JOURNAL OF LEARNING ANALYTICS

SOLAR SOCIETY for LEARNING ANALYTICS RESEARCH

Volume 10(2), 126–141. https://doi.org/10.18608/jla.2023.7867

Processing and Understanding Moodle Log Data and Their Temporal Dimension

Daniela Rotelli[1] and Anna Monreale[2]

Abstract

The increased adoption of online learning environments has resulted in the availability of vast amounts of educational log data, which raises questions that could be answered by a thorough and accurate examination of students' online learning behaviours. Event logs describe something that occurred on a platform and provide multiple dimensions that help to characterize what actions students take, when, and where (in which course and in which part of the course). Temporal analysis has been shown to be relevant in learning analytics (LA) research, and capturing time-on-task as a proxy to model learning behaviour, predict performance, and prevent drop-out has been the subject of several studies. In Moodle, one of the most used learning management systems, while most events are logged at their beginning, other events are recorded at their end. The duration of an event is usually calculated as the difference between two consecutive records assuming that a log records the action's starting time. Therefore, when an event is logged at its end, the difference between the starting and the ending event identifies their sum, not the duration of the first. Moreover, in the pursuit of a better user experience, increasingly more online learning platforms' functions are shifted to the client, with the unintended effect of reducing significant logs and conceivably misinterpreting student behaviour. The purpose of this study is to present Moodle's logging system to illustrate where the temporal dimension of Moodle log data can be difficult to interpret and how this knowledge can be used to improve data processing. Starting from the correct extraction of Moodle logs, we focus on factors to consider when preparing data for temporal dimensional analysis. Considering the significance of the correct interpretation of log data to the LA community, we intend to initiate a discussion on this domain understanding to prevent the loss of data-related knowledge.

Notes for Practice

- Log files are often used to perform student behaviour analysis. The correct interpretation of Moodle log data could present some complexities that should be addressed. Biased datasets and, as a result, incorrect interpretation of study findings could result in incorrect knowledge and comprehension of data creation dynamics.
- The contribution of this study is to present the Moodle logging system to illustrate where the temporal dimension of Moodle log data can be difficult to interpret and how this knowledge can be used to improve data processing for temporal analysis.
- Time-on-task is often used as a proxy to model learning behaviour, predict performance, and prevent dropout. A more precise estimation of time may yield insightful information to correctly understand and monitor the timing strategies and behaviours that learners adopt to study.

Keywords

Learning log data, educational log data, Moodle log data collection, time-on-task, temporal dimension.

Submitted: 13/09/2022 — **Accepted:** 24/04/2023 — **Published:** 11/08/2023

Corresponding author [1] *Email: daniela.rotelli@phd.unipi.it Address: Department of Computer Science, University of Pisa, Pisa, Italy. ORCID ID: https://orcid.org/0000-0002-0943-6922*

[2] *Email: anna.monreale@unipi.it Address: Department of Computer Science, University of Pisa, Pisa, Italy. ORCID ID: https://orcid.org/0000-0001-8541-0284*

1. Introduction

Research is constantly in search of more efficient and powerful ways to support quality teaching and learning (Lang et al., 2017). The rising use of learning management systems (LMSs), massive open online courses (MOOCs), and other online

learning environments (OLEs), which are considerably increasing the availability of significant volumes of data, is leveraging the opportunity to inspect learners' behaviours during their online learning. Since the main challenge is to understand how to "translate" available data into valuable knowledge to provide insights, the advancement of distance learning favours more in-depth monitoring to establish its effectiveness in educational environments (de Oliveira et al., 2016; Alario-Hoyos et al., 2020).

The definition of learning analytics (LA) as "the measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs" (Siemens & Long, 2011) highlights that the measurement of educational data is the first challenge to face in deploying LA techniques. Any measurement activity must begin by defining the item to be measured, followed by selecting a measuring device and evaluating the measurement results (Pedhazur & Pedhazur Schmelkin, 1991). Measuring students' behaviour during active learning is difficult because it requires knowledge of their prior experiences, cognitive ability, personality traits, emotional states, attitudes, and skills. These *constructs*, namely concepts that have theoretical significance and can be abstract or not immediately observable (Calder et al., 2021), are difficult to measure, and they are typically measured via tests, quizzes, questionnaires, and surveys (Bergner, 2017). Online construct measurement is frequently based on log files that record the sequence of user actions while interacting in OLEs. To extract meaningful information from educational data, enabling knowledge construction and supporting decision-making, several authors use data mining methods to identify patterns in student behaviour (Kularbphettong, 2018; Wang et al., 2019; Gushchina & Ochepovsky, 2020; Akhuseyinoglu & Brusilovsky, 2022; Rotelli et al., 2022), building specific analytical models for their goals or case studies. Although a model cannot fully explain the complexity of the learning process, a model-based representation of user interactions can capture the key evidence and contribute to discovering some features that would otherwise go unnoticed and unexplored in a traditional setting. These authors investigate features such as resource usage, action frequency, average latency, login frequency, number of module accesses, login time, login regularity, total studying time, regularity of learning interval, and the type and sequence of actions carried out (Bernardini & Conati, 2010; Conijn et al., 2017; Cooley et al., 1999; Darlington, 2017; Siemens & Long, 2011; Sael et al., 2013; Bovo et al., 2013; Yu & Jo, 2014; Fincham et al., 2019; Poon et al., 2017).

Since the greater timing flexibility of online environments may affect learning differently, temporal analysis has been demonstrated to be relevant in LA research (Riel et al., 2018). To understand and monitor the timing strategies and behaviours that learners adopt in their study, analyzing student temporal learning behaviour may yield insightful information. As a result, numerous studies have focused on capturing time-on-task as a proxy to model learning behaviour, predict performance, and prevent dropout (Knight et al., 2017). Event logs describe something that occurred on a platform, and their temporal information, the Unix timestamps, indicates the time at which an event occurred. Usually, the duration of an event is calculated as the difference between two consecutive records, assuming that a log records the action starting time (Kovanović et al., 2016; Nguyen, 2020; Rotelli & Monreale, 2022). However, in Moodle, one of the most used LMSs, logs are recorded at different moments of the user actions, not only at their start. The correct interpretation of log data could present some complexities. Thus, the risk of misinterpreting the learning log data and, consequently, making incorrect choices in log data processing and training machine learning algorithms on biased datasets could lead to erroneous investigations of the learning process. In this scenario, knowing the context is the first step in obtaining meaningful information from log data.

Given these premises, this work intends to present Moodle's logging system to illustrate where the temporal dimension of Moodle log data can be difficult to interpret and how this knowledge can be used to improve data processing. To prevent the loss of data-related knowledge, we must ensure that the captured data precisely reflect the temporal behaviour we wish to capture. To this aim, we first provide an overview of Moodle log data generation (Section 2.1) and some suggestions for collecting them correctly (Section 2.2). Then, we present an in-depth study of temporal information of log data and the pitfalls they hide, as well as possible solutions to overcome them (Section 3).

2. Moodle Log Data

Moodle is a widely used LMS that supports educators in the creation and management of knowledge, providing two main kinds of learning and teaching modules: *Resources*[1], items that can be browsed online or downloaded to be used offline, and *Activities*[2], more advanced modules that involve student interaction. Moreover, Moodle includes communication tools (such as *Messaging*[3] and *Forums*[4]) to promote active involvement in the learning process and within the learning community.

[1]MoodleDocs—Resources—https://docs.moodle.org/39/en/Resources.
[2]MoodleDocs—Activities—https://docs.moodle.org/39/en/Activities.
[3]MoodleDocs—Messaging—https://docs.moodle.org/39/en/Messaging.
[4]MoodleDocs—Forum activity—https://docs.moodle.org/39/en/Forum_activity.

2.1 Moodle Log Data Generation

Moodle stores user activity in a relational database. Stored data logs, converted into a sortable, filterable, and human-readable tabular format, can be generated according to different needs by easily selecting a combination of fields in the *log generation interface*[5] depicted in Figure 1:

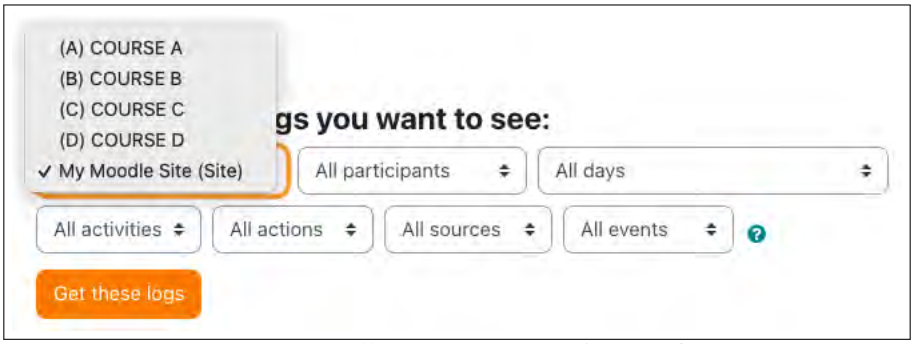

Figure 1. Moodle Log Data Generation Interface

- *Group:* The overall site or a specific course. Logs pertaining to a specific course can be generated by selecting the name of the course in the field *group* (*Course A*, . . . , *Course D*, in Figure 1). Courses are the areas in which teachers can add modules—resources and activities—for students to access[6]. All user actions carried out on the courses' modules are logged at the *course* level. However, being online does not necessarily imply participating in learning activities because users can browse the online environment without accessing any course. At the *site* level (by selecting the name of the Moodle site instance—*My Moodle Site* in Figure 1), it is possible to generate all of the log data for any action taken on the platform, both inside and outside of any course. Thus, the site logs include not only logs for individual courses but also logs for actions involving *user-user* relationships or the communicative interactions between the members of the learning community (for example, message sending and reading[3]) or student self-awareness, such as the non-course-specific grading overview and other aspects of the user profile[7].
- *Participants:* All participants or a specific participant. This field's items vary depending on the selected group. In Moodle, when users log in, the role of *authenticated user* is automatically assigned to them. The list will contain every authenticated user if the selected group is the overall site. Along with the authenticated user role, a user can have additional roles, depending on the context[8], namely the space where roles are assigned, such as student in Course A or teacher/tutor in Course B. If the selected group is Course C, a student enrolled in Course A will not appear in the list.
- *Date:* All days or a specific day. This field and the subsequent ones are filtered based on previous selections.
- *Activities:* All activities or a specific activity. For instance, an activity can be either the *Forum* on the homepage of the site or the *Assignment*, *Lesson*, *Quiz*, *Wiki* on the course.
- *Actions:* All actions or separated into *create*, *view*, *update*, *delete*, or *all changes*.
- *Sources: CLI:* Events triggered by a CLI script unrelated to user action[9]; *Restore:* typically automatically generated events related to role permissions[10], forum subscription[11], and grade items creation[12]; *Web:* Computer-based online user activities; *Web service:* User activities via mobile app.
- *Events:* All events or separated into *teaching*, *participating*, and *other*. *Teaching:* Actions usually performed by teachers that affect the student learning experience (for instance, grading an assignment or creating/updating a module); *Participating:* Actions related to course participant (student, teacher, or others) experiences, such as the user-artefact relationship, namely the user-learning resource connection (for example, module view, posting to a forum, or attempting a quiz); *Other:* Actions that include, among others, the user-user relationship (such as profile views), or authentication in the platform.

