Data Quality Challenges in Educational Process Mining: Building Process-Oriented Event Logs from Process-Unaware Online Learning Systems

Rahila Umer
Balochistan University of Information Technology, Engineering & Management Sciences, Quetta, Pakistan
rumer@buitms.edu.pk

Teo Susnjak
School of Natural and Computational Sciences, Massey University, New Zealand
t.susnjak@massey.ac.nz

Anuradha Mathrani
School of Natural and Computational Sciences, Massey University, New Zealand
a.s.mathrani@massey.ac.nz

Suriadi Suriadi
Queensland University of Technology, Brisbane, Australia
s.suriadi@qut.edu.au

ABSTRACT

Educational process mining utilizes process-oriented event logs to enable discovery of learning practices that can be used for the learner's advantage. However, learning platforms are often process-unaware, therefore do not accurately reflect ongoing learner interactions. We demonstrate how contextually relevant process models can be constructed from process-unaware systems. Using a popular learning management system (Moodle), we have extracted stand-alone activities from the underlying database and formatted it to link the learners' data explicitly to process instances (cases). With a running example that describes quiz-taking activities undertaken by students, we describe how learner interactions can be captured to build process-oriented event logs. This article contributes to the fields of learning analytics and education process mining by providing lessons learned on the extraction and conversion of process-unaware data to event logs for the purpose of analysing online education data.

KEYWORDS

Learning analytics; process mining; quiz-taking behaviour; learning management system; education data; process instance; data quality

Cite as:

Rahila Umer, Teo Susnjak, Anuradha Mathrani and Suriadi Suriadi (2022). Data Quality Challenges in Educational Process Mining: Building Process-Oriented Event Logs from Process-Unaware Online Learning Systems. *International Journal of Business Information Systems*. Vol. 39, No. 4. P. 569 – 592. doi: 10.1504/IJBIS.2020.10027967

Data Quality Challenges in Educational Process Mining: Building Process-Oriented Event Logs from Process-Unaware Online Learning Systems

1. Introduction

Modern day educational practices are integrated with learning management systems (LMS), which ensure cost-effective web-enabled subject deliveries. LMS aid in dissemination of educational material, in facilitating learning and assessment activities, and in supporting associated educational administrative functions (Such, Ritzhaupt & Thompson, 2017). LMS integrates many perspectives, including a communication perspective, a decision-making perspective, a technological perspective and a people-based perspective since students, faculty and administration staff all rely on it. LMS reflect the institutional culture since it informs about the overall learning and teaching practices, besides revealing other aspects of student learning, student motivation, instructor attitudes and different administrative practices that are being used. Therefore, if we look beyond the technical nature of the tool, LMS can provide more contextual awareness of learner interactions to create organisational transformations that can be effectively used for the learner's advantage.

Learning Analytics (LA) is an emerging field that utilizes large amounts of data extracted from learners' online activities to provide operational and actionable insights for enhancing their overall learning experience (Ferguson, 2012; Hwang et al., 2017; Siemens and Long, 2011). For an institution intent on improving their course completion rates and providing a conducive learning environment, education data can be capitalised using LA to provide more clarity on effective teaching and learning strategies. LA has garnered much interest in recent research as data-driven analytical methods to understand current issues and challenges within learners' contexts, which can then be addressed to improve the learning process (Avella et al., 2016; Gwo-Jen Hwang, 2017; Peña-Ayala; Siemens, 2013; Vahdat et al., 2015; Wong et al., 2018). One of the most common uses of education data is to co-relate students' online behaviour with academic performance to enable identification of those students who are less likely to succeed academically and to provide them with timely support to overcome their learning difficulties (Chang et al., 2014; Hsiao et al., 2018; Umer et al., 2017; Yang et al., 2016).

An emerging field of research, educational process mining (EPM) has led the way for advanced learning analytics, where learner produced activity data can be contextualised to understand students' learning styles/habits that influence their academic performance. Moreover, these habits can then be represented visually through process models (Romero and Ventura, 2013). While process mining is a combination of business analytics and data mining methods (van Der Aalst et al., 2011; van Der Aalst, 2016), EPM is combination of learning analytics and data mining methods. However, the importance of input data quality to make relevant contributions via EPM cannot be underestimated. The static data stored in LMS needs to be extracted and transformed into a pre-defined format so that pertinent event logs can be constructed prior to applying process mining techniques. But poor data quality can reduce the transformational impact of data, although data can never be of perfect quality (Baesens et al., 2016). Quality has always been a relative concept that may be considered as acceptable/unacceptable built upon trustworthiness, completeness, traceability and appropriateness.

Event logs serve as a starting point in process-oriented analysis as they capture time-stamped information of business processes like what was done, by whom, for whom, where or when (Tax et al. 2019). This helps in providing detailed sequential information of various activities associated with the business process. In other words, event logs comprise process instances (or cases) where sets of related activities are mapped to specific execution of tasks. Event logs are basic inputs to which process mining techniques are applied to enable discovery of complementary process flows, which can be then analysed and further optimised to develop data-driven strategies that increase performance efficiencies. First, the event log data is inputted to produce process-related information and derive process models. Analytics can then be embedded into process models to answer different questions related to performance and conformance. The quality of process mining analytics depends on the accuracy and completeness of event logs. The input data often comes in raw form from different sources (e.g., database, enterprise systems, flat files, logs, etc.). Identifying process-related data from the static data, and the subsequent extraction and conversion of static data to required format for building an event log is a complex task. Data are scattered and not found collectively in one place. Further, lack of uniform definitions among the scattered data elements can lead to inconsistent mappings and result in less rewarding benefits from the analytics conducted (Kwon et al., 2014). LMS often lack good instructional design with complex implementations of the underlying processes that are not maintained or updated in subsequent versions (Adnan and Ritzhaput, 2018). Therefore, to ask relevant questions, having domain knowledge of the underlying learning platform is crucial.

The challenge in EPM is the extraction and the merging of disparate datasets such that they collectively represent accurate process flows for subsequent process mining applications. More often than not, the data stored in enterprise systems are static and not process-oriented; therefore, data needs to be transformed into process-aware event logs that reflect the execution of completed process instances and inform about related activities for each process instance. However, before attempting to build process-aware logs one needs to define the scope of a 'process' and consider the feasibility of extracting the required data.

This article contributes to the fields of learning analytics and process mining by providing lessons learned in the extraction and conversion of process-unaware data to event logs for the purpose of analysing online education data. We share many quality issues generally facing education data and with a specific running example (i.e., quiz-taking) to demonstrate these issues. With a few exceptions (e.g., Jans and Soffer, 2017), to the best of our knowledge, these practice experiences are relatively scarce within existing literature; therefore, this work contributes significantly to the process mining body of knowledge. Moreover, analytics is an ever-evolving field, such that even recently graduated analysts have to be continually re-trained in data capture, data modelling, data analysis, data predictive and data-driven decision-making approaches (Baesens et al., 2016).

The structure of the paper is as follows. The next section describes the background and intent of this study. Prior work on data quality issues are elaborated next. Following section discusses process mining concepts and explains the use of event logs as inputs for conducting process mining. Next, we demonstrate the steps involved in making event logs with a specific running example. The challenges faced during the event extraction, their impact on the quality of the data and our proposed resolutions to address these challenges are discussed next. Finally, the main lessons learned in building event logs from process-unaware systems are summarised and study's contributions are stated.

2. Background and study's intent

This study focuses on preparation of event logs from educational data gathered from different courses delivered in a university setting. Data have been gathered from two platforms, that is the student management system (SMS) comprising student grades and profile information and the learning management system (Moodle) comprising course-related instructional tasks and which is used for communicating with students over the course duration. Moodle is a very popular open-source learning platform and considered an exemplar of LMS by many higher education institutions (Ifinedo, Pyke & Anwar, 2018). Our intent has been to generate event logs from incompatible underlying process-unaware platforms for conducting educational process mining analysis. This study is limited to data extraction, case identification and data quality improvements in using process-unaware data from online learning platforms. We analysed more than 50 courses for event log formations; however, in our demonstration of a running example we have described data from one course only. Data related to student's participation in an online quiz of one course has been used to construct reasonable event logs for future use in EPM. We describe some challenges that were faced in data usability and data integration and propose recommendations that can assist in creating a process-oriented event log from process-unaware databases.

