is, they viewed the description ($< A1dvi$), uploaded it ($A2sup$) and then evaluated the solution ($A4sev$).

In order to identify the actions in activity P1 of one of the 44 students passed, we selected the one that generated the most events in it (in total, 32 events). The events flow for this student is represented by a distance matrix, in which the values indicate the number of events that follow from one action X to another Y (denoted by $X =>_W Y$), as shown in Table VI. For example, from the action $A2sup$ to $A4sev$, there were nine events.

Table VI
EVENTS FLOW MATRIX IN THE P1 OF ONE OF THE PASSED STUDENTS.

X \ Y	*A1dvi*	*A2sup*	*A3sed*	*A4sev*	*A5svi*	*A6sde*
A1dvi	2	1	1	0	1	0
A2sup	0	0	0	9	1	0
A3sed	0	0	0	0	1	0
A4sev	0	7	0	0	1	2
A5svi	2	0	0	1	0	0
A6sde	0	2	0	0	0	0

```
[<A1dvi, A1dvi, A2sup, A4sev, A2sup, A4sev, A2sup, A4sev,
A2sup, A4sev, A2sup, A4sev, A6sde, A2sup, A4sev, A6sde,
A2sup, A4sev, A2sup, A4sev, A2sup, A4sev, A2sup, A5svi,
A1dvi, A5svi, A4sev, A5svi, A1dvi, A1dvi, A3sed, A5svi>]
```

Figure 9. Subsequences of the actions in the P1 activity of a student passed.

Fig. 9 shows the generated Trace, in which the student starts activity P1 with the action $A1dvi$ and ends it with $A5svi$. In addition, the subsequence $(A2sup, A4sev)$ that appears throughout the Trace was performed nine times, and this can indicate that the student has submitted and evaluated P1 in search of the best solution. In the opposite direction, $(A4sev, A2sup)$, seven events were performed, and this shows that the student made successive attempts in between the pair of actions (a loop submit-evaluate-submit). For example, the subsequence $(A2sup, A4sev, A2sup)$ allows verifying that this 'loop' was performed six times.

Another analysis was based on a specific point of Trace. In this case, starting from the second event is the subsequence

$(A1dvi, A2sup, A4sev)$, which shows that the student *read the description, submitted and evaluated* and this can indicate that a path expected by the teacher was performed, even if any other subsequence is allowed.

V. Final Considerations

This paper aimed to present PM and SPM to verify students' learning behaviors in an Introduction to Programming course conducted in the LMS-Moodle for 12 weeks in semi-presential modality. For this, three event logs extracted from unit 3 (U03) of the discipline were used, segmented based on the students who were passed and failed the finals. The study obtained process models and statistical information through the PM and common subsequences of events through the SPM.

The results showed that the students had distinct behaviors when performed the activities. Overall, we highlight the following findings that enabled a better understanding of the different behaviors: (1) verifying process models of Levels 2 and 3 attributes allows to deepen the analysis in the flow of the events of the activities, specifying the accesses and the order in which they were performed; (2) process models based on Level 2 also allow defining activities relevant for more detailed analysis in the actions performed on it, based on common subsequences among students; and (3) The subsequences of interest extracted via SPM can complement the analysis of the students' behaviors, for the Level 2 (specific activities) as well as for the Level 3 (actions).

As future work, this research intends to develop a tool to facilitate the teacher's work in the analysis of the learning behaviors using PM and SPM. For SPM, for example, it must be allowed to the professor sets parameters to obtain subsequences of interest so that are better appropriate for human analysis. For example, the teacher can obtain subsequences based on a defined length (of a particular length, from a length or a range of lengths), obtain subsequences with or without a loop, among others.

References

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

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

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

[4] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh International Conference on Data Engineering*, 1995, pp. 3–14.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] A. Bogarín, R. Cerezo, and C. Romero, "Discovering learning processes using inductive miner: A case study with learning management systems (lmss)," *Psicothema*, pp. 322–329, 08 2018.

[10] 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.

[11] H. Awatef, B. Gueni, M. Fhima, A. CAIRNS, and S. David, "Process mining in the education domain," 02 2015, pp. 219–232.

[12] L. Poon, S.-c. Kong, T. Yau, M. Wong, and M. H. Ling, "Learning analytics for monitoring students participation online: Visualizing navigational patterns on learning management system," 05 2017, pp. 166–176.

[13] P. Fournier Viger, C.-W. Lin, U. Rage, Y. S. Koh, and R. Thomas, "A survey of sequential pattern mining," *Data Science and Pattern Recognition*, vol. 1, pp. 54–77, 02 2017.

[14] L. Poon, S.-c. Kong, M. Wong, and T. Yau, "Mining sequential patterns of students' access on learning management system," 06 2017, pp. 191–198.

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

[16] E. Real, E. Pimentel, L. Oliveira, J. Braga, and I. Stiubiener, "Educational process mining for verifying student learning paths in an introductory programming course," in *2020 IEEE Frontiers in Education Conference (FIE)*, 2020, pp. 1–9.

[17] E. Pimentel, E. Real, J. Braga, and W. Botelho, "Análise dos resultados de insucesso escolar com o suporte de mineração de processos educacionais," in *Anais do XXXI Simpósio Brasileiro de Informática na Educação*. Porto Alegre, RS, Brasil: SBC, 2020, pp. 132–141. [Online]. Available: https://sol.sbc.org.br/index.php/sbie/article/view/12769

[18] W. Matcha, D. Gasevic, N. Ahmad Uzir, J. Jovanovic, A. Pardo, L. Lim, J. Maldonado, S. Gentili, M. Pérez-Sanagustín, and Y.-S. Tsai, "Analytics of learning strategies: Role of course design and delivery modality," *Journal of Learning Analytics*, vol. 7, pp. 45–71, 09 2020.

[19] F. Roque, C. Cechinel, R. Muñoz, R. Lemos, and T. O. Weber, "Encontrando os padrões sequenciais em apresentações orais de estudantes utilizando sequential pattern mining," 2019, pp. 1896–1905.

[20] E. Doko, L. Abazi Bexheti, M. Hamiti, and B. Prevalla, "Sequential pattern mining model to identify the most important or difficult learning topics via mobile technologies," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 12, 2018.

[21] R. Venant, K. Sharma, P. Vidal, P. Dillenbourg, and J. Broisin, "Using sequential pattern mining to explore learners' behaviors and evaluate their correlation with performance in inquiry-based learning," 2017, pp. 286–299.

[22] A. Berti, S. J. v. Zelst, and W. M. v. d. Aalst, "Process mining for python (PM4Py): Bridging the gap between process-and data science," in *Proceedings of the ICPM Demo Track 2019, co-located with 1st International Conference on Process Mining (ICPM 2019), Aachen, Germany, June 24-26, 2019.*, 2019, p. 13–16. [Online]. Available: http://ceur-ws.org/Vol-2374/

[23] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, ser. EDBT '96. Berlin, Heidelberg: Springer-Verlag, 1996, p. 3–17.

[24] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, p. 487–499.

[25] J. A. d. Prado Lima, "Gsp-py - generalized sequence pattern algorithm in python," jul 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3333988