[5]MoodleDocs—Logs—https://docs.moodle.org/39/en/Logs.
[6]Moodle Docs—Managing a Moodle course—https://docs.moodle.org/39/en/Managing_a_Moodle_course.
[7]MoodleDocs—User profiles—https://docs.moodle.org/39/en/User_profiles.
[8]MoodleDocs—Context—https://docs.moodle.org/39/en/Context.
[9]MoodleDocs—Administration via command line—https://docs.moodle.org/39/en/Administration_via_command_line.
[10]MoodleDocs—Assign roles—https://docs.moodle.org/39/en/Assign_roles.
[11]MoodleDocs—Using Forum—https://docs.moodle.org/39/en/Using_Forum.
[12]MoodleDocs—Grade items—https://docs.moodle.org/39/en/Grade_items

The generated default *Standard Logs*[13] table, which can be displayed or downloaded in many file formats[5], is very detailed and supplied with nine features described in the following.

- **Time:** The system sequential time in a standard format (yyyy-MM-dd 'T' HH:mm), which has a one-minute granularity[14].
- **User full name:** The user performing the action.
- **Affected user:** The eventual recipient of the action (for instance, the user who receives a message).
- **Event context:** The context within the platform[8].
- **Component:** The module type (for example, *Wiki*, *Page*, *File*, *Url*, *Quiz*[2]).
- **Event name:** The type of action performed on the module (such as *viewed*, *deleted*, *updated*, *created*, and *submitted*).
- **Description:** The description of the event.
- **Origin:** The selected *sources* and the **IP address**.

Table 1 displays a typical log data table in the Moodle platform. Usernames are replaced with IDs for privacy concerns.

Table 1. Log Data Generated in Moodle

Time	User full name	Affected user	Event context	Component	Event name	Description	Origin	IP address
19 Jan 21, 17:35	User 59	-	Lesson: SQL	Lesson	Course module viewed	The user with id "59" viewed the "lesson" activity with course module id "156."	web	154.23.39.193
19 Jan 21, 17:35	User 59	-	Lesson: SQL	Lesson	Question answered	The user with id "59" has answered the True/false question with id "264" in the lesson activity with course module id "156."	web	154.23.39.193
19 Jan 21, 17:34	User 59	-	System	System	Group message sent	The user with id "59" sent a message with id "374" to the conversation with id "92."	web	154.23.39.193
19 Jan 21, 17:33	User 59	User 62	User 59	System	Message viewed	The user with id "59" read a message from the user with id "62."	web	154.23.39.193
19 Jan 21, 17:33	User 59	-	Lesson: SQL	Lesson	Question viewed	The user with id "59" has viewed the True/false question with id "264" in the lesson activity with course module id "156."	web	154.23.39.193

2.2 Moodle Log Data Collection

Being online does not necessarily mean engaging in learning activities. Users can access and leave a course many times, both navigating the online environment (without accessing any course) and, if they are enrolled in more than one course, accessing other courses. All actions related to the online environment's browsing (for example, viewing the *Dashboard*[15]), as well as the *user-user* relationship (for instance, sending and reading messages via the platform Messaging[3] system rather than the *Chat* module[16] that must be added to a course), or involving student self-awareness (that is, the grading overview and other aspects related to the user profile[7]) are not logged at the *course* level (*group:* a specific course), but at the *site* level (*group:* the overall site). Consider the following scenario. A user launches two browser tabs, one for Course A and one for Course B. After performing some actions on Course A, the user starts messaging, then continues on Course A with further actions. The logs record the following sequence: Course A $\langle event, event, event \rangle$, site $\langle messaging, messaging \rangle$, Course A $\langle event, event \rangle$. In Moodle, users can send messages to their contacts, to anyone in their courses, or to the members of the groups they belong to (if any)[3]. This means that a user can initiate interaction with a contact who is not necessarily enrolled in the same course. Because it is possible to communicate with contacts who are not necessarily enrolled in Course A, we cannot assert that these

[13] MoodleDocs—Logging—https://docs.moodle.org/39/en/Logging.
[14] Starting from Moodle 4.0 the granularity is per second and the standard format is (yyyy-MM-dd 'T' HH:mm:ss).
[15] MoodleDocs—Dashboard—https://docs.moodle.org/39/en/Dashboard.
[16] MoodleDocs—Chat activity—https://docs.moodle.org/39/en/Chat_activity.

messaging activities are associated with this course. Furthermore, if the user decides to move to Course B and this is already open in another tab, the user can access Course B without generating additional logs. Supposing that the following user action is clicking on the speech bubble icon on Course B rather than anywhere else, the logs will record the following sequence: site $\langle messaging, messaging \rangle$ without giving us any indication that the user switched the course. As a consequence, only if the user performs some actions on Course B will these actions be logged as Course B $\langle event, event, \ldots \rangle$. Since the nature of the messaging system and the peculiarities of logging (messaging logs are recorded at the *site* level) prevent us from stating that the user is engaged in any course-related learning activity, assigning the messaging actions to Course A or B, as has been attempted in prior efforts (Knobbout et al., 2019), would be conceptually incorrect. Moreover, if the logs are generated at the *course* level, these messaging logs are not visible (Section 2.1). If the logs are generated at the *site*, inspecting the sequence of logs, it appears as though the user has left the course, but they never actually did.

As a result, since the collection of log data only at the *course* level (*group:* a specific course) may cause some crucial information about student activity to be missed, it prevents complete comprehension of what the student is doing. For instance, while browsing a particular area of the course, a message from another participant instructing them to review a document could influence their behaviour. Unfortunately, a number of works only use course logs (Joksimović et al., 2015; Kovanović et al., 2016; Cerezo et al., 2016; Saqr et al., 2018; Villalobos et al., 2022). Instead, when investigating student behaviour, although the investigation is limited to a single course, the decisions made by students may be influenced by events outside of the course.

This is even more evident when focusing on the temporal dimension of student behaviour. Since the actual sequence of actions performed would be incomplete, the time spent on a course may be improperly calculated. As an example, we refer to a dataset of log data describing students' interactions on several courses (*Course A*, ..., *Course D*) in the following, organized on a Moodle platform (version 3.9.4+) of a postgraduate master's program. In Figure 2, the yellow records correspond to Course B, the greens correspond to Course A, and the reds represent various actions taken on the platform that are unrelated to any course.

ID	Time	User full name	Affected user	Event context	Component	Event name	Description	Origin	IP address
89	15 January 2021, 5:33 PM	USER 59	-	Course: B	System	Course viewed	The user with id '59' viewed the course with id '2'.	web	87.15.222.193
90	15 January 2021, 5:33 PM	USER 59	USER 59	Attendance: OpenLab	Attendance	Session report viewed	User with id 59 viewed attendance sessions for student with id 59	web	87.15.222.193
91	15 January 2021, 5:35 PM	USER 59	USER 59	User: USER 59	System	Dashboard viewed	The user with id '59' has viewed their dashboard	web	87.15.222.193
92	15 January 2021, 5:42 PM	USER 59	USER 59	User: USER 59	System	Dashboard viewed	The user with id '59' has viewed their dashboard	web	87.15.222.193
93	15 January 2021, 6:02 PM	USER 59	-	Course: A	System	Course viewed	The user with id '59' viewed the course with id '21'.	web	87.15.222.193
94	15 January 2021, 6:02 PM	USER 59	-	H5P: Answer the questions	H5P	Course module viewed	The user with id '59' viewed the 'h5pactivity' activity with course module id '36'.	web	87.15.222.193
95	15 January 2021, 6:02 PM	USER 59	-	H5P: Answer the questions	H5P Package	H5P content viewed	The user with id '59' has viewed the H5P with the id '28'.	web	87.15.222.193
96	15 January 2021, 6:04 PM	USER 59	-	H5P: Answer the questions	H5P	xAPI statement received	The user with the id '59' send a tracking statement for a H5P activity with the course module id '36'.	web	87.15.222.193
97	15 January 2021, 6:04 PM	USER 59	-	Course: A	System	Course viewed	The user with id '59' viewed the course with id '21'.	web	87.15.222.193
98	15 January 2021, 6:05 PM	USER 59	-	Category: Overall Site	System	Category viewed	The user with id '59' viewed the course category with id '2'.	web	87.15.222.193
99	15 January 2021, 6:05 PM	USER 59	-	Course: B	System	Course viewed	The user with id '59' viewed the course with id '2'.	web	87.15.222.193
100	15 January 2021, 6:05 PM	USER 59	-	File: Intro 01	File	Course module viewed	The user with id '59' viewed the 'resource' activity with course module id '245'.	web	87.15.222.193

Figure 2. Site Logs

When generating logs at the *course* level, and being interested in Course B (*group:* Course B), only records with the IDs 89, 90, 99, and 100 are extracted. If we are interested in determining the time spent in the course, we have to sum up the duration of all the activities carried out. Since the duration of an event is usually calculated as the difference between the timestamps of two consecutive logs (Kovanović et al., 2016; Nguyen, 2020; Rotelli & Monreale, 2022), it would appear as if User 59 is viewing the attendance session report for 32 minutes (ID 99 − ID 90). Instead, the user is engaged elsewhere but not in the course in question (IDs 91 to 98). Therefore, collecting data at the *course level* excludes these actions, and it seems like the user spends more time than expected on specific resources or activities.

The analysis of the activity duration distributions of logs highlights how the extraction of logs at the *course* level could affect the results. Figure 3 shows the duration in seconds of the Course B logs. The duration is calculated as the difference in time between an activity timestamp (t_i) and that of the next activity in the sequence (t_{i+1}), that is, $d_i = t_{i+1} - t_i$. The chart on the left displays the duration of logs extracted at the *course* level; namely, in the *group* field of the log generation interface, Course B has been selected. The right chart represents logs of Course B extracted at the *site* level; namely, the overall site has been selected in the *group* field of the log generation interface. Then, after calculating the duration of each record, only logs for Course B have been selected to be included in the chart. It is worth noting that the total time spent of logs extracted at the *course* level is more than three times as much as for the same logs extracted at the *site* level.

Nonetheless, it should be taken into account that some events may still be characterized by an exceptionally long duration. In fact, we cannot assume that the total time students spend online is entirely dedicated to interacting on the platform. While connected, they can also be inactive, take breaks, or browse other websites; that is, they can be *off-task*. Thus, although our methodology has a significant impact on time-on-task, outliers should be dealt with before undertaking any temporal analysis (Rotelli & Monreale, 2022).