3. Previous research

This section presents previous work done in the domain of data quality for process mining analysis. Process mining manifesto (van Der Aalst et al., 2011) created by the IEEE task force on process mining provides a mechanism to measure the quality of the event data and to rank them (from 1 star to 5 stars). Best quality event logs (5 stars) are characterised as complete and trustworthy as they are recorded automatically, whereas poor quality (1 and 2 stars) event logs are incomplete and considered not as reliable since they are mostly recorded manually. The manifesto suggests that only event logs rated from 3 to 5 stars are suitable for process mining analysis. Four broad categories, namely missing data, incorrect data, imprecise data and irrelevant data are common issues that affect event log quality when used for process mining (Bose et al., 2013). Such data inconsistencies and data anomalies pose huge challenges for the analyst as process-mining is data-driven. Processes provide a workflow perspective where various tasks require time-ordered operational data that can be queued to demonstrate dynamic behaviours. Calvanese et al. (2015) used an ontology-based approach to extract event logs from a relational database. The data stored in databases are flattened as XES file using both domain ontology and event ontology. Although, this technique can provide access to the data in databases using query unfolding and by applying ontology-based data access (ODBA) methods (Poggi et al., 2008), however performance issues can occur when dealing with large databases.

Many studies have highlighted different challenges that are faced during the extraction process of data from a process-unaware system. For instance, Van Der Aalst (2015) presented a different approach to transform relational database into event logs. They used classification to create multiple event logs for making comparisons; however, there was no consideration to business-related decisions, rather their study mainly discussed theoretical challenges of extracting event logs. Real world business operations are driven by external environments, organisational policies and managerial decisions as businesses strive to have a competitive advantage. Another study by Pérez-Castillo et al. (2014) created event logs by extracting logs from a non-process-oriented system based on correlation of events and with similarity between attributes. Jans and Soffer (2017) also discuss issues in extraction of event data from

a relational database using an end-user's perspective. Using an example of procure-to-pay process, they described a structured procedure to extract data from a relational database and convert to an event log. Even so, they stress on the multiplicity of decision-making factors that influence the selection of process instance and associated activities, which can impact the quality of the event log.

Further, Selig (2017) demonstrated how results of process analysis of data produced by process-unaware systems are different than the one produced by process mining system. Using an example of purchasing process from a SAP module of an enterprise system platform, Selig shows the emergence of ambiguous cases such as divergence and convergence. The author proposes continuous data extractions from the enterprise system and transforming it into an event log by considering proper case attributes, the notion of a case and the granularity of events to address data quality issues. Kim et al. (2003) presented a detailed taxonomy of 'dirty' data as a framework for understanding the origin of such data. They proposed a metric for measuring the quality of data and have referred to data as 'dirty' if the results of data analysis are not up to expectations due to the low quality. Moreover, their study explored the impact of such data on data mining and provided techniques for dealing with unclean data. A different approach is to consider data quality problems in relation to its source or origin (i.e., single-source versus multi-source) and its granularity level (i.e., at instance-level or at schema level). Rahm and Do (2000) emphasise the use of transformation techniques for cleaning of data to cover both instance and schema perspectives in an integrated manner and have presented commercial tools with their limitations for data cleaning. While these data quality taxonomies cover some of the problems that have an impact on process mining analysis, they are not completely related to the process mining perspective. Suriadi et al. (2017) used pattern-based approaches to identify common data quality problems which were distilled from their experience and labelled them with pattern terminology. Their patterns are validated with use of event logs from practice and have been evaluated by research experts in the domain to serve as knowledge repository for event log preparation and provide recommendations on improving data quality.

Different tools are available to extract event data from database and convert it to XES file. For example, XESame (Verbeek et al., 2010) provides a platform where data is selected and matched with XES elements. However, there is no direct access to the database and database is only used for storage purpose. Other similar commercial tools examples are *Minit* [1] and *Celonis* [2]. Best part of these tools is high efficiency in data extraction, but the downside is that they cannot handle huge amounts of data, especially if the computer memory is exceeded because the transformation takes place in memory. Similar to previous tools, data cannot be accessed directly from database. To address the issues of memory and have direct access to the database, another technique was proposed by van Dongen and Shabani (2015) which used comparatively less memory; however, there is no empirical evidence yet to ascertain its time-performance.

4. Process mining

The objective of process mining is to conduct data analyses to project a process-oriented perspective and answer questions related to business process flows. Figure 1 shows the workflow for process mining. The first step involves articulating a plan for understanding the

[1] https://www.minit.io/

[2] https://www.celonis.com/

available data and the business domain which is aligned with the expected outcomes. Next step is to extract the raw data which might have originated from multiple sources (e.g., flat file, excel sheet, database table etc.). This phase of data collection is quite challenging; more so since data is typically not structured or sometimes the metadata is missing. Data management issues such as the integration and sharing of data are common challenges faced by any analytics teams in organisations. Moreover, there is abundance of data, and all data is not meant to be used; instead, data is extracted based on the questions or scope of the problem. Scoping the problem requires a combination of quantitative and software skills alongside required business domain knowledge (Lismont, et al., 2017).

Event log is created which is further filtered depending on the granularity level of the problem under analysis. The filtered event log is the input for process mining applications: process discovery, conformance and enhancement.

- *Discovery:* By analysing event-based data (i.e., event log), analysts can reveal process flows to show how different processes are being performed for a given task. In education domain, a discovery example would relate to detection of patterns of various activities that students perform while taking a quiz.
- *Conformance:* Here the event log is analysed to verify whether the existing business model was being followed or not. Event log is again an input for conformance checking.
- *Enhancement:* Enhancement is used to analyse other process perspectives by exploiting the discovered model, such as resource analysis, performance analysis, as well as decision mining.

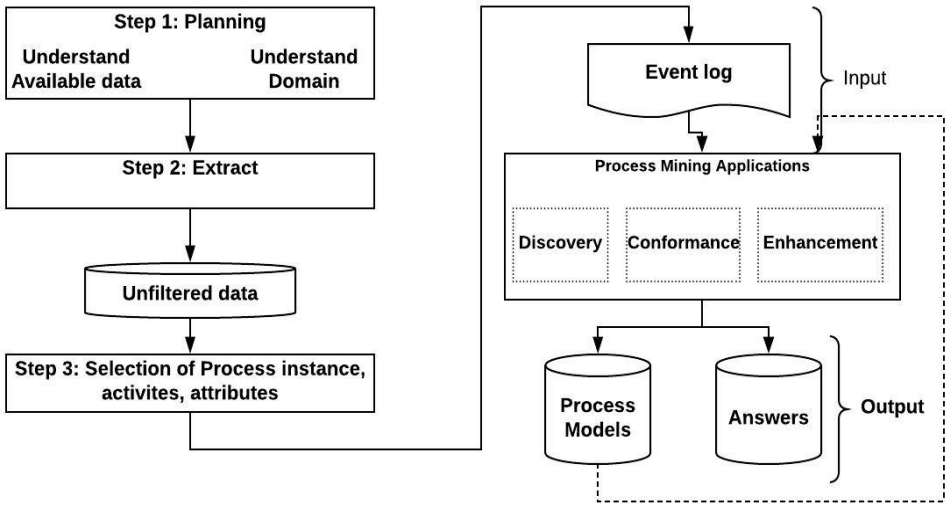

Figure 1. Life cycle model describing major steps for process mining project consists of planning, data extraction and selection of case, activities and attributes.