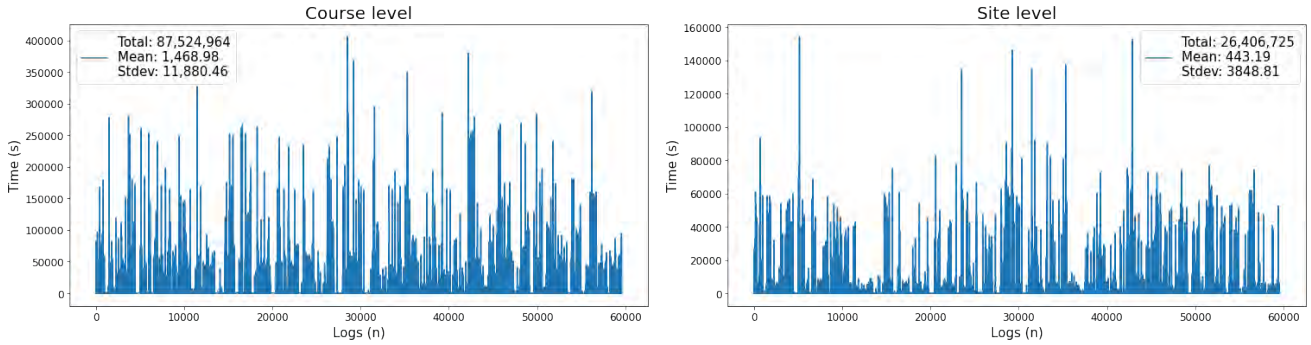

Figure 3. Duration Distribution of Logs of Course B Extracted at the Course and at the Site Level, with Duration Expressed in Seconds

On the other hand, the collection of log data only at the *site* level (*group:* the overall site and not a specific course) has some drawbacks. Because it returns the logs of all actions performed on the platform (that is, the logs of the actions performed outside of any course, as well as the logs of all the courses, following the timestamp sequence), determining which logs correspond to which course is not an easy task. Although complete, the information about the course is missing in the generated log table. Records of Figure 2 are coloured to make it clearer, but records of Figure 1 are indistinguishable. At first glance, it may appear that accessing the homepage will aid in identifying the course by scrolling through the log sequence. Nevertheless, users could browse modules of different courses by clicking a link (for example, an email sent by the teacher) or by opening multiple browser tabs, without accessing the homepage of the course.

In order to address this issue, after collecting all data at the site level, in Knobbout and colleagues (2019), the authors proposed to identify the course the student was working on by extracting the information about the course ID from the *Description* field (for example, *The user with ID "78" viewed the* ***course with ID*** *"3"*). Nevertheless, descriptions of logs pertaining to module activities provide no information about the course in which they occur. The only derivable information is the IDs identifying the user and the module (for example, *The user with ID "36" viewed the "lesson" activity with* ***course module ID*** *"184"*). As a consequence, to achieve the goal, it is necessary to extract the list of module IDs associated with each course directly from the Moodle database and to use regular expressions to match with the IDs of the description gathered. However, not all descriptions include the module ID, preventing the correct identification of the corresponding course (for example, *The user with ID "68" viewed the user report in the gradebook*). In addition, the description presents some irregularities introduced by the formatting of database log reporting[17]. Table 2 depicts some examples of conversion issues from raw data.

Table 2. Conversion Issues from Raw Data

ID	Description
1a	The user with ID "18" viewed the "**assign**" activity with course module ID "708."
1b	The user with ID "18" has viewed the submission status page for the **assignment** with course module ID "708."
2a	The user with ID "52" updated the completion state for **the course module with ID** "107" for the user with ID "52."
2b	The user with ID "52" has added the option with ID "6" for the user with ID "52" from the choice activity **with course module ID** "107."
3a	The user with ID "47" viewed **the "chat" activity** with course module ID "146."
3b	The user with ID "52" has sent a message in **the chat** with course module ID "146."

Case 1: The "assign" activity of description *a* becomes *assignment* in description *b*. The same type of activity is stored in the database table *mdl_modules* with two different names and IDs (ID 1: assign; ID 2: assignment).

[17] MoodleDev—Events API—https://docs.moodle.org/dev/Events_API.

Case 2: The *course module with ID* of description *a* becomes *with course module ID* in description *b*.

Case 3: The *"chat" activity* in description *a* becomes *chat* in description *b*.

Additionally, a problem arises when a module is deleted from a course: its ID is removed from the database, not from the logs. This implies that the matching will not take place. To address deleted activity logs, a solution could consist of removing, before matching, those records where the "Event context" is *Other* because they represent actions performed on deleted modules. Hence, the use of the *Description* field to identify the course is not trivial and should consider all potential conversion issue scenarios, as well as the possibility that the module ID can be missing.

Given these considerations, we propose a data collection methodology that helps to overcome these issues. To provide a comprehensive picture of user activity on the platform, eliminate temporal gaps, and fully comprehend user behaviour without information loss, we recommend extracting logs in two stages:

1. **Site logs:** Select the overall site in the *group* field of the log generation interface (Section 2.1) to extract all logs at the *site* level.
2. **Database data:** Extracted Moodle logs (Figure 1) contain no information about the place where actions were carried out. To differentiate between actions performed on the platform and actions performed on courses, the extracted logs must be supplemented with the course name. This information can be extracted directly from the database[18]. Moreover, although the system uses Unix timestamps, the outcome of extracting logs from Moodle contains timestamped events with a granularity of one minute. Since a large number of actions can be performed in a minute, to get time information with a granularity of one second, the timestamps must be taken directly from the database and added as a new field to the log table. The timestamp values could then be converted in a Date&Time standard format (yyyy-MM-dd 'T' HH:mm:ss) to have a human-readable time format[19].

Database data and *site logs* tables are then joined into a single dataset to maintain all logs relevant to user activities without information loss. When the data are combined, all actions performed in the courses are immediately detected. The integrated table of log data (Table 3) keeps track of the sequences of all user interactions and describes user activity sequences that occur on the platform, both at the *course* level and at the *site* level. Sequential data consist of a set of attributes (columns) and a set of records (rows). Each record, which stores information about a particular user action, is an object identified by an ID and described by the values of the table's attributes, as depicted in Table 3: the *Time* representing the timestamp of the event, the *Course_Area* identifying the course for *course* logs and presenting a missing value for *site* logs, the *Date&Time*, the *user* performing the action, the possible *recipient*, the *context* of the event[20], the *component* indicating the learning module type, the *event name* and *description* of the event, and the originating *IP address*. Missing Course_Area values for site-level logs might be easily replaced with a value indicating the overall platform or a category specifying a particular portion of the platform as proposed in Rotelli and Monreale (2022). We made the entire source code and a detailed description of the steps available on GitHub[21].

Table 3. Integrated Log Record

ID	Time	Course_Area	Date&Time	User full name	Affected user
402267	1615214006	Course B	2021-03-08T15:33:26	USER 37	–
Event context	**Component**	**Event name**	**Description**	**Origin**	**IP address**
Lesson: Introduction	Lesson	Question viewed	The user with ID "37" has viewed the Numerical question with ID "484" in the lesson activity with course module ID "641."	web	109.52.45.53

It is important to note that collected records can mask some pitfalls. In an attempt to illustrate the point as clearly as possible, in the following section, we provide several examples and some recommendations based on the results of our investigation. From now on, we will use the word *student* instead of *user* when an action can be performed only by the student role[22].

[18]Moodle stores user activity in a relational database (https://moodleschema.zoola.io/). To access the database, it is also possible to employ the *Configurable reports* plugin (https://moodle.org/plugins/block_configurable_reports), which enables the execution of a custom SQL query.

[19]Note that starting from Moodle 4.0, Moodle interface logs are provided with a one-second granularity. However, by extracting this information from the database, the conversion is avoided and it can be used immediately to calculate duration.

[20]MoodleDocs—Context—https://docs.moodle.org/39/en/Context. Note that in Moodle, the context is the space where roles can be assigned; it is not a learning module.

[21]Moodle-Log-Data-Consolidation: https://github.com/AD1529/Moodle-Log-Data-Consolidation.

[22]MoodleDocs—Student role—https://docs.moodle.org/39/en/Student_role.

3. The Temporal Dimension

Time is a fundamental aspect to consider when modelling student behaviour, and the analysis and interpretation of the actions cannot fail to pass through the analysis of the time taken to perform them. However, the analysis of time is a very tricky task faced in a number of works (Kapusta et al., 2012; Kovanović et al., 2016; Chen et al., 2018; Lee, 2018; Knobbout et al., 2019; Dermy & Brun, 2020; Rotelli & Monreale, 2022; Rotelli et al., 2022), which revealed some challenges that still need to be addressed.

Digital traces of learner interactions are collected in sequence, and each record is supplied with temporal information, a timestamp that typically identifies the Unix time of the action performed. Log generation from the Moodle interface converts the Unix time to a human-readable format (for instance, 27 April 2022, 7:21 PM) with a minute-time granularity[19]. For our purposes, since a large number of actions could be performed in a minute, reducing our understanding, we extracted the Unix time directly from the database to get a granularity of one second (Section 2.2).

Usually, the duration of an event is calculated as the difference between the timestamps of two consecutive records, assuming that a log records the action's starting time (Kovanović et al., 2016; Nguyen, 2020; Rotelli & Monreale, 2022). Yet, in our investigation, based on trials, simulations, and log data analysis, we found that Moodle records learning activities at different moments of user action. While most events are logged at their beginning (for example, accessing a file), other events are recorded at their end, and we do not know their exact starting time (for example, writing some content and posting it on a forum). Furthermore, although event logs typically record events involving a server interaction, rich client-side interfaces allow for many local operations without the need to request additional information from the server[23]. As a result, despite improving the user experience (in terms of responsiveness and bandwidth usage), they hide some local user activities, preventing the recording of some essential and valuable clicks that can be used to investigate student behaviour.

To prevent the loss of data-related knowledge, we must ensure that the collected data accurately reflect the temporal behaviour we wish to capture. Since Moodle records learning activities at different moments of user action, to cover all possible scenarios, we classify learning activities into five event categories:

- **Starting:** Actions that are recorded at their beginning, where the timestamp marks their starting time; this is the case with the majority of "view" (for example, *Course viewed*, *Discussion viewed*, *Submission form viewed*), as depicted in Table 4. Since a starting event ends when a new event begins, its duration is the difference between its timestamp and the timestamp of the next event in the sequence ($t_{i+1} - t_i = d_i$).

Table 4. Starting Events

T	Date&Time	Difference	Event name	Category	Computation	Duration
0	2021-01-20T17:58:36	4	Course viewed	Starting	$d_i = t_{i+1} - t_i$	4
1	2021-01-20T17:58:40	12	Session report viewed	Starting	$d_i = t_{i+1} - t_i$	12
2	2021-01-20T17:58:52	3	Course viewed	Starting	$d_i = t_{i+1} - t_i$	3
3	2021-01-20T17:59:55	37	Course module viewed	Starting	$d_i = t_{i+1} - t_i$	37

- **Ending:** Actions that are recorded when they are completed, whose timestamp indicates their ending time. These events, which usually imply writing some content (for example, *Comment created*, *Some content has been posted*, *Message sent*), are always preceded by a *starting* event that usually implies reading some content. Unfortunately, we do not know precisely when the *starting* event ends and the *ending* event starts. Let us consider a user who starts viewing a discussion thread in a forum, as depicted in Table 5. The log *Discussion viewed* is recorded, marking the beginning of the *starting* event. After a certain amount of time has passed (there can be only one or multiple posts to read), the user decides to click on the "Reply" button to write some content. Because the rich client-side interface allows this operation without any server request, no log is recorded. After completing their writing, the user clicks on the "Post to forum" button, and the event *Some content has been posted* is logged (T_2). Since we do not know precisely when the action of writing started, we cannot derive its duration. It goes without saying that the difference between a *starting* and an *ending* event identifies their sum, not the duration of the first ($t_{i+1} - t_i = d_i + d_{i+1}$). Since we cannot determine with certainty how long the two events last separately, we can assume that they last the same amount of time on average or we can also attempt to infer the duration through a more in-depth examination of the read and written content. For instance, the estimated reading and writing time of a post could be based on the sentence complexity or the number of posts that were published before the user accessed the forum and replied.