The filtering methods enable discovery of behaviourally specific process models that is not overgeneralised with too many behavioural possibilities (Tax et al., 2019). Tax et al. refer to simultaneous activities as 'chaotic activities' since they can cause chaos in deciphering behaviours accurately as they lead to many process traces that together reflect the completeness of the event log. Therefore, selection of process instances, activities and attributes is crucial before playing of the process discovery model.

Event logs for process mining comprises following elements.

- *Case:* Each entry in the log maps to a single case. A case is also called process instance which reflects the single execution of a process.
- *Events:* Events are single entries in the log. Events are mapped to cases which are ordered based on time-stamp. Events relate to specific activities and associated attributes.
- *Variant:* A specific sequence of activities referred as variants is used to compare different cases. An event log comprises a set of events (or time-stamped order of performed activities). Every event is mapped to a case which has specific orders of events. Each unique sequence of activities makes a variant.
- *Attributes:* Certain specification or attributes can give more information at either the event or the case level.

In 2010, the IEEE Task Force on Process Mining adopted XES (Extendable Event Stream) as a standard for process mining event logs (van Der Aalst et al., 2011). XES is XML based, in which each event log entry comprises an event type, a time stamp and associated attributes. Attributes are assigned at different levels, i.e., log level, case level, or event level. An XES log can contain a number of cases (traces in XES language), where each case describes a sequence of events pertaining to this case. This requires that every event is already assigned to a case and all events related to a case are known.

5. Event extraction

Our next step was to understand the underlying Moodle database from where relevant data can be collected to meet the primary goal, that is, to get a process-oriented perspective from the static datasets. The required datasets are often scattered in multiple tables. Therefore, this step requires an understanding of the underlying database for making proper selection of tables. A process perspective requires time-stamped information of the activities undertaken to link the actions pertaining to a particular 'case' sequentially. Extracting this information may include merging additional tables by using appropriately defined unique identifiers. Therefore, a schematic wide view of all tables and their relationships with each other must be known.

5.1. Understanding available data

Moodle has a built-in logging system that tracks and stores navigational activity performed by the user (Rice and William, 2006). In this manner all student activities are recorded in a standard logging system. These logs are stored in relational database (MySQL or PostgreSQL). In total there are more than 346 tables in Moodle database. However, data extraction may not involve all of them, since all modules (quiz, assignment, lesson, surveys etc.) are not necessarily used in a course delivery.

The table 'logstore-standard-log' keeps tracks of all activities performed in Moodle. We can filter this table to extract more specific logs, such as by course, by participant, by day or by session (or any combination of these). These logs can give detailed views of activities performed by students during the course duration. For example, in a quiz activity, we can determine how much time students took to complete the quiz, their scores and the number of attempts among other similar bits of information. Figure 2 shows the high-level activities performed during the quiz, which starts with 'Start Quiz' step and ends in 'Submit Quiz'. A quiz can be viewed or reviewed multiple times, so there could be a self-loop in these steps.

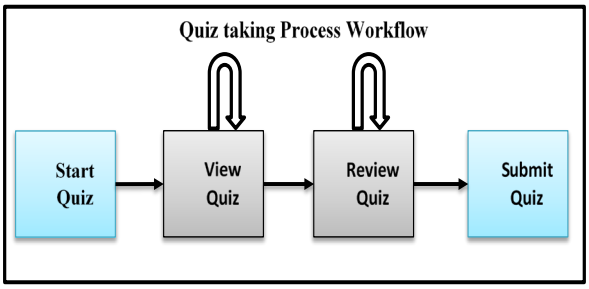

Figure 2. Basic activities in process of quiz-taking

Basic activities that are related to the quiz-taking process are stored in log table with time stamps. Log table provides only high-level picture of the process. In order to get more details of activities undertaken, additional tables that are associated with quiz-module need to be merged. The quiz module enables teachers to design quizzes as a part of students learning activity. A quiz consists of multiple questions of different types (e.g., calculated, multi-choice, description, essay etc.). Table 1 describes the tables related to quiz module.

Table 1. Details of all related tables in quiz-module.

Table Name	**Description**
mdl_quiz	Has quiz information like name, grading methods, number of attempts allowed, quiz time open, quiz time close, maximum grade, etc.
mdl_course-modules	Stores information about courses and its modules.
mdl_quiz-attempts	Stores information about quiz attempt, attempt start time, attempt finish time, grade in attempt, attempt state (e.g. in progress, complete, to do etc.)
mdl_quiz-grades	Stores final grades of students in quizzes.
mdl_question-usages	A unique id is assigned to each attempt made on a set of questions. A question usage is made up of a number of question_attempts.
mdl_question-attempts	Each row here corresponds to an attempt at one question, as part of a question-usage. A question-attempt will have some question-attempt-steps.
mdl_question	Stores questions that are in the quiz
mdl_question-attempt-steps	Stores one step in in a question attempt
mdl_logstore-standard-log	Stores information about activates performed in Moodle with their timestamp

Figure 3 represents a part of the database and displays six tables relevant to our running example. Table *quiz* contains information about quiz such as name, total marks, number of attempts allowed, etc. Table *quiz-grades* stores overall grades for each student on the quiz, based on their various attempts. Records in *quiz-grades* table refer to the quiz in the quiz table. One quiz can be attempted multiple times; so, the *quiz-attempts* table stores information

about each quiz attempts. Quiz's attempts are made of multiple question attempts. The *question-usages* is a bridging table that connects each quiz-attempt to a set of question attempts. *Question-attempts* table records information about an attempt at one question, which is further connected to *question-attempt-steps* table. Question and its state with the timestamp are stored in *question-attempt-steps* table.

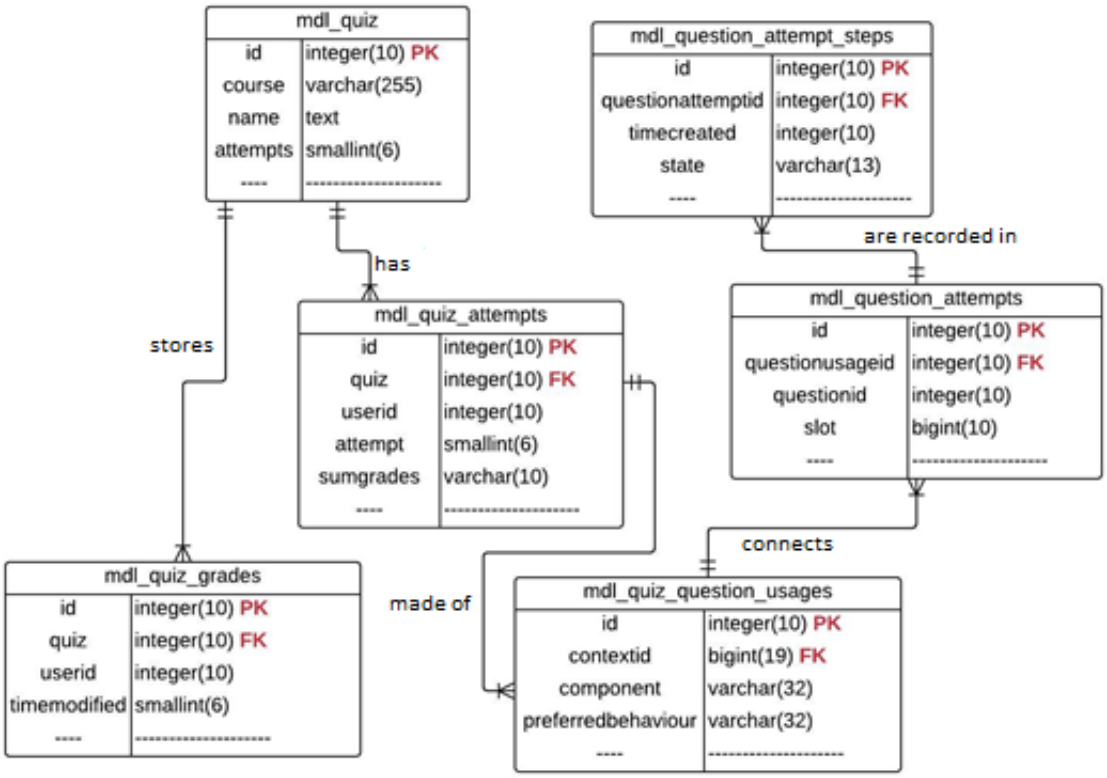

Figure 3. Entity relationship diagram of Moodle tables that are related to quiz-module.