[23] MoodleDev—AJAX—https://docs.moodle.org/dev/AJAX.

Table 5. Ending Events

T	Date&Time	Difference	Event name	Category	Computation	Duration
0	2021-01-13T16:34:52	2	Course module viewed	Starting	$d_i = t_{i+1} - t_i$	2
1	2021-01-13T16:34:54	97	Discussion viewed	Starting	-	-
2	2021-01-13T16:36:31	0	Some content has been posted	Ending	$d_i + d_{i+1} = t_{i+1} - t_i$	97

- **Simultaneous:** Logs that do not correspond to direct user actions but are automatically generated by the system while the user performs a specific action. Let us take as an example an Assignment activity[24] that collects submissions from students as uploaded files (Table 6). At time T_0, the log records the student accessing the assignment. At time T_1, one more entry records the status of the submission as "viewed." These two logs are concurrent ($T_0 = T_1$) because they happen at the same time: the student accesses the assignment page, the module description is displayed on top (T_0), and the status of the submission is shown at the bottom (T_1). At time T_2, the student clicks on the "Add submission" link, a new request to the server is sent, but the log records the same event as $T_2 = T_0$; this occurs because the top of the loaded page remains unchanged, while the form to submit a file (T_3) has replaced the submission status at the bottom of the page (T_1). When the file is uploaded[25], five concurrent events ($T_4 = T_5 = T_6 = T_7 = T_8$) are recorded: event T_4 indicates that the server request steadily loads the same top page ($T_4 = T_2 = T_0$); T_5 records the uploading of the file; and, concurrently, the submission creation is logged (T_6). Moreover, since the student is redirected to the initial page, events $T_7 = T_0$ and $T_8 = T_1$ are logged: the top page shows the module description (T_7) and the bottom the updated submission status (T_8). Simultaneous activities necessarily have the same timestamp, and their duration, calculated as the difference between two consecutive records, is clearly 0 ($d_i = 0$).

Table 6. Simultaneous Events

T	Date&Time	Difference	Event name	Category	Computation	Duration
0	2022-03-12T18:12:20	0	Course module viewed	Simultaneous	$d_i = 0$	0
1	2022-03-12T18:12:20	3	The status of the submission has been viewed	Starting	$d_i = t_{i+1} - t_i$	3
2	2022-03-12T18:12:23	0	Course module viewed	Simultaneous	$d_i = 0$	0
3	2022-03-12T18:12:23	6	Submission form viewed	Starting	$d_i = t_{i+1} - t_i$	6
4	2022-03-12T18:12:29	0	Course module viewed	Simultaneous	$d_i = 0$	0
5	2022-03-12T18:12:29	0	A file has been uploaded	Simultaneous	$d_i = 0$	0
6	2022-03-12T18:12:29	0	Submission created	Ending[26]	$d_i = 0$	0
7	2022-03-12T18:12:29	0	Course module viewed	Simultaneous	$d_i = 0$	0
8	2022-03-12T18:12:29	5	The status of the submission has been viewed	Starting	$d_i = t_{i+1} - t_i$	5

To understand these event logs in depth, we replicated the assignment module and repeated the experiment many times by changing the settings to analyze the differences (some examples are provided in Appendix B). We noticed that the sequence and the number of logs, as well as the number of simultaneous events, depend not only on the actions performed at the student side (that is, edit submission, remove submission, add comment) but also on the module settings at the teacher side (such as accepting both online text and file submission or requiring the student to click the "Submit" button and accepting the submission statement)[27]. Therefore, our results show that while measuring the number of actions performed on modules of different courses to analyze student behaviour on tasks or the time spent on them (the number of actions required from the student for submission lengthens the time-on-task), all of these factors should be taken into account to avoid biased results. Aggregating logs or an in-depth cleaning activity could be reasonable solutions. It is worth noting that sometimes the duration of these actions could be 1. This happens because we are utilizing a one-second granularity Unix timestamp value. Looking at the timestamps with a granularity per millisecond, they are not truly simultaneous. As a result, it is possible for one action to be recorded in the millisecond of one second and another in the next second. To provide an estimate of the number of concurrent occurrences on a Unix time basis, only 127,637 of the 181,803 events we examined do not have a concurrent event.

- **Opening & closing:** Actions whose duration started with an *opening* event and ended with a *closing* event (for example, *Question answered*, *Survey response submitted*, *Feedback Response submitted*). Let us consider a Lesson[28] (Table 7),

[24] MoodleDocs—Assignment activity—https://docs.moodle.org/39/en/Assignment_activity.

[25] A file can be uploaded via the "File picker" (https://docs.moodle.org/39/en/File_picker) or via the "drag and drop." The event name and the sequence of logs does not change.

[26] To be categorized as an "Ending" event, this event must be moved and placed before the "course module viewed" event.

[27] The database tables contain information regarding the configuration settings (some examples are provided in the footnotes of Appendix A). For the Assignment, the settings are recorded in the *mdl_assign* table.

[28] MoodleDocs—Lesson activity—https://docs.moodle.org/39/en/Lesson_activity.

where students can view some page content or are requested to answer some questions[29]. Every time a student visualizes a page or a question, two events with the same timestamp are recorded: the simultaneous event *Course module viewed* (T_0, T_2) and the starting event *Content page viewed* (T_1) or the opening event *Question viewed* (T_3). In this latter case, when the student presses the "Submit" button to answer, the log records the *closing* action *Question answered* (T_4). We can assume that the act of answering is intertwined with the act of reading the question; that is, answering cannot avoid reading the question. Therefore, in contrast to an *ending* event, the beginning of which is unknown because the user might choose to read or not read existing posts, in *closing* events, the beginning is known: the *opening* event. With respect to a *starting* event, an *opening* event doesn't have a real duration since it is tied to the *closing* event. The duration of a *closing* event is calculated as the difference between its timestamp and that of the previous event in the sequence ($t_{i+1} - t_i = d_{i+1}$). It is worth noting that the act of answering (when the *Question answered* log is recorded) initiates a new event: the text of the question and the response to the answer (whether correct or incorrect) are displayed so that the student can continue to examine the question and the given response. This time, which is not spent answering but checking the answer or reading feedback, if any, is a starting action interrupted when the student presses the "Continue" button. To clarify, Table 7 depicts a sequence of logs within a Lesson. Since we assume that a student must first read a

Table 7. Opening & Closing Events

T	Date&Time	Difference	Event name	Category	Computation	Duration
0	2021-01-19T08:23:58	0	Course module viewed	Simultaneous	$d_i = 0$	0
1	2021-01-19T08:23:58	76	Content page viewed	Starting	$d_i = t_{i+1} - t_i$	76
2	2021-01-19T08:25:14	0	Course module viewed	Simultaneous	$d_i = 0$	0
3	2021-01-19T08:25:14	**21**	**Question viewed**	Opening	-	-
4	2021-01-19T08:25:35	**13**	Question answered	Closing	$d_{i+1} = t_{i+1} - t_i$	**21**
5	–	–	**Question summary viewed**	Starting	$d_i = t_{i+1} - t_i$	**13**
6	2021-01-19T08:25:48	0	Course module viewed	Simultaneous	$d_i = 0$	0
7	2021-01-19T08:25:48	367	Content page viewed	Starting	$d_i = t_{i+1} - t_i$	367

question before answering it, we did not consider the *Question viewed* event. Instead, we added the *Question summary viewed* event to represent the time spent after the completed action. We also stress a key point: if the student views the question but chooses not to respond, namely the *Question answered* log is missing, the *Question viewed* event should be considered a *starting* event.

- **Instantaneous:** Actions limited to a single click's duration (for example, *Zip archive of folder downloaded*, *Book/Chapter printed*). When the log recording interrupts the prior action, users are redirected to the page they were on before taking the instantaneous action. However, as suggested in Kovanović and colleagues (2016), this time period should be added to the action that came before the *instantaneous* event ($(t_{i+1} - t_i) + (t_i - t_{i-1}) = d_{i-1}$; $d_i = 0$). Let us consider a user who opens a Book[30] (Table 8) and reads some Book chapter (T_1, T_2), then decides to print the chapter by selecting the "Print this chapter" button: a pop-up opens to allow the user to print the chapter; the log *Chapter printed* is recorded (T_3), interrupting the preceding log's duration (T_2), but the user is still on the same chapter page. Thus, if we do not consider this aspect, any time the user remains on the chapter page (T_2) will likewise be "erroneously" attributed to the *Chapter printed* log[31] (T_3). It is worth noting that instantaneous events can also last 0 seconds if they occur concurrently with other events.

Table 8. Instantaneous Events

T	Date&Time	Difference	Event name	Category	Computation	Duration
0	2021-03-17T12:53:54	0	Course module viewed	Simultaneous	$d_i = 0$	0
1	2021-03-17T12:53:54	14	Chapter viewed	Starting	$d_i = t_{i+1} - t_i$	14
2	2021-03-17T12:54:08	13	Chapter viewed	Starting	$d_{i-1} = (t_{i+1} - t_i) + (t_i - t_{i-1})$	106
3	2021-03-17T12:54:21	93	Chapter printed	Instantaneous	$d_i = 0$	0
4	2021-03-17T12:55:44	11	Chapter viewed	Starting	$d_i = t_{i+1} - t_i$	11

[29] In a Lesson activity, Moodle allows a student to select among a number of question types: essay, matching, multichoice, numerical, short answer, true/false: Building lesson—https://docs.moodle.org/39/en/Building_Lesson.

[30] MoodleDocs—Book resource—https://docs.moodle.org/39/en/Book_resource.

[31] Determining the log's category is not always straightforward. If we consider the "search" on the forum threads' discussion board, the action *Course searched* lasts the period of a click and should be considered "instantaneous." This step, however, creates a new page with a list of threads containing the searched term(s). As a consequence, a new "starting" event is recorded. Note that this log should not be confused with *Courses searched*, whose component is *System*, indicating a search among all courses in the platform.

To illustrate the importance of these differences when calculating time-on-tasks, in Table 9 we report on some log sequences when posting, updating, and deleting some content on a Forum[32] to compare the time spent depending on the event category and the type of action performed.

Table 9. Log Sequences in a Forum

T	Date&Time	Event name	Category
1a	2021-01-08T20:40:13	**Course module** viewed	Starting
1b	2021-01-08T20:45:08	Some content has been posted	Ending
1c	2021-01-08T20:45:08	**Discussion** created	Simultaneous
1d	2021-01-08T20:45:08	Course module viewed	Starting
2a	2021-01-13T16:34:52	**Course module** viewed	Starting
2b	2021-01-13T16:34:54	**Discussion** viewed	Starting
2c	2021-01-13T16:36:31	Some content has been posted	Ending
2d	2021-01-13T16:36:31	**Post** created	Simultaneous
3a	2021-02-17T12:22:30	**Course module** viewed	Starting
3b	2021-02-17T12:22:35	Discussion viewed	Starting
3c	2021-02-17T12:23:01	Post **deleted**	Instantaneous
3d	2021-02-17T12:23:01	**Discussion** viewed	Starting
4a	2021-02-06T21:03:50	**Course module** viewed	Starting
4b	2021-02-06T21:03:55	Discussion viewed	Starting
4c	2021-02-06T21:05:03	Some content has been posted	Ending
4d	2021-02-06T21:05:03	Post **updated**	Simultaneous
4e	2021-02-06T21:05:03	**Discussion** viewed	Starting

Let us start with the creation of a new discussion. When the user enters the forum, the event *Course module viewed* is recorded at time $1a$[33]. After the user clicks the "Add a new discussion topic" button, an editor box opens, and the user starts writing (note that the rich client-side interface makes this action possible without a server request). At time $1b$, the button "Post to forum" is clicked, and the log records the event *Some content has been posted*, marking the end of the writing action. The user is then redirected to the general view of the forum, *Course module viewed*, which contains the list of all threads. At the very same time ($1b = 1c = 1d$), two more events are logged: the creation of the discussion and the module view due to the redirecting action.

The second example takes into account the creation of a new post in a discussion already created. The user enters the forum at time $2a$ with the list of all threads available, then selects the thread at time $2b$ and starts to read the posts (there may be one to many posts to read). After a period of time, which is unknown because it relies on the number of posts, because the user has already visited the forum, or the user has decided not to read the posts, the user clicks the "Reply" button, an editor box opens, and the user starts writing some text. Because the rich client-side interface enables this action without a server request, we cannot determine the precise time at which the user began writing. At time $2c$, the user clicks the "Post to forum" button, and the content is posted; at the same time ($2c = 2d$), one more log is recorded: the creation of the new post. In contrast to the prior example ($1c$), the user remains on the discussion page and is not redirected to the course module page with the list of all available discussions (like in $1d$). Remaining on the discussion page, the *Post created* log, although its name may lead to confusion, has to be considered a *starting* event that could be followed by other posts or any other activity. If no new posts are created, we can assume that the time following the *Post created* event is spent reading other posts or the user's posts. Indeed, after a post is added, Moodle grants 30 minutes to delete or edit it before sending a notification to the mailboxes of all subscribers. On the other hand, the user is also allowed to answer several posts in the same discussion.

The third example illustrates a sequence of actions performed to delete a post. The user enters the forum at time $3a$, then selects the discussion ($3b$) and starts reading. At a certain point, the user decides to delete a post by clicking on the "Delete" button[34]. A request to the server is sent, and a new record is logged ($3c$). Moreover, being instantaneous, this event causes the last activity to be interrupted while the user is instantly redirected ($3d$) to the same forum discussion as $3b$.

The fourth example shows a typical log sequence when a post is updated. The sequence is pretty much the same as the second example. After entering the forum ($4a$), the user selects a discussion ($4b$), then decides to update a post and clicks the "Edit" button to start updating it ($4c$, $4d$). However, unlike the second example (where the "Reply" button opens an editor box and this action is performed on the client side), an editing page is loaded to update a post, and the user leaves the discussion. After clicking the "Save changes" button, the user is redirected to the discussion; consequently, a new *Discussion viewed* log is

[32] The subscription mode option is set to "Auto subscription" to focus only on important occurrences and avoid irrelevant logs related to a forum subscription.
[33] In contrast to the Lesson or the Assignment, where the *Course module viewed* event is *simultaneous*, in the Forum, this is a *starting* event.
[34] A user can only delete or edit their own previously created post.

recorded (4*e*).

It is worth noting that only in the first example does the user spend all of their time writing content, whereas in the third example, the time is spent reading posts (clearly, the user may not read any posts, but no writing action is performed). In examples two and four, the user is involved in both reading and writing. Because we cannot differentiate between reading and writing time, we can only calculate the time spent reading and writing on the forum. We also stress a key point. The event *Some content has been posted* is logged without any explicit distinction before the events *Discussion created*, *Post created*, and *Post updated*, namely the records that inform about the type of action performed. Therefore, when counting the number of forum posts or calculating the time spent in different activities to model student behaviour (Macfadyen & Dawson, 2010), it should be considered that the real action will be clarified later.

To provide greater clarity, we first calculated the time spent on forums by *event* by summing the event duration estimated as the difference between two consecutive logs (Table 10). Then, we considered four different actions: *module view*, *reading*, *writing*, and *reading and writing*, and we compared the variation of action duration based on the following event categories: *starting*, *ending*, *opening*[35], *closing*, *simultaneous*, and *instantaneous* (Table 11). The comparison was performed on Course B, which consists of six forums, and only student logs were taken into account. The results of the computation are displayed in Table 10; no outlier detection methodology has been applied.

Table 10. Forums of Course B: Time Spent (in seconds) by Event Log

	Events						
Forum ID	**Course module viewed**	**Discussion viewed**	**Discussion created**	**Post created**	**Post updated**	**Some content has been posted**	**Forum Total**
1	7,685	122,303	0	0	0	0	129,988
2	5,835	197,106	0	76,559	0	0	279,500
3	63,972	132,590	0	41,223	0	0	237,785
4	321	0	0	0	0	0	321
5	1,165	12,643	0	16,026	0	0	29,834
6	716	1,762	0	1,611	0	0	4,089

Table 11. Forums of Course B: Time Spent (in seconds) by Action Calculated without and with Event Category

	Without event category					With event category			
Forum ID	**Module view**	**Reading**	**Writing**	**Reading & Writing**	**Forum Total**	**Module view**	**Reading**	**Writing**	**Reading & Writing**
1	7,685	122,303	0	0	129,988	7,685	122,303	0	0
2	5,835	273,575	0	0	279,500	5,771	268,734	64	4,920
3	63,972	173,813	0	0	237,785	62,607	172,231	1,365	1,582
4	321	0	0	0	321	321	0	0	0
5	1,165	28,669	0	0	29,834	663	27,558	502	1,111
6	716	3,373	0	0	4,089	469	3,150	247	223

The first forum is the "Annoucements" forum[36], where only teachers can post. Since students can only access the learning module containing the list of threads (*Course module viewed*) and read the discussions (*Discussion viewed*), the time spent in all other events is clearly 0. In the fourth forum, no discussion was created (*Discussion viewed* = 0), but students accessed the learning module, which is likely to verify the presence of some discussions. In forums 2, 3, 5, and 6, students read and wrote. In general, the event *Discussion viewed*, especially in forums 1, 2, and 3, takes up most of the time. In forums 2, 3, and 5 the students also invested considerable time in the *Post created* event.

On the other hand, if we consider the event categories, as in the examples in Table 9, and differentiate by type of action taken (*module view*, *reading*, *writing*, *reading & writing*), the results show a completely different picture. When students access a forum, the list of threads is displayed, the event *Course module viewed* is logged, and this action is considered a simple view of the forum module. If students enter a forum to read a thread, the log *Discussion viewed* is recorded. If they create a new discussion, this has to be considered as a writing action, but if they post some content in an existing discussion, the *Some content has been posted* log marks a reading and writing action. However, as previously stated (Table 9, ID 2), after posting to a forum, students still remain in the discussion. As a result, the time frame that starts with the *Post created* log must also be considered a reading action. As a result, if the event categories are not considered, all actions should be classified as *reading* actions since we can only calculate the time spent in all the events. Conversely, using the category overcomes this issue by differentiating the time spent in each event.

The values of Tables 10 and 11 are depicted in Figure 4. We observe that the majority of students' time was spent reading discussions. As previously mentioned, students access forum 1 (*module view*) only to read posts. Students devote little time to

[35] An *opening* event should be considered a *starting* event if it is not followed by a *closing* event.

[36] MoodleDocs—Annoucements forum—https://docs.moodle.org/39/en/Announcements_forum.

creating discussions in forum 2 (*writing*), although they reply a lot (*reading & writing*). Nevertheless, the majority of them accessed the forum to read posts. In forums 3, 5, and 6, students create new discussions and participate in equal measures. According to the event logs of forum 4, students accessed a module that was lacking discussions.

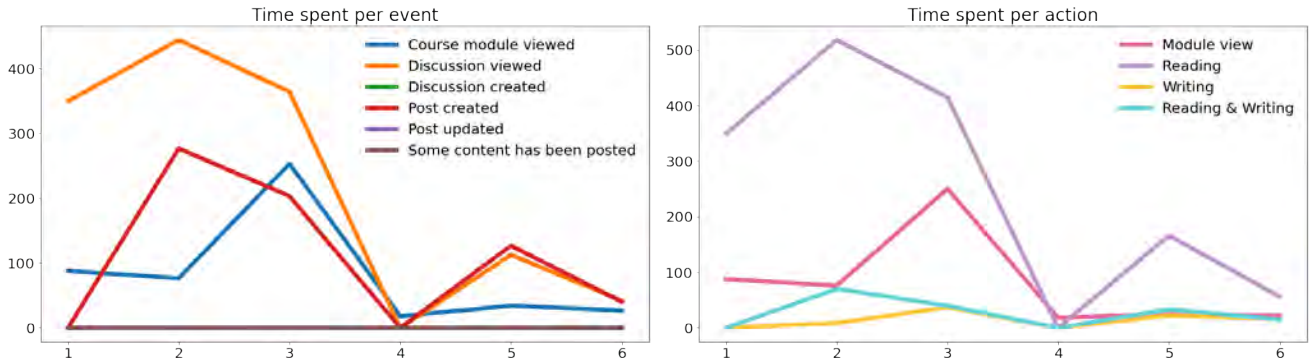

Figure 4. The Six Forums in Course B by Events and Actions; Due to the Significant Difference in Values, We Employed Square Values for a Better Representation

We would also like to stress a key point: depending on the module settings, the category of an event can change. For example, a File[37] can be downloaded or embedded to be browsed online within the course. The former event is *instantaneous*; the latter is *starting*. However, since Moodle makes no distinction between the two activities (the component is in both cases *File* and the event name *Course module viewed*), knowing the exact context the data is collected in is fundamental (Ridzuan & Zainon, 2019). To illustrate the point, in Course D of our dataset, we count 25 Files and 1165 accesses to these resources. Since all Files have been set to "Force download"[38], students had to save the file to open it locally rather than through the Moodle platform. Thus, the events being *instantaneous*, the time spent online on the File resources should be 0. On the contrary, the time spent is 3,098,209 seconds. This is most likely because students begin reading the resource locally after downloading it without logging out from the course. In Moodle, if users take no action (no requests to the server) during a specific time period set in the site administration settings (8 hours by default[39]), their session is ended, and they are logged out. Users can access the platform many times within this time window, even closing the browser. This means that one action can last up to 28,799 seconds (7 h 59 min 59 s) with no need to log in again. The use of event categories, a complete list of which is provided in Appendix A, could help overcome this issue. We made available on GitHub[40] the complete source code to assign categories and calculate event duration based on category.