Figure 4 shows state transition of a question that moves from one state to other state which is copied from Moodle docs[3]. A question starts as 'incomplete' and moves to 'complete' state after the student enters his response. The question now moves to grading – which can be graded manually or automated (as 'incorrect', 'partially correct' or 'correct'). A possibility of the student giving up and not entering the answer is also there, in which case the grade would specify it as 'gave-up'. Once grading is finished, each question is annotated with a comment.

[3] https://docs.moodle.org/dev/Overview_of_the_Moodle_question_engine

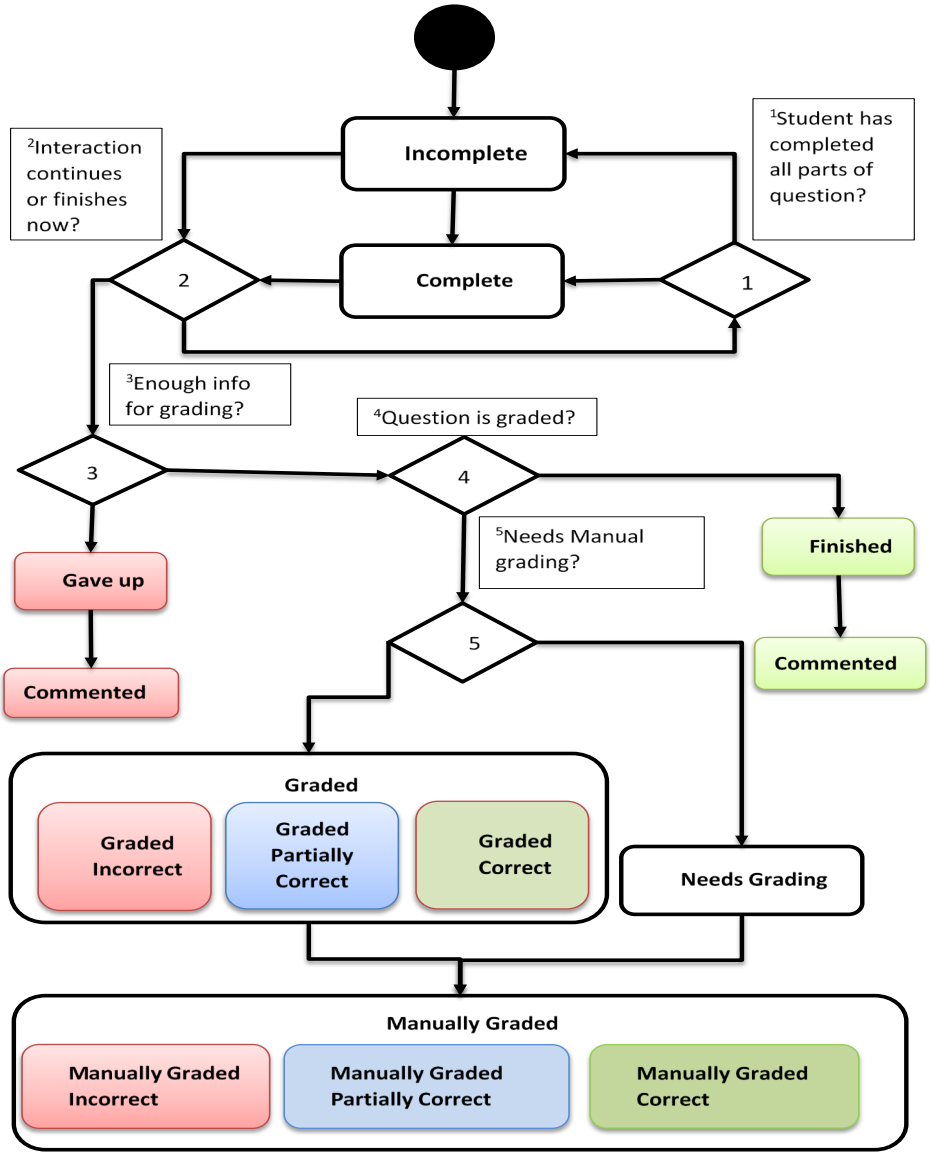

Figure 4. Moodle Question Engine overview.

5.2. Selection of process instance

This step is for selection of process instance and to make decisions about the granularity level. The selection of the process instance and granularity level have major impact on the quality of event log. Decision is to be made by considering the parent-child relationship between tables.

Main activities involved in quiz-taking process are start a quiz, view/review a quiz, view a question, view feedback (optional) and submit a quiz. There are five tables that records time stamps of events related to the quiz-taking process: *quiz, quiz-grades, quiz-attempts,*

question-attempts and *question-attempt-steps*. However, for building an event log each event

should relate to a case. Therefore, we need to integrate five tables in one table with column 'Case id' and all events should be referred to at least one case.

In our running example there are multiple options for selecting a process instance (case). For example, 'case' could be per student that allows us to obtain different patterns student followed while taking a quiz, or 'case' could be per question to obtain a process occurred during exercises. Alternatively, it could be at quiz level, that is, at a session where quiz occurred during the course where students worked on same quiz and performed different activities. Since our scope is limited to the quiz level, we have next analysed the activities performed during the quiz session. Therefore, we choose case per record in the quiz table. We list all the events that are associated to a quiz (case). Figure 5 shows all the events that could be related to one case.

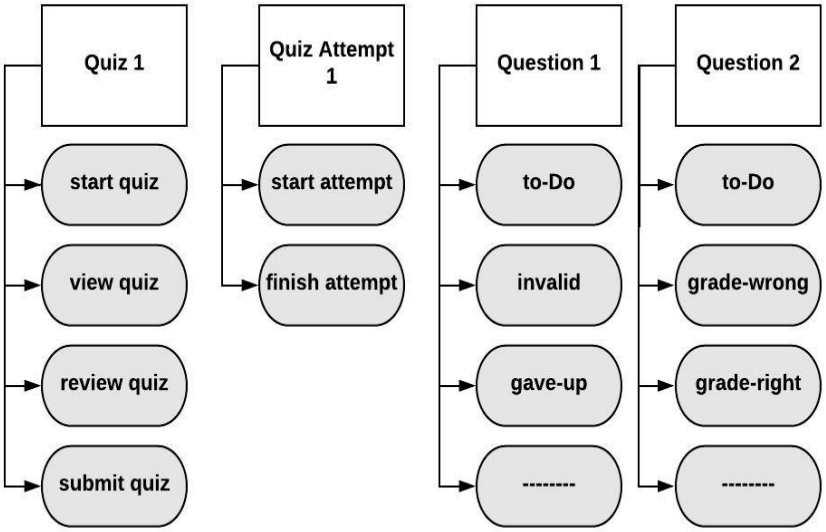

Figure 5. Activities associated with quiz-module tables.

5.3. Selection of activities and related attributes

Table *quiz* has four timestamps per record. If only the *quiz* table is considered, then there will only four events per case and all information about the quiz attempts and questions states will be unused. It will result in a sequential process consisting of following steps; 'start quiz',' view quiz',' review quiz' and 'submit quiz' shown in Figure 2.

To develop a process model that consists of more activities we need to consider other tables as well. By using the domain knowledge such as cardinality, references are made to link other tables. If *quiz*, *quiz-attempts* and *question-attempt-steps* tables are considered then all the events or subset of events with their time stamps can be considered along their relevant activities. Table 2 shows list of activities and attributes that are associated with selected tables.

After selection of process instance and granularity level, all relevant activities have to be selected. Activities have to be grouped based their time stamps. In addition, those attributes that give more information about the process instance or about the activity are identified and

selected. For example, activity performed by whom and where (location) or a role that is responsible for the activity.

Table 2: List of events associated with quiz module

Case: Quiz		
Table Name	**Activity**	**Attributes**
mdl_quiz	start-quiz view-quiz review-quiz submit-quiz	quiz-grades student-id quiz-status
mdl_quiz-attempts	start-quiz-attempt finish-quiz-attempt	quiz-attempt-state quiz-attempt-grade student-id
mdl_quiz-attempt-steps	state-to-do state-finish state-invalid state-complete state-unprocessed state-gaveup state-needsgrading state-grade-write state-grade-wrong state-grade-partial	question-grade student-id

6. Event log challenges

Educational technologies have facilitated large volumes of data from heterogeneous sources to be stored in different formats and at different granularity levels (Calders and Pechenizkiy, 2012; Romero et al., 2010). Collection and integration of such voluminous data scattered over the learning platform, is not a trivial task. Moreover, precision is to be considered in filtering of event activities for process models to discover specific behaviours. Real life data extracted from business domains (e.g., LMS) comprise multiple and parallel activities that may or may not be very frequent. Learner behaviours have much variability, therefore, filtering steps to form event logs are much harder compared to other business processes (e.g., payment at checkout kiosks). Following sub-sections highlights challenges in data extraction for creating a process-oriented event log.

6.1. Complex and voluminous data

Moodle database comprises more than 346 tables. Depending upon the case granularity level, multiple tables have to be queried to get information about one activity. To select cases and accordingly write queries that join a number of tables requires in-depth domain knowledge. Fortunately, Moodle is open source and has plenty of online resources. The Moodle developer community is very responsive to forum questions and they regularly contribute to the online library to help users understand the complexities of the database.

6.1.1. Moodle documentation -database schema

The technical documentation from the online Moodle resources provided an in-depth view of the underlying database schema (i.e., purpose of the table, column names and description,

keys, etc.). With more than 346 tables overall, of which many are not relevant to the problem in hand, the documentation can help to identify sub-schema applicable to the context being investigated.

6.1.2. Moodle online community

An online community forum provides a platform for people having shared interest on Moodle know-how to interact with each other. The community members are quick to respond to problems posted on the forum. One forum group ADH (or the 'ad-hoc contributed my SQL queriers' group) in particular supports users to extract data More than 1000 SQL queries are posted on ADH to enable users extract information from a MySQL database. This forum group provides many resources to assist in data extractions from database. Figure 6 illustrates a query to list logged users from last 120 days.

```
SELECT id,username,FROM_UNIXTIME(`lastlogin`) AS days
FROM `prefix_user`
WHERE DATEDIFF( NOW(),FROM_UNIXTIME(`lastlogin`) ) < 120
```

Figure 6. A snapshot of a query given in ad-hoc contributed reports in Moodle discussion forum

6.2. Dealing with many-to-many relationships

Quiz-taking is a relatively small process considering the bigger picture of other activities undertaken by students during a course. We noted that the quiz-taking process involved over 5 tables with table relationships not limited to simple one-to-one relationships. Therefore, data extractions have to be handled logically to avoid ambiguous case definitions.

Our running example deals with activities performed by a student during the quiz-taking process. The first challenge is selection of the case-id, since the case-id must be related to end-to-end process activities. For example, quiz-taking process broadly involves three steps: 'start quiz', 'view quiz' and 'submit quiz'. Figure 7 is a simple representation of event log activities performed by a student during a quiz-taking process for a particular quiz. Activities like 'quiz-started', 'quiz-submitted' and 'quiz-viewed' are repeated many times for the selected quiz (which is evident from the event time stamps). Within this context, by selecting 'Quiz-id' as case, all activities that belong to the quiz are transformed to a process map (shown in right side of Figure 7) after applying process mining on the event log.

Case-id	Timestamp	Activity	Grade
	2017-02-27 13:05:09		3.75
	2017-02-27 13:05:09	quiz_viewed	3.75
	2017-02-27 13:05:40	quiz_started	3.75
	2017-02-27 13:07:43	quiz_viewed	3.75
	2017-02-27 13:14:16	quiz_submitted	3.75
Quiz-Id	2017-02-27 13:17:37	quiz_viewed	5
	2017-02-27 13:17:37	quiz_started	5
	2017-02-27 13:19:25	quiz_viewed	5
	2017-02-27 13:27:21	quiz_viewed	5
	2017-02-27 13:33:26	quiz_submitted	5

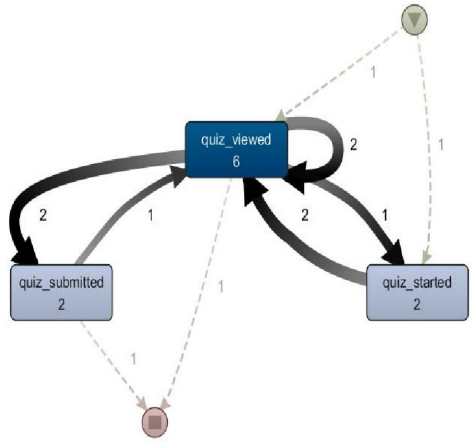

Figure 7. An example of event log where case id is selected as per quiz and corresponding process model view.

The process map in Figure 7 shows that a student started and submitted the quiz twice; but this activity has been performed on multiple quiz-attempts. This situation is known as divergence of data (Lu, 2013) since it occurs as a result of many-to-many relationships (i.e., each student can attempt a quiz many times and each quiz are attempted by many students). Figure 8 shows three different perspectives of quiz-taking process, (1) quiz-taking as a process, (2) timeline view of the activities performed during the quiz-taking process and (3) process mining view of the quiz-taking process. Therefore, process map shown in Figure 7 does not fully demonstrate how the process occurred. So, the real challenge is the perspective taken in the investigation.

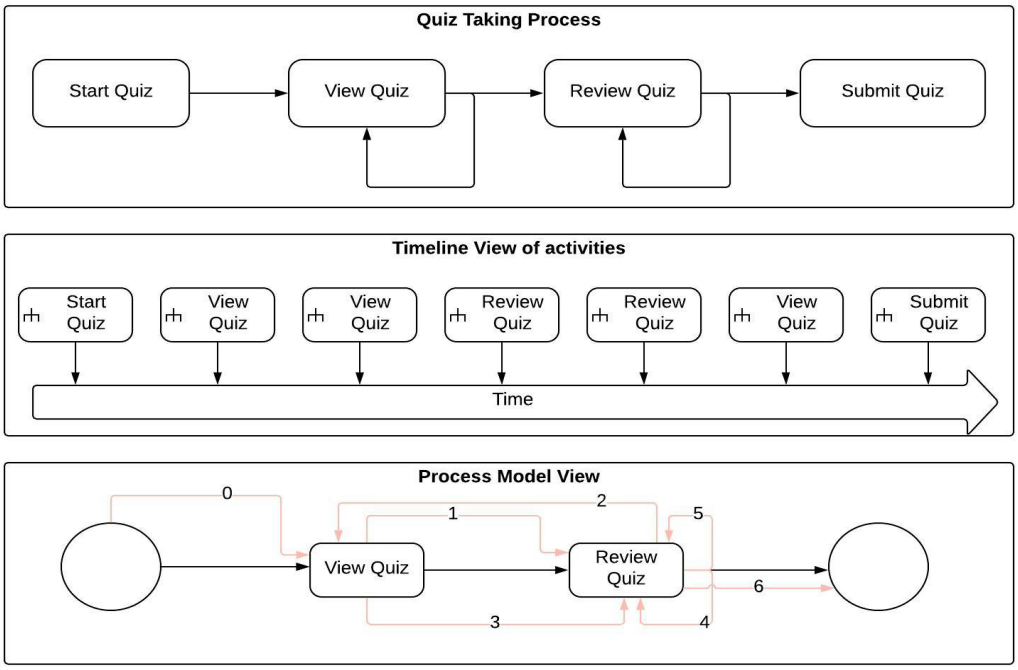

Figure 8. Divergences of cases visualised reality vs. process mining view of activities

Divergence can be resolved by adding more granularities to activities. Like in given example if we just consider the quiz-taking event then we can face divergence issue, but if we add quiz-attempt or add question level granularity then this problem can be resolved. By changing the perspective, we get a process model (shown in Figure 9) that reflects more details of the quiz-taking process. It is clear from the process map that student viewed and submitted quiz in two different attempts and this was evident by adding to the granularity level in the event log.