4. Conclusion

This study aimed to show where Moodle log data can be difficult to comprehend and how to use this information to improve data processing. We investigated and demonstrated many sorts of log sequences and offered various factors to consider while preparing data for investigation and analysis of the temporal dimension. Biased datasets and, as a result, incorrect interpretation of study findings could result in incorrect knowledge and comprehension of data creation dynamics. Moreover, a more precise estimation of time could result in the display of more accurate dashboard values. Dashboards support both *teachers* in understanding and monitoring the types of timing strategies and behaviours that learners employ and *students* in monitoring, controlling, and acquiring awareness of their actual habits in relation to their perception.

Another major implication of this work is that the counts' measurements used in LA research may need to be reassessed. We sought to emphasize that a thorough understanding of the actions taken requires that all log fields be considered together. Because of the difficulties in detecting learner behaviours and the lack of a clear understanding of logs and activity settings, the use of count measurements may result in misinterpretations of student behaviour. This is especially true when comparing data from several courses.

We also want to bring out another important point, more geared toward web application developers, because these findings have substantial consequences for understanding how to treat data before any analysis. The smarter the client becomes, the fewer data are available to those who want to analyze them. Furthermore, clicking the back button to return to a previously visited page does not result in a log entry because the previously requested page is fetched from the browser cache with no

[37] MoodleDocs—File resource—https://docs.moodle.org/39/en/File_resource.

[38] MoodleDocs—File resource settings—https://docs.moodle.org/39/en/File_resource_settings. This information can be retrieved from the Moodle database table *mdl_resource*, feature display=4.

[39] MoodleDocs—Session handling—https://docs.moodle.org/39/en/Session_handling.

[40] Moodle-Log-Data-Temporal-Dimension—https://github.com/AD1529/Moodle-Log-Data-Temporal-Dimension.

contact with the server. This weakness makes it difficult to tell if students are lingering on a page because they are seeking anything, because they are clicking on the navigation drawer menu on the left (where the anchor links do not log any data), or because they are doing something else outside the platform.

To the best of our knowledge, Moodle's logging system, which is described in depth in Moodle's Developing Documentation, has not changed despite updates to the platform. To prevent the loss of data-related knowledge, we must ensure that the collected data accurately reflect the temporal behaviour we wish to capture. While reflecting on the vast number of similar cases, researchers should be conscious of the risk of misinterpretation while tracing data to conduct analytical studies, and we hope that our study will lead to additional investigations into the pitfalls that Moodle logging hides in LA research.

Declaration of Conflicting Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This work has been partially supported by EU—Horizon 2020 Program under the scheme "INFRAIA-01-2018-2019—Integrating Activities for Advanced Communities," Grant Agreement n.871042; "SoBigData++: European Integrated Infrastructure for Social Mining and Big Data Analytics" (http://www.sobigdata.eu); the scheme "HORIZON-INFRA-2021-DEV-02—Developing and consolidating the European research infrastructures landscape, maintaining global leadership (2021)," Grant Agreement n.101079043; "SoBigData RI PPP: SoBigData RI Preparatory Phase Project," by NextGenerationEU—National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR)—Project: "SoBigData.it—Strengthening the Italian RI for Social Mining and Big Data Analytics"—Prot. IR000001 3—Avviso n. 3264 del 28/12/2021; and by PNRR—M4C2—Investimento 1.3, Partenariato Esteso PE00000013—"FAIR—Future Artificial Intelligence Research"—Spoke 1 "Human-centered AI," funded by the European Commission under the NextGeneration EU programme.

References

Akhuseyinoglu, K., & Brusilovsky, P. (2022). Exploring behavioral patterns for data-driven modeling of learners' individual differences. *Frontiers in Artificial Intelligence*, *5*, 9 pages. https://doi.org/10.3389/frai.2022.807320

Alario-Hoyos, C., Rodrıguez-Triana, M. J., Scheffel, M., Arnedillo-Sánchez, I., & Dennerlein, S. M. (2020). Addressing global challenges and quality education: Proceedings of the 15th European Conference on Technology Enhanced Learning (EC-TEL 2020), 14–18 September 2020, Heidelberg, Germany. In *Lecture notes in computer science*. Springer Nature. https://doi.org/10.1007/978-3-030-57717-9

Bergner, Y. (2017). Measurement and its uses in learning analytics. In C. Lang, G. Siemens, A. Wise, & D. Gašević (Eds.), *Handbook of learning analytics* (pp. 35–48). Society for Learning Analytics Research (SoLAR). https://doi.org/10.18608/hla17.003

Bernardini, A., & Conati, C. (2010). Discovering and recognizing student interaction patterns in exploratory learning environments. In V. Aleven, J. Kay, & J. Mostow (Eds.), *Proceedings of the 10th International Conference on Intelligent Tutoring Systems Part I* (ITS 2010), 14–18 June 2010, Pittsburgh, PA*, Lecture notes in computer science,* vol. 6094 (pp. 125–134). https://doi.org/10.1007/978-3-642-13388-6_17

Bovo, A., Sanchez, S., Heguy, O., & Duthen, Y. (2013). Clustering Moodle data as a tool for profiling students. In *Proceedings of the Second International Conference on E-Learning and E-Technologies in Education* (ICEEE 2013), 23–25 September 2013, Lodz, Poland (pp. 121–126). IEEE. https://doi.org/10.1109/icelete.2013.6644359

Calder, B. J., Brendl, C. M., Tybout, A. M., & Sternthal, B. (2021). Distinguishing constructs from variables in designing research. *Journal of Consumer Psychology*, *31*(1), 188–208. https://doi.org/10.1002/jcpy.1204

Cerezo, R., Sánchez-Santillán, M., Paule-Ruiz, M. P., & Núñez, J. C. (2016). Students' LMS interaction patterns and their relationship with achievement: A case study in higher education. *Computers & Education*, *96*, 42–54. https://doi.org/10.1016/j.compedu.2016.02.006

Chen, B., Knight, S., & Wise, A. (2018). Critical issues in designing and implementing temporal analytics. *Journal of Learning Analytics*, *5*(1), 1–9. https://doi.org/10.18608/jla.2018.53.1

Conijn, R., Snijders, C., Kleingeld, A., & Matzat, U. (2017). Predicting student performance from LMS data: A comparison of 17 blended courses using moodle LMS. *IEEE Transactions on Learning Technologies*, *10*(1), 17–29. https://doi.org/10.1109/TLT.2016.2616312

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, *1*(1), 5–32. https://doi.org/10.1007/bf03325089

Darlington, W. J. (2017). *Predicting underperformance from students in upper level engineering courses*. Rochester Institute of Technology.

de Oliveira, P. C., de Almeida Cunha, C. J. C., & Nakayama, M. K. (2016). Learning management systems (LMS) and e-learning management: An integrative review and research agenda. *JISTEM—Journal of Information Systems and Technology Management*, *13*, 157–180. https://doi.org/10.4301/S1807-17752016000200001

Dermy, O., & Brun, A. (2020). Can we take advantage of time-interval pattern mining to model students activity?. In A. N. Rafferty, J. Whitehill, C. Romero, & V. Cavalli-Sforza (Eds.), *Proceedings of the 13th International Conference on Educational Data Mining* (EDM 2020), 10–13 July 2020, online (pp. 69–80). https://educationaldatamining.org/files/conferences/EDM2020/EDM2020Proceedings.pdf

Fincham, E., Whitelock-Wainwright, A., Kovanović, V., Joksimović, S., van Staalduinen, J.-P., & Gašević, D. (2019). Counting clicks is not enough: Validating a theorized model of engagement in learning analytics. In *Proceedings of the Ninth International Conference on Learning Analytics and Knowledge* (LAK 2019), 4–8 March 2019, Tempe, AZ (pp. 501–510). ACM. https://doi.org/10.1145/3303772.3303775

Gushchina, O., & Ochepovsky, A. (2020). Data mining of students' behavior in e-learning system. *Journal of Physics: Conference Series*, *1553*(1), 012027. https://doi.org/10.1088/1742-6596/1553/1/012027

Joksimović, S., Gašević, D., Loughin, T. M., Kovanović, V., & Hatala, M. (2015). Learning at distance: Effects of interaction traces on academic achievement. *Computers & Education*, *87*, 204–217. https://doi.org/10.1016/j.compedu.2015.07.002

Kapusta, J., Munk, M., & Drlik, M. (2012). Cut-off time calculation for user session identification by reference length. In *Proceedings of the Sixth International Conference on Application of Information and Communication Technologies* (AICT 2012), 17–19 October 2012, Tbilisi, Georgia (pp. 1–6). IEEE. https://doi.org/10.1109/icaict.2012.6398500

Knight, S., Wise, A. F., & Chen, B. (2017). Time for change: Why learning analytics needs temporal analysis. *Journal of Learning Analytics*, *4*(3), 7–17. https://doi.org/10.18608/jla.2017.43.2

Knobbout, J., Everaert, H., & van der Stappen, E. (2019). From dirty data to multiple versions of truth: How different choices in data cleaning lead to different learning analytics outcomes. In A. Pucihar, M. K. Borštnar, R. Bons, J. Seitz, H. Cripps, & D. Vidmar (Eds.), *Proceedings of the 32nd Bled eConference Humanizing Technology for a Sustainable Society,* 16–19 June 2019, Bled, Slovenia (pp. 49–66). Hogeschool Utrecht. https://doi.org/10.18690/978-961-286-280-0.4

Kovanović, V., Gašević, D., Dawson, S., Joksimovic, S., & Baker, R. (2016). Does time-on-task estimation matter? implications on validity of learning analytics findings. *Journal of Learning Analytics*, *2*(3), 81–110. https://doi.org/10.18608/jla.2015.23.6

Kularbphettong, K. (2018). Analysis of students' behavior based on educational data mining. In R. Silhavy, P. Silhavy, & Z. Prokopova (Eds.), *Applied computational intelligence and mathematical methods: Computational methods in systems and software 2017* (pp. 167–172). https://doi.org/10.1007/978-3-319-67621-0_15

Lang, C., Siemens, G., Wise, A., & Gasevic, D. (2017). *Handbook of learning analytics*. Society for Learning Analytics; Research (SoLAR). https://doi.org/10.18608/hla17

Lee, Y. (2018). Effect of uninterrupted time-on-task on students' success in massive open online courses (moocs). *Computers in Human Behavior*, *86*, 174–180. https://doi.org/10.1016/j.chb.2018.04.043

Macfadyen, L. P., & Dawson, S. (2010). Mining LMS data to develop an "early warning system" for educators: A proof of concept. *Computers & Education*, *54*(2), 588–599. https://doi.org/10.1016/j.compedu.2009.09.008

Nguyen, Q. (2020). Rethinking time-on-task estimation with outlier detection accounting for individual, time, and task differences. In *Proceedings of the 10th International Conference on Learning Analytics and Knowledge* (LAK 2020), 23–27 June 2020, Munich, Germany (pp. 376–381). ACM. https://doi.org/10.1145/3375462.3375538

Pedhazur, E. J., & Pedhazur Schmelkin, L. (1991). *Measurement, design, and analysis: An integrated approach*. Lawrence Erlbaum Associates.