CaseID	EventID	Timestamp	Activity	Grade
1	1	2017-02-27 13:05:09	quiz_viewed	3.75
1	1	2017-02-27 13:05:09	quiz_started	3.75
1	1	2017-02-27 13:05:40	quiz_viewed	3.75
1	1	2017-02-27 13:07:43	quiz_viewed	3.75
1	1	2017-02-27 13:14:16	quiz_submitted	3.75
1	2	2017-02-27 13:17:37	quiz_viewed	5
1	2	2017-02-27 13:17:37	quiz_started	5
1	2	2017-02-27 13:19:25	quiz_viewed	5
1	2	2017-02-27 13:27:21	quiz_viewed	5
1	2	2017-02-27 13:33:26	quiz_submitted	5

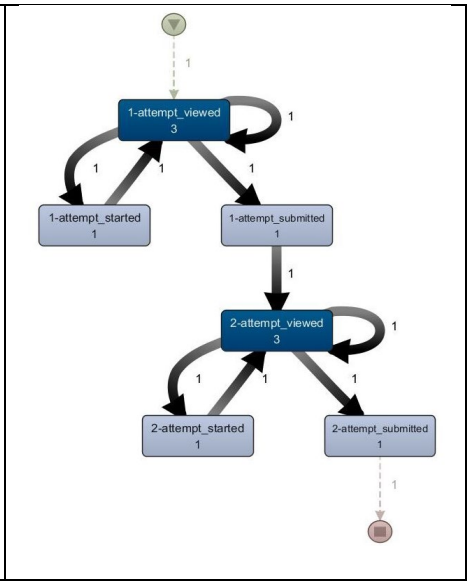

Figure 9. An example of event log where case-id is selected as per quiz and event is selected as quiz attempt number and corresponding process model view of event log

6.3. Missing time stamps

In the event log, events are ordered per case. For each case we merge time stamped data from multiple tables. However, we faced missing time stamp issues in the event log. A small fragment of the quiz table (Figure 10) shows each quiz with two time stamp records, namely 'timeopen' (i.e., the time quizzes were available to students for attempting the quiz) and 'timeclose' (i.e., the time after which the quiz will not be available). As is evident in Figure 10, some quizzes have specific set time limits while others have no time limit; therefore, we have set the 'timeclose' to null value to represent no time limit. Following sub-sections detail situations having missing time stamps for events.

6.3.1. Missing quiz-open time and quiz-close time

Missing quiz-open and quiz-close times situations occur when the quiz has no predefined time bound, or the quiz is available and visible from the beginning of the course. Some instructors often setup non-compulsory practice quizzes to help students learn some subject concept; and since quizzes are not compulsory, it is up to the students to attempt them or not. Most of the time practice quizzes do not contribute towards the final mark.

	quizid	name	timeopen	timeclose	grade	timelimit	overduehandling	grademethod
0	66234	Test Yourself - Lab 3	1970-01-01 12:00:00+12:00	1970-01-01 12:00:00+12:00	10.00000	0	autoabandon	1
1	66229	Mastery Test 2	2017-03-06 00:30:00+13:00	2017-03-20 19:00:00+13:00	10.00000	2	autosubmit	1
2	66228	Mastery Test 1	2017-02-27 09:00:00+13:00	2017-03-12 23:00:00+13:00	10.00000	2	autosubmit	1
3	66232	Test Yourself - Lab 1	1970-01-01 12:00:00+12:00	1970-01-01 12:00:00+12:00	10.00000	0	autoabandon	1
4	66233	Test Yourself - Lab 2	1970-01-01 12:00:00+12:00	1970-01-01 12:00:00+12:00	10.00000	0	autoabandon	1
5	66230	Mastery Test 3	2017-03-13 00:30:00+13:00	2017-03-27 03:00:00+13:00	10.00000	2	autosubmit	1
6	69389	Test Yourself--Lab 4	1970-01-01 12:00:00+12:00	1970-01-01 12:00:00+12:00	7.00000	0	autoabandon	1
7	66231	Mastery Test 4	2017-03-20 00:30:00+13:00	2017-04-03 03:00:00+12:00	10.00000	2	autosubmit	1

Figure 10. Some records of the quiz table.

We were interested to see student groups who attempted practice quizzes. With missing quiz 'timeopen' and 'timeclose' on the generated process map, we could not get a complete view on students' activities. Having common time allocations for quiz opening and its subsequent closing can help generate process maps which can be used to analyse student behaviours during the quiz sessions. But in situations where students could do practice quizzes at their convenience, we cannot analyse all students' behaviour from one common session.

6.3.2. Missing quiz-close time only

This situation occurs when there are no bounds are set to quiz attempts. The quiz can be attempted at any time during the course or as long as it is visible to students.

6.3.3. Missing attempt-finish time

A quiz can be in following four different states (refer Figure 11). These are:

- *In-progress:* Quiz has started but not yet finished. In this state, the 'attempt-finish-time' is null.

- *Finished:* The quiz attempt is submitted. The 'attempt-finish-time' is the timestamp when the student submits.

- *Abandoned:* if quiz is not submitted on time then attempt is considered as abandoned. The state is again directed to 'in-progress' and 'attempt-finish-time' is null.

- *Overdue:* In some cases, students are given grace-period time to submit the quiz after the set time. If quiz is submitted within the given grace period time, it changes to 'finished' state otherwise it remains in the 'abandoned' state.

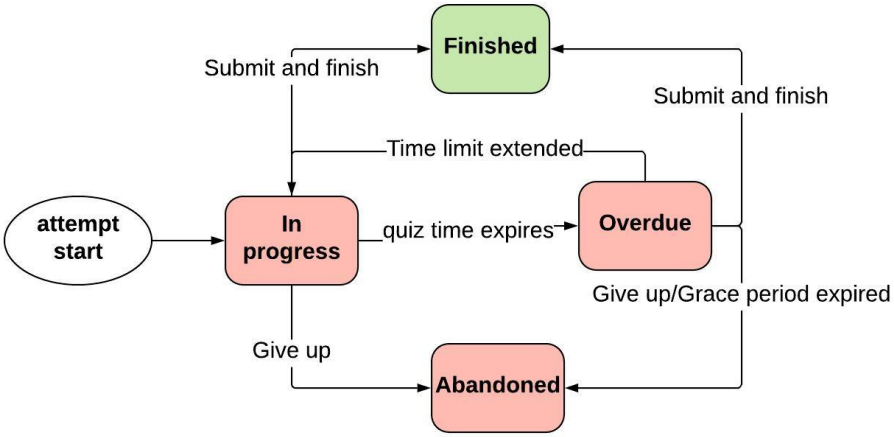

Figure 11. Different states of the quiz. 'Attempt-finish-time' will be missing if quiz is in 'in-progress' state.

In scenarios where the finish time is missing and there is no bound on the quiz closing time, the students might attempt the quiz at a much later time. Therefore, event logs which capture these late attempts would refer to long-running processes. One possible solution is to remove those cases which are incomplete, or the finish time is missing especially when the average duration of that process is short. However, this is not an optimal solution since it will result in loss of useful information like behaviour of students who did not complete quiz. To get proper picture of the process we need to involve the state of the quiz as well. For example, where finish time is missing, and state is 'abandoned' or 'overdue', we can add another activity called 'abandoned-quiz' for such records. Therefore, the end of quiz will be related to either of the two activities: 'submit-quiz' or 'abandoned-quiz'. Both end states are illustrated in Figure 12.

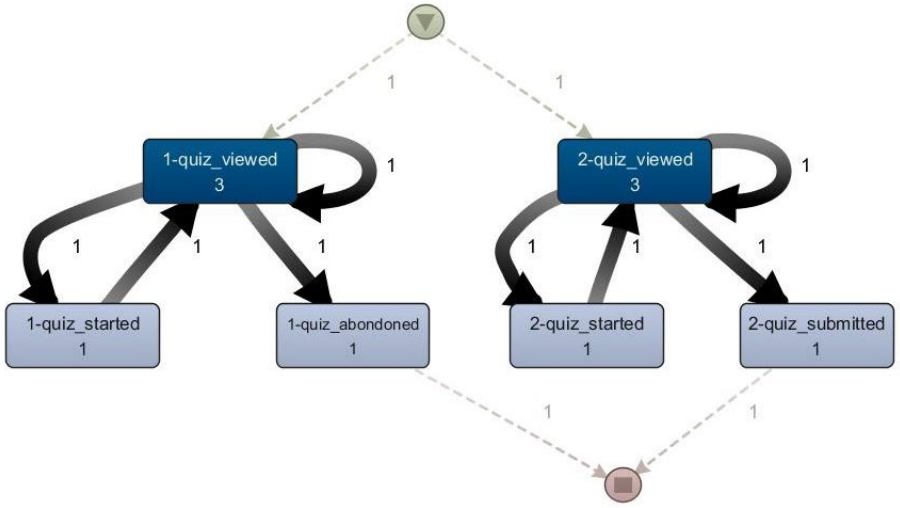

Figure 12. Process map presenting status of two quiz attempts by a student – one of which is 'abandoned'

6.4. Granularity

Process with large number of activities results in a fine-granular event log. Like other high-tech systems, Moodle's events are generated automatically to support information system. Fine-granular events are hard to handle as it results in 'spaghetti' like process models which are difficult to interpret. It is beyond the human cognitive system to understand the process models generated from fine granular event log. It is quite challenging to decide which level of granularity one should go. If we go for higher level abstraction, we might get simple process models, but it might overlap with low level activities. One has to decide to fill the gap between high level abstraction to the low-level events that are relevant to the end user who is interested to understand or improve the processes.

From our running example it is evident that selection of the relevant events and activities was not easy. Each activity can be broken down to many sub-activities corresponding to different states. Therefore, if all activities in 'question-attempt' level are selected, then a complex process model would be generated, and which would be hard to analyse. One 'question-attempt' could go in sixteen different states as previously shown in Figure 4. Every state could be considered as an activity as the time stamps are also recorded. If quiz is selected as process instance, then granularity level till question attempt will result in a 'spaghetti' process model. Therefore, we bound granularity level to 'quiz-attempt' to avoid getting complex process models. An alternate solution was to aggregate the related events for example, 'gradedincorrect', 'gradecorrect', 'gradepartialcorrect', to one event that is 'grade' if we are not interested in the outcome of grade but more interested in the timestamp of 'grade'.

6.5. Collateral events

Collateral events are multiple events that are essentially referring to one particular step in a process within a case (Suriadi et al., 2017). Moodle logs very low level of activities. For example, when student submits a quiz online there are states that capture the state of the quiz or question, also there are other states triggered based on the outcome of the grade (e.g., a 'complete' question state can further trigger events like 'gradecorrect', 'gradeincorrect' or 'gradepatiallycorrect' with duplication of time stamps or with difference of very short time period). Few examples are shown in Figure 13.

	id	studentid	questionattemptid	state	grade	actiontime
0	95517751	73604	33561110	todo	None	2017-03-14 16:55:14+13:00
1	95518459	73604	33561110	invalid	None	2017-03-14 17:00:00+13:00
2	95519186	73604	33561110	complete	1.0000000	2017-03-14 17:04:42+13:00
3	95524330	73604	33561110	gradedright	1.0000000	2017-03-14 17:46:02+13:00
4	95913934	73604	33698674	todo	None	2017-03-18 15:33:28+13:00

Figure 13. Some records of the quiz-grade table showing collateral events with difference of short time period.

These events are independent of each other with different labels but are repeated with difference of short time period to show that one important step of a process. These kinds of collateral events can make the process model unnecessarily complex and does not give useful

insights about process. We followed recommendation by Suriadi et al. (2017) to merge such activities into a single activity and consider timestamp as either earliest or the latest.

6.6. Partial or incomplete traces

This refers to the situation where one or more events are missing in a trace. To refer to an example, there are many ways to submit assignments. Some students prefer to submit online, some submit hard copy, or some others could email their assignment to the instructor. For those who submit hard copy or email separately, the 'submit-time' remains null in table *assignment*. Such missing events results in a process model represent partial reality, since submit assignment event has occurred in reality but has not been logged in the database. There are methods that can filter incomplete traces, but it will result in loss of information.

7. Lessons learned

Analysing students' data and detecting patterns of interaction from learning management systems have gained much attention among the learning analytics and process mining research communities. Learning analytics emphasizes on use of education data to co-relate students' online behaviour with their academic performance so as to provide timely support to students; however, there is no focus on the process of learning as a whole. Research shows that process mining provides robust methods to leverage temporal data and inform us on dynamic behavioural patterns. Like any other data-driven approach, process mining takes event log as an input to produce process-related information and provide visual representation of the process for further analysis. The quality of process mining analysis depends on the accuracy and completeness of the event logs. The input data is often scattered and stored in enterprise systems that are static and not process-oriented. Therefore, identifying process related data, extraction and conversion of static data to the required format to build an event log is a complex task.

This paper has described multiple challenges faced in constructing event logs from process unaware LMS. Following are the lessons learned.

- Moodle database is complex and overwhelming; therefore, it is not possible to understand the entire database schema at once. An advisable approach would be to focus on smaller modules or subsystems; since the corresponding sub-schema represents a smaller database and has more contextual relevancy.
- Investigate different pathways, to identify all possible activities that are related to an event. This will help to find all necessary data that can be part of an event log, which can then be extracted. However, in doing any extractions, we must particularly take care of existing parent-child table relationships.
- Judiciously scope the problem in line with the motivating research questions. Select all the events and activities that are of interest to the end user who requires them rendered in a process model.
- Make a distinction on defining the process instance at parent or child level and consider the implications of this selection. If the selected process instance is at parent level, there might be low-level event at child level which needs to be aggregated or ignored (if not relevant) to avoid complex process model that is hard to comprehend.
- When selecting events or activities, focus should also be given to the other attributes that might be helpful to give more insights. An example of this in our study, was the 'state' attribute, which helped us to make a new event.

- Avoid repetition of collateral events that are referring to one important event that matters in process model.
- The context of the data is necessary to be considered when interpreting the results of process mining.

8. Contributions

Our study has demonstrated operational measures to include relevant activity data in an event log from a process-unaware database. This study reports challenges and lessons learned while extracting static data from non-process-oriented systems that do not follow the format required for process mining projects. We outlined the process of extracting data and their impact on the quality of event data.

The dataset examined is gathered from Moodle (LMS) used in one tertiary institution in New Zealand. Moodle is a cost-effective LMS that exemplifies tracking, reporting, administering and delivering educational courses (Ifinedo, et al. 2018). Educational institutions often hesitate to upgrade or use a different LMS due to the finances, time and effort expended in making the LMS functional (Such et al., 2017). Our study has provided a much-needed internal view of this very popular learning platform and highlighted how process-aware event logs can be prepared by educational technology specialists for educational process mining purposes.

We demonstrated a running example with student data extracted from activity logs when students engaged in quiz-taking process. The study context has been narrowed down to a simple quiz-taking example to capture sub-schema (comprising five tables) from the underlying database schema (comprising 346 tables). Having domain knowledge and understanding of the data structure are the first steps before a case can be identified. Secondly, the selected dataset transformation into event logs is aligned with possible process instances, related activities and attributes. The consequences of such selections are explained. We have focussed on quality issues that occur during data extraction and its subsequent alignment to avoid identification of ambiguous cases. We addressed possible optimal solutions to construct the desired process model as a result of transformation and selection of proper granularity level of the events. The context of the data is necessary to be considered when interpreting the results of process mining.