Poon, L. K., Kong, S.-C., Wong, M. Y., & Yau, T. S. (2017). Mining sequential patterns of students' access on learning management system. In Y. Tan, H. Takagi, & Y. Shi (Eds.), *Proceedings of the Second International Conference on Data Mining and Big Data* (DMBD 2017), 27 July–1 August 2017, Fukuoka, Japan. *Lecture notes in computer science,* Vol. 10387 (pp. 191–198). Springer. https://doi.org/10.1007/978-3-319-61845-6_20

Ridzuan, F., & Zainon, W. M. N. W. (2019). A review on data cleansing methods for big data. *Procedia Computer Science*, *161*, 731–738. https://doi.org/10.1016/j.procs.2019.11.177

Riel, J., Lawless, K. A., & Brown, S. W. (2018). Timing matters: Approaches for measuring and visualizing behaviours of timing and spacing of work in self-paced online teacher professional development courses. *Journal of Learning Analytics*, *5*(1), 25–40. https://doi.org/10.18608/jla.2018.51.3

Rotelli, D., & Monreale, A. (2022). Time-on-task estimation by data-driven outlier detection based on learning activities. In *12th International Conference on Learning Analytics and Knowledge* (LAK 2022), 21–25 March 2022, online (pp. 336–346). ACM. https://doi.org/10.1145/3506860.3506913

Rotelli, D., Monreale, A., & Guidotti, R. (2022). Uncovering student temporal learning patterns. In I. Hilliger, P. Muñoz-Merino, T. De Laet, A. Ortega-Arranz, & T. Farrell (Eds.), *Educating for a New Future: Making Sense of Technology-Enhanced Learning Adoption: Proceedings of the 17th European Conference on Technology Enhanced Learning* (EC-TEL 2022), 12–16 September 2022, Toulouse, France. *Lecture notes in computer science,* Vol. 13450 (pp. 340–353). https://doi.org/10.1007/978-3-031-16290-9_25

Sael, N., Marzak, A., & Behja, H. (2013). Web usage mining data preprocessing and multi level analysis on moodle. In *Proceedings of the 2013 ACS International Conference on Computer Systems and Applications* (AICCSA), 27–30 May 2013, Ifrane, Morocco. IEEE. https://doi.org/10.1109/aiccsa.2013.6616427

Saqr, M., Nouri, J., & Fors, U. (2018). Temporality matters. A learning analytics study of the patterns of interactions and its relation to performance. In *Proceedings of the 10th International Conference on Education and New Learning Technologies* (EDULEARN 2018), 2–4 July 2018, Palma, Spain (pp. 5386–5393). https://doi.org/10.21125/edulearn.2018.1305

Siemens, G., & Long, P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review*, *46*(5), 30. https://doi.org/10.17471/2499-4324/195

Villalobos, E., Pérez-Sanagustin, M., Sanza, C., Tricot, A., & Broisin, J. (2022). Supporting self-regulated learning in BL: Exploring learners' tactics and strategies. In *Educating for a New Future: Making Sense of Technology-Enhanced Learning Adoption: Proceedings of the 17th European Conference on Technology Enhanced Learning* (EC-TEL 2022), 12–16 September, 2022, Toulouse, France (pp. 407–420). https://doi.org/10.1007/978-3-031-16290-9_30

Wang, Y., Li, T., Geng, C., & Wang, Y. (2019). Recognizing patterns of student's modeling behaviour patterns via process mining. *Smart Learning Environments*, *6*(1), 1–16. https://doi.org/10.1186/s40561-019-0097-y

Yu, T., & Jo, I.-H. (2014). Educational technology approach toward learning analytics: Relationship between student online behavior and learning performance in higher education. In *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge* (LAK 2014), 24–28 March 2014, Indianapolis, IN (pp. 269–270). ACM. https://doi.org/10.1145/2567574.2567594

Appendix A

Although complete, the examples in the table are tied to the logs extracted from the aforementioned courses. They represent the vast majority of collectible *Students'* logs for Moodle's standard modules and related settings. Other course modules (particularly those related to additional plugins[41]) may generate components and event names that are not listed in the table, but which are updated from time to time in the GitHub repository[40]. Note that the components marked with an asterisk are usually referred to as "System" but have been redefined, as described in Rotelli and Monreale (2022).

Component	Event name	Category	Annotation
Assignment	A file has been uploaded	Simultaneous	
	A submission has been submitted	Simultaneous	The event is *instantaneous* if students are requested to press the "Submit" button.[27]
	An online text has been uploaded	Simultaneous	
	Comment created	Ending	The time that begins when this event is recorded must be considered "The status of the submission has been viewed" because the user remains on the same page.
	Comment deleted	Instantaneous	
	Course module instance list viewed	Starting	
	Course module viewed	Simultaneous	
	Feedback viewed	Simultaneous	
	Remove submission confirmation viewed	Starting	
	Submission confirmation form viewed	Starting	
	Submission created	Ending	This event must be swapped with the "Course module viewed" event to be *ending* (Table 6).
	Submission form viewed	Starting	
	Submission updated	Ending	This event must be swapped with the "Course module viewed" event to be *ending* (Table 6).
	The status of the submission has been updated	Simultaneous	
	The status of the submission has been viewed	Starting	
	The user duplicated their submission	Ending	This event must be swapped with the "Course module viewed" event to be *ending* (Table 6).
Badge	Badge listing viewed	Starting	
	Badge viewed	Starting	
Blog	Blog entries viewed	Starting	
Book	Book printed	Instantaneous	
	Chapter printed	Instantaneous	
	Chapter viewed	Starting	
	Course module viewed	Simultaneous	
Calendar*	Calendar event created	Ending	Since these events are not accompanied by other concurrent events indicating the type of action (such as in the Forum), the time that begins when they are recorded should be considered "Calendar viewed." Whenever a user with the teacher role creates activities with a due date, these events are *Simultaneous*.
	Calendar event deleted	Ending	
	Calendar event updated	Ending	
Chat	Course module viewed	Starting	
	Message sent	Ending	The time that begins when this event is recorded must be considered "Course module viewed" because the user remains on the same page.
	Sessions viewed	Starting	

[41] MoodleDocs—Installing plugins—https://docs.moodle.org/39/en/Installing_plugins.

Component	Event name	Category	Annotation
Choice	Choice answer added	Closing	When the "Save my choice" button is clicked, a number of logs equal to the number of responses provided is recorded if multiple answers are required. The time that begins when this event is recorded must be considered "Choice summary viewed."
	Choice answer deleted	Closing	When a modifiable answer is saved, a number of logs equal to the number of modified answers is recorded. The time that begins when this event is recorded must be considered "Course module viewed" since the user is redirected to the initial page.
	Course module viewed	Opening	This event is *opening* if it is followed by an answer (*closing*); if the choice is viewed but no answer is given, the event is *starting*.
Contact request*	Notification sent	Instantaneous	
Course home*	Course viewed	Starting	
Courses list*	Category viewed	Starting	
	Courses searched	Starting	
Dashboard	Dashboard reset	Instantaneous	
	Dashboard viewed	Starting	
Database	Course module viewed	Starting	
Feedback	Course module viewed	Opening	This event is *opening* if the response is submitted (*closing*); if the feedback is viewed without a submission, the event is *starting*.
	Response submitted	Closing	The event *Feedback summary viewed* must be added after this closing event to represent the time spent after the completed action.
File	Course module viewed	Starting	If the download is forced in the settings, the event is *instantaneous*.[42]
Folder	Course module viewed	Starting	
	Zip archive of folder downloaded	Instantaneous	
Forum	Course module instance list viewed	Starting	
	Course module viewed	Starting	
	Course searched	Starting	Cf. footnote 31.
	Discussion created	Simultaneous	
	Discussion subscription created	Simultaneous	If the user manually subscribes to the discussion, this event is *instantaneous*[43].
	Discussion subscription deleted	Simultaneous	If the user manually unsubscribes from the discussion, this event is *instantaneous*[43].
	Discussion viewed	Starting	
	Post created	Simultaneous	The time that begins when this event is recorded must be considered "Discussion viewed" because the user remains on the same page.
	Post deleted	Instantaneous	
	Post updated	Simultaneous	
	Some content has been posted	Ending	
	Subscription created	Simultaneous	If the user manually subscribes to the forum, this event is *Instantaneous*[44].
	Subscription deleted	Simultaneous	If the user manually unsubscribes from the forum, this event is *Instantaneous*[44].

[42] This information can be retrieved from the *mdl_resource* database table.
[43] This information can be retrieved from the *mdl_forum_discussion_subs* database table.
[44] This information can be retrieved from the *mdl_forum_subscriptions* database table.

Component	Event name	Category	Annotation
	User report viewed	Starting	
Glossary	Course module viewed	Starting	
	Entry has been viewed	Starting	When the entry is viewed outside of the Glossary, the time spent viewing it cannot be determined because no log is recorded when the entry is closed.
Grades*	Course user report viewed	Starting	
	Grade overview report viewed	Starting	
	Grade user report viewed	Starting	
H5P	Course module viewed	Simultaneous	
	H5P content viewed	Opening	This event is *opening* if the response is submitted (*closing*); if the H5P is viewed without a submission, the event is *starting*[45].
	Report viewed	Starting	
	xAPI statement received	Closing	The event *H5P summary viewed* should be added after this closing event to represent the time spent after the completed action.
Lesson	Content page viewed	Starting	
	Course module viewed	Simultaneous	
	Lesson ended	Starting	This event, contrary to its name, opens a page indicating that the student has reached the end of the lesson.
	Lesson restarted	Simultaneous	
	Lesson resumed	Simultaneous	
	Lesson started	Simultaneous	
	Question answered	Closing	The event *Question summary viewed* should be added after this closing event to represent the time spent after the completed action.
	Question viewed	Opening	This event is *opening* if the question is answered (*closing*); if the question is viewed without an answer, the event is *starting*.
Login*	User has logged in	Simultaneous	The session begins, but depending on the settings, another event is recorded simultaneously[46].
Logout*	User logged out	Starting	The session has ended and the user starts to be inactive.
Messaging*	Group message sent	Ending	The time that begins when this event is recorded must be considered "Message viewed" because the user remains on the same page.
	Message contact added	Instantaneous	
	Message deleted	Instantaneous	
	Message sent	Ending	The time that begins when this event is recorded must be considered "Message viewed" because the user remains on the same page.
	Message viewed	Starting	
Notification*	Notification viewed	Starting	
Page	Course module viewed	Starting	
Participant profile*	User list viewed	Starting	
	User profile viewed	Starting	
Quiz	Course module viewed	Starting	
	Quiz attempt reviewed	Starting	

[45] For Flashcards, no submission is required.
[46] MoodleDocs—Front page settings—https://docs.moodle.org/39/en/Front_page_settings.

Component	Event name	Category	Annotation
	Quiz attempt started	Simultaneous	As soon as a quiz starts, a question is displayed, so this event is simultaneous to the "Quiz attempt viewed" event.
	Quiz attempt submitted	Simultaneous	When a quiz is submitted, the given answers are displayed, so this event is concurrent to the "Quiz attempt reviewed" event.
	Quiz attempt summary viewed	Starting	
	Quiz attempt viewed	Starting	This log is recorded every time students click on the "Next/Previous" button to view a question on the Quiz, not necessarily giving the answer. The action of answering, however, is not recorded.
Site home*	Course viewed	Starting	
Survey	Course module viewed	Opening	This event is *opening* if the response is submitted (*closing*); if the Survey is viewed without a submission, the event is considered to be *starting*.
	Survey response submitted	Closing	The event *Survey summary viewed* should be added after this closing event to represent the time spent after the completed action.
Tag	Tag added to an item	Instantaneous	
	Tag created	Simultaneous	This event is logged before the "Tag added to an item" event. If the tag already exists, this event is not recorded.
	Tag deleted	Simultaneous	
	Tag removed from an item	Instantaneous	
URL	Course module viewed	Starting	
User profile*	User password updated	Ending	The time that begins when this event is recorded must be considered "User profile viewed" since the user is redirected to their profile preference page.
	User profile viewed	Starting	
	User updated	Ending	The time that begins when this event is recorded must be considered "User profile viewed" since the user is redirected to their profile page.
Wiki	Comments viewed	Starting	
	Course module viewed	Starting	
	Wiki diff viewed	Starting	
	Wiki history viewed	Starting	
	Wiki page created	Opening	This event is *opening* if the user clicks the "Save" button and the page is updated (*closing*); if the user creates a page without saving it, the event is considered to be *starting*.
	Wiki page locks deleted	Simultaneous	
	Wiki page map viewed	Starting	
	Wiki page updated	Closing	If the page already exists, the log is the same, but the event is *ending*. Since this log is already accompanied by the *Wiki page viewed* event, no additional event should be created.
	Wiki page version viewed	Starting	
	Wiki page viewed	Starting	
All components	Course activity completion updated	Simultaneous	If the user manually marks the activity as completed, this event is *Instantaneous*[47].

Table 12. Event Category

[47]This information can be retrieved from the settings in the database tables. However, since those events have some issues that still need to be fixed (https://moodle.org/mod/forum/discuss.php?d=391272) and they are not informative for temporal analysis, we do not consider them. If they were marked automatically (*simultaneous*), students would be engaged in other activities. Activities can only be manually marked as completed (*instantaneous*) from the course homepage, meaning that the act of marking does not temporally interrupt any learning activities.

Appendix B

As stated in Section 3, we replicated the *Assignment* module and repeated the experiment many times by changing the settings to analyze the differences. Depending on the settings[27], students can upload files or write online text, click the submit button, manually or automatically mark the task as completed[47], and click the submit button. Tables 13, 14, and 15 show the various logs that can be collected by modifying the configuration settings. Table 13 illustrates the sequence of logs collected by setting the basic configuration. To submit an assignment, the number of logs collected is 11:

- Submission type = File submissions
- Require students to click the submit button = No
- Require that students accept the submission statement = No
- Completion tracking = Students can manually mark the activity as completed

Table 13. The Assignment Activity—Test 1

T	Timestamp	Event context	Component	Event name
1	2022-03-12T8:15:41	Assignment: Test 1	Assignment	Course module viewed
2	2022-03-12T8:15:41	Assignment: Test 1	Assignment	The status of the submission has been viewed
3	2022-03-12T8:15:43	Assignment: Test 1	Assignment	Course module viewed
4	2022-03-12T8:15:44	Assignment: Test 1	Assignment	Submission form viewed
5	2022-03-12T8:15:57	Assignment: Test 1	Assignment	Course module viewed
6	2022-03-12T8:15:57	Assignment: Test 1	File submissions	A file has been uploaded
7	2022-03-12T8:15:57	Assignment: Test 1	File submissions	Submission created
8	2022-03-12T8:15:57	Assignment: Test 1	System	Course activity completion updated
9	2022-03-12T8:15:57	Assignment: Test 1	Assignment	A submission has been submitted
10	2022-03-12T8:15:57	Assignment: Test 1	Assignment	Course module viewed
11	2022-03-12T8:15:57	Assignment: Test 1	Assignment	The status of the submission has been viewed

Table 14 shows the sequence of logs collected by modifying the settings as follows. To submit an assignment, the number of logs collected is 18. It is worth noting that both *Online text submissions* and *File submissions* are recorded regardless of whether the student submits a file or writes an online text.

- Submission type = File submissions & Online text
- Require students to click the submit button = Yes
- Require that students accept the submission statement = Yes
- Completion tracking = Students must make a submission

Table 14. The Assignment Activity—Test 2

T	Timestamp	Event context	Component	Event name
1	2022-03-12T8:16:13	Assignment: Test 2	Assignment	Course module viewed
2	2022-03-12T8:16:13	Assignment: Test 2	Assignment	The status of the submission has been viewed
3	2022-03-12T8:16:15	Assignment: Test 2	Assignment	Course module viewed
4	2022-03-12T8:16:15	Assignment: Test 2	Assignment	Submission form viewed
5	2022-03-12T8:16:29	Assignment: Test 2	Assignment	Course module viewed
6	2022-03-12T8:16:29	Assignment: Test 2	Online text submissions	An online text has been uploaded
7	2022-03-12T8:16:29	Assignment: Test 2	Online text submissions	Submission created
8	2022-03-12T8:16:29	Assignment: Test 2	File submissions	A file has been uploaded
9	2022-03-12T8:16:29	Assignment: Test 2	File submissions	Submission created
10	2022-03-12T8:16:29	Assignment: Test 2	Assignment	Course module viewed
11	2022-03-12T8:16:29	Assignment: Test 2	Assignment	The status of the submission has been viewed
12	2022-03-12T8:16:36	Assignment: Test 2	Assignment	Course module viewed
13	2022-03-12T8:16:36	Assignment: Test 2	Assignment	Submission confirmation form viewed
14	2022-03-12T8:16:39	Assignment: Test 2	Assignment	Course module viewed
15	2022-03-12T8:16:39	Assignment: Test 2	System	Course activity completion updated
16	2022-03-12T8:16:39	Assignment: Test 2	Assignment	A submission has been submitted
17	2022-03-12T8:16:39	Assignment: Test 2	Assignment	Course module viewed
18	2022-03-12T8:16:39	Assignment: Test 2	Assignment	The status of the submission has been viewed

Table 15 collects a greater number of logs ($n = 25$) because the student updated the submission and added some comments. This example expands Table 6.

- Submission type = File submissions
- Require students to click the submit button = Yes
- Require that students accept the submission statement = Yes
- Completion tracking = Students can manually mark the activity as completed & Students must make a submission

Table 15. The Assignment Activity—Test 3

T	Timestamp	Event context	Component	Event name
1	2022-03-12T18:12:20	Assignment: Test 3	System	Course activity completion updated
2	2022-03-12T18:12:20	Assignment: Test 3	Assignment	Course module viewed
3	2022-03-12T18:12:20	Assignment: Test 3	Assignment	The status of the submission has been viewed
4	2022-03-12T18:12:23	Assignment: Test 3	Assignment	Course module viewed
5	2022-03-12T18:12:23	Assignment: Test 3	Assignment	Submission form viewed
6	2022-03-12T18:12:29	Assignment: Test 3	Assignment	Course module viewed
7	2022-03-12T18:12:29	Assignment: Test 3	File submissions	A file has been uploaded
8	2022-03-12T18:12:29	Assignment: Test 3	File submissions	Submission created
9	2022-03-12T18:12:29	Assignment: Test 3	Assignment	Course module viewed
10	2022-03-12T18:12:29	Assignment: Test 3	Assignment	The status of the submission has been viewed
11	2022-03-12T18:12:34	Assignment: Test 3	Submission comments	Comment created
12	2022-03-12T18:12:37	Assignment: Test 3	Assignment	Course module viewed
13	2022-03-12T18:12:37	Assignment: Test 3	Assignment	Submission form viewed
14	2022-03-12T18:12:46	Assignment: Test 3	Assignment	Course module viewed
15	2022-03-12T18:12:46	Assignment: Test 3	File submissions	A file has been uploaded
16	2022-03-12T18:12:46	Assignment: Test 3	File submissions	Submission updated
17	2022-03-12T18:12:46	Assignment: Test 3	Assignment	Course module viewed
18	2022-03-12T18:12:46	Assignment: Test 3	Assignment	The status of the submission has been viewed
19	2022-03-12T18:12:49	Assignment: Test 3	Assignment	Course module viewed
20	2022-03-123T8:20:49	Assignment: Test 3	Assignment	Submission confirmation form viewed
21	2022-03-12T18:12:50	Assignment: Test 3	Assignment	Course module viewed
22	2022-03-12T18:12:50	Assignment: Test 3	System	Course activity completion updated
23	2022-03-12T18:12:50	Assignment: Test 3	Assignment	A submission has been submitted
24	2022-03-12T18:12:50	Assignment: Test 3	Assignment	Course module viewed
25	2022-03-12T18:12:50	Assignment: Test 3	Assignment	The status of the submission has been viewed

At time T_2, the log records the student accessing the activity "Assignment: Test 3," but at time T_1 (the same as T_2 but earlier in the list), the activity has already been logged as completed. At time T_3, one more entry records the status of the submission as "viewed." These three logs are concurrent ($T_1 = T_2 = T_3$) because they happen at the very same time: the student accesses the assignment, and the activity is completed (T_1); the module description is displayed at the top of the page (T_2), while the status of the submission is shown at the bottom T_3). At time T_4, the student clicks on the "Add submission" link and a new request to the server is sent (*&action=editsubmission*). However, the log records the same event as T_2: this happens because the top of the loaded page is the same as before, while at the bottom ($T_5 = T_4$), the submission status has been replaced by the form to submit the file. When the file is uploaded, five concurrent events ($T_6 = T_7 = T_8 = T_9 = T_{10}$) are recorded: event T_6 identifies that the new request to the server (*&action=view*) always loads the same page (like T_2 and T_4), T_7 records the uploading of the file (the component changes to "File submissions" and identifies the module settings; if students were asked to submit text online, the component would be "Online text submissions," but both components can be present depending on the settings), and concurrently (T_8) the creation of the submission is logged. Moreover, events at times T_9 and T_{10} are exactly the same as T_2 ($= T_4 = T_6$) and T_3 because the page shows the module description at the top and the updated submission status at the bottom.

At time T_{11}, the student adds a comment in the submission status, and the component changes in "Submission comments" to indicate that comments differ from the Assignment activity, although in the same context. The log is recorded only at the end of the event after the student clicks on the "Save comment" link. Then, at times T_{12} and T_{13}, there are the same couple of events as in T_4 and T_5. We cannot understand what is happening until the event at time T_{16} highlights that the student updated the submission (created at time T_8) by repeating the same action as before: clicking on the "Edit submission" button. This button, available for activity settings "Require students to click the submit button," allows students to edit the submission many times before clicking it. Therefore, T_{17} and T_{18} repeat the events T_9 and T_{10}. At time T_{19}, the student finally clicks on the "Submit assignment" button, and the "Confirm submission" page (*&action=submit*) is loaded at the same time (T_{20}). When at time

T_{21}, the student clicks on the "Continue" button, there are four more concurrent events: the system automatically records that the activity is completed (T_{22}) and that the submission is submitted (T_{23}); then the assignment page is updated (T_{24}) with the submission information (T_{25}).

It is evident that the number of logs, as well as the number of simultaneous events, depends on the actions performed not only on the student side (edit submission, remove submission) but also on the teacher side (module settings). Therefore, when counting the number of actions performed on modules of different courses to analyze student behaviour on tasks or the time spent on them (the number of actions required from the student for submission lengthens the time on task), all these aspects should be taken into consideration to avoid biased results.