While this study demonstrated a quiz-taking process which is only one of the learning activities performed during the course of study, there are other learning activities too which can be used. Future studies can follow a similar educational process mining approach by accommodating other learning activities like assignment submission, reading online resources, watching video lectures, etc. This article contributes to the fields of learning analytics and process mining by providing lessons learned with the extraction and conversion of process-unaware data to event logs for the purpose of analysing online education data. Detailed insights are shared regarding quality issues faced with education data in general and more specifically how process mining can be applied to other historical structure-oriented datasets.

References

Adnan, N.H. and A.D. Ritzhaupt, (2018). Software Engineering Design Principles Applied to Instructional Design: What can we learn from our Sister Discipline? *Tech Trends*, 62(1): 77-94.

Avella, J. T., Kebritchi, M., Nunn, S. G., and Kanai, T. (2016). Learning analytics methods, benefits, and challenges in higher education: A systematic literature review. *Online Learning*, 20(2):13–29.

Baesens, B, Bapna R., Marsden, J.R., Vanthienen, J. and Zhao. J.L. (2016). Transformational issues of big data and analytics in networked business. *MISQ*, 40(4): 807-818. https://doi.org/10.25300/MISQ/2016/40:4.03

Bose, R.J.C., Mans, R.S., and van Der Aalst, W.M. (2013). Wanna improve process mining results? In *Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on*, pages 127–134. IEEE.

Calders, T. and Pechenizkiy, M. (2012). Introduction to the special section on educational data mining. *ACM Sigkdd Explorations Newsletter*, 13(2):3–6.

Calvanese, D., Montali, M., Syamsiyah, A., and van Der Aalst, W. M. (2015). Ontology-driven extraction of event logs from relational databases. In *International Conference on Business Process Management*, pages 140–153. Springer.

Chang, C.-S., Liu, E. Z.-F., Sung, H.-Y., Lin, C.-H., Chen, N.-S., and Cheng, S.-S. (2014). Effects of online college student's internet self-efficacy on learning motivation and performance. *Innovations in education and teaching international*, 51(4):366–377.

Ferguson, R. (2012). Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, 4(5/6):304–317.

Hwang, G-J., Chu H.-C., and Yin C. (2017). Objectives, methodologies and research issues of learning analytics, interactive learning environments. *Interactive Learning Environments*, 25(2):143–146.

Hsiao, C., Huang, J. C., Huang, A. Y., Lu, O. H., Yin, C., and Yang, S. J. (2018). Exploring the effects of online learning behaviours on short-term and long-term learning outcomes in flipped classrooms. *Interactive Learning Environments*, pages 1–18.

Hwang, G.-J., Hsu, T.-C., Lai, C.-L., and Hsueh, C.-J. (2017). Interaction of problem-based gaming and learning anxiety in language students' English listening performance and progressive behavioural patterns. *Computers & Education*, 106:26–42.

Ifinedo, P., Pyke, J. and Anwar, A. (2018). Business undergraduates' perceived use outcomes of Moodle in a blended learning environment: The roles of usability factors and external support, *Telematics and Informatics*, 35(1): 93-102, ISSN 0736-5853, https://doi.org/10.1016/j.tele.2017.10.001.

Jans, M. and Soffer, P. (2017). From relational database to event log: decisions with quality impact. In *Teniente E., Weidlich M. (eds) Business Process Management Workshops. BPM 2017. Lecture Notes in Business Information Processing, 308*:588–599. Springer, Cham.

Kim, W., Choi, B.-J., Hong, E.-K., Kim, S.-K., and Lee, D. (2003). A taxonomy of dirty data. *Data mining and knowledge discovery*, 7(1):81–99.

Kwon, O., Lee, N. and Shinb. (2014). Data quality management, data usage experience and acquisition intention of big data analytics, *International Journal of Information Management*, 34:387-394

Lismont, J., Vanthienen, V., Baesens, B. and Lemahieu, W. (2017). Defining analytics maturity indicators: A survey approach, *International Journal of Information Management*, 37(3):114-124, ISSN 0268-4012, https://doi.org/10.1016/j.ijinfomgt.2016.12.003.

Lu, X. (2013). Artifact-centric log extraction and process discovery. *Unpublished master's thesis, Eindhoven University of Technology*.

Peña-Ayala, A. Learning analytics: Fundaments, applications, and trends.

Pérez-Castillo, R., Weber, B., de Guzmán, I. G.-R., Piattini, M., and Pinggera, J. (2014). Assessing event correlation in non-process-aware information systems. *Software & Systems Modelling*, 13(3):1117–1139.

Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., and Rosati, R. (2008). Linking data to ontologies. In *Journal on data semantics X*, 133–173. Springer.

Rahm, E. and Do, H. H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Eng.*

Bull., 23(4):3–13.

Rice, W. H. and William, H. (2006). *Moodle*. Packt Publishing Birmingham.

Romero, C. and Ventura, S. (2013). Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(1):12–27.

Romero, C., Ventura, S., Pechenizkiy, M., and Baker, R. S. (2010). *Handbook of educational data mining*. CRC press.

Selig, H. (2017). Continuous event log extraction for process mining (dissertation) retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-210710.

Siemens, G. (2013). Learning analytics: The emergence of a discipline. *American Behavioural Scientist*, 57(10):1380–1400.

Siemens, G. and Long, P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE review*, 46(5):30.

Such, Brenda L. R.; Ritzhaupt, Albert D.; and Thompson, George S. (2017) "Migrating learning management systems: A case of a large public university," *Administrative Issues Journal*: 7(2):57-69.

Suriadi, S., Andrews, R., ter Hofstede, A. H., and Wynn, M. T. (2017). Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. *Information Systems*, 64:132–150.

Tax, N., Sidorova, N. & van der Aalst, W.M.P. Journal of Intelligent Information Systems (2019) 52: 107. https://doi.org/10.1007/s10844-018-0507-6

Umer, R., Susnjak, T., Mathrani, A., and Suriadi, S. (2017). Prediction of students' dropout in MOOC environment. *International Journal of Knowledge Engineering*, 3(2).

Vahdat, M., Ghio, A., Oneto, L., Anguita, D., Funk, M., and Rauterberg, M. (2015). Advances in learning analytics and educational data mining. *Proc. of ESANN2015*, 297–306.

van Der Aalst, W., Adriansyah, A., De Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., Bose, J. C., van Den Brand, P., Brandtjen, R., Buijs, J., et al. (August 2011). Process mining manifesto. In *International Conference on Business Process Management*, 169–194. Berlin, Heidelberg, Springer.

van Der Aalst, W. M. (2015). Extracting event data from databases to unleash process mining. In *BPM-Driving innovation in a digital world*, 105–128. Springer.

van Der Aalst, W. M. (2016). *Process mining: data science in action*. Springer.

van Der Aalst, W. M., Reijers, H. A., and Song, M. (2005). Discovering social networks from event logs. *Computer Supported Cooperative Work (CSCW)*, 14(6):549–593.

van Dongen, B. F. and Shabani, S. (2015). Relational XES: Data management for process mining. In *CAiSE Forum*, pages 169–176.

Verbeek, H., Buijs, J. C., van Dongen, B. F., and van Der Aalst, W. M. (2010). XES, XESame and PROM 6. In *Forum at the Conference on Advanced Information Systems Engineering (CAiSE)*, 60–75. Springer.

Wong, B. T.-M., Li, K. C., and Choi, S. P.-M. (2018). Trends in learning analytics practices: a review of higher education institutions. *Interactive Technology and Smart Education*, 15(2):132–154.

Yang, J. C., Quadir, B., Chen, N.-S., and Miao, Q. (2016). Effects of online presence on learning performance in a blog-based online course. *The Internet and Higher Education*, 30:11–20.