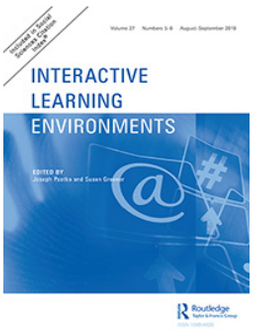


Interactive Learning Environments

ISSN: 1049-4820 (Print) 1744-5191 (Online) Journal homepage: https://www.tandfonline.com/loi/nile20

User behavior pattern detection in unstructured processes – a learning management system case study

David Codish, Eyal Rabin & Gilad Ravid

To cite this article: David Codish, Eyal Rabin & Gilad Ravid (2019) User behavior pattern detection in unstructured processes – a learning management system case study, Interactive Learning Environments, 27:5-6, 699-725, DOI: 10.1080/10494820.2019.1610456

To link to this article: https://doi.org/10.1080/10494820.2019.1610456

Published online: 01 May 2019.

INTERACTIVE LEARNING ENVIRONMENTS
2019, VOL. 27, NOS. 5–6, 699–725
https://doi.org/10.1080/10494820.2019.1610456

Routledge
Taylor & Francis Group

User behavior pattern detection in unstructured processes – a learning management system case study

David Codish [a], Eyal Rabin[b] and Gilad Ravid[a]

[a]Industrial Engineering and Management, Ben-Gurion University, Beer-Sheva, Israel; [b]Management, Science and Technology, Open University of the Netherlands, Valkenburgerweg, Netherlands

ABSTRACT
Process mining methodologies are designed to uncover underlying business processes, deviations from them, and in general, usage patterns. One of the key limitations of these methodologies is that they struggle in cases in which there is no structured process, or when a process can be performed in many ways. Learning Management Systems are a classic case of unstructured processes since each learner follows a different learning process. In this paper, we address this limitation by proposing and validating the user behavior pattern detection (UBPD) methodology which is based on detecting very short user activities and clustering them based on shared variance to construct a more meaningful behavior. We develop and validate this methodology by using two datasets of unstructured processes from different implementations of a learning management system. The first dataset uses a gamified course where users have the freedom to choose how to use the system, and the second dataset uses data from a massive online open course, where again, system usage is based on personal learning preferences. The key contribution of the methodology is its ability to discover user-specific usage patterns and cluster users based on them, even in noisy systems with no clear process. It provides great value to course designers and teachers trying to understand how learner interact with their system and sets the foundation for additional research in this class of systems.

ARTICLE HISTORY
Received 31 October 2018
Accepted 1 April 2019

KEYWORDS
Learning analytics; learning management systems; process mining; spaghetti processes; pattern detection; gamification

Introduction

Process mining is a method used to discover underlying business processes, or deviations from such processes, through the analysis of system log files, which represent the actual behavior of users within a system (Van den Beemt, Buijs, & Van der Aalst, 2018; van der Aalst et al., 2012; van der Aalst & Weijters, 2004). While process mining has been successful in discovering well-structured processes, it has been less successful in non-structured processes, resulting in spaghetti-like process maps which are hard to interpret and use (Chinces & Salomie, 2015; Li, Bose, & van der Aalst, 2010). Well-structured processes are processes that are followed by all users, while less structured processes allow users to perform them in different ways. These deviations from the process may, or may not, be acceptable from a designer's point of view.

Structured processes are common and desired in business environments where employees are expected to follow a certain flow of actions to achieve an objective such as the completion of a purchase order, reporting their monthly working hours, or filling a reimbursement form. Despite each of the examples above having deviations in their processes such as in the case of a purchase order that

CONTACT David Codish codishd@post.bgu.ac.il

does not match company guidelines or a reimbursement request for a large sum, they can still be considered structured, as even these deviations from the processes are well-defined and structured.

Unstructured processes, on the other hand, have no single process to follow, and users can follow any course of action at any point in time. Two such cases are the focus of this article. First, cases where there is no clear process at all, such as in learning management systems (LMS), news consumption sites or a social networking application where there is no point in searching for an overall process since it does not exist. Second, processes which may have existed, but due to a change in the system, such as adding gamification, the process is no longer structured. Gamification is the use of game design elements in a non-gaming environment (Deterding, Dixon, Khaled, & Nacke, 2011) with the intent of increasing user engagement (Kankanhalli, Taher, Cavusoglu, & Kim, 2012; Werbach, 2014), hedonic motivation (Lowry, Gaskin, Twyman, Hammer, & Roberts, 2013; Van der Heijden, 2004), or achieving other business goals (Hamari & Koivisto, 2015). The gamification of information systems involves adding different game elements to existing systems which, as a result, changes the way users interact with them. For example, granting points or badges for specific actions is expected to incentivize these actions, and including user profiles in an application is expected to increase social interaction. Gamification typically involves adding several game elements to a system, and given the voluntary nature of gamification, this means that different users would interact with them differently. As a result, even streamlined processes become less structured, making process mining less beneficial. Gamification of information systems is becoming common within organizations and thus, should receive special interest from system developers and researchers.

Although most process mining methods are not suitable for less-structured processes such as in the case of gamified systems, some methods can still address these limitations. For example, sequence mining (Srikant & Agrawal, 1996), episode mining (Mannila, Toivonen, & Verkamo, 1997), and the apriori and generalized sequential pattern (GSP) methods (Agrawal & Srikant, 1994; Srikant & Agrawal, 1996) are designed to detect recurring patterns, or sub-processes, within an overall noisy process. The sequence hierarchy discovery algorithm (Greco, Guzzo, & Pontieri, 2005) attempts to detect sub-processes and reconstruct them into the full process, assuming it exists. However, these algorithms assume that a process exists and that all users follow it similarly, which is not always true. Our research question is thus: Within a non-structured process or system, can we automatically identify recurring user-level behavior patterns and perform user clustering based on these patterns?

In this paper, we develop and validate the user behavior pattern detection (UBPD) algorithm employing system logs to automatically detects user behavior patterns and cluster users based on these patterns. We define user behavior patterns as usage patterns that certain users perform more, or less than, others. Both case studies used in this paper are based on educational settings, thus from an educational point of view, behavior patterns can easily be interpreted as learner behavior patterns. Our key contributions in this paper are the development of an automated end-to-end process to detect structured behavior patterns within an otherwise non-structured environment. An additional benefit is the algorithm's ability to detect these sub-processes at the user level, while most existing methods search for sub-processes at the system level. For instance, if half of the users perform task A and then task B and half perform task B and then task A, a methodology seeking for patterns at the system level, would not detect this as a pattern, while UBPD would. The discovered user behavior patterns can be used for additional user clustering or a deeper understanding by system designers as to how their system is being used. Its main applicability is in cases in which there is no structured process, or no process at all, such as LMSs where learners typically log in to perform a specific task and then log out and news websites where users consume news in no particular order. With the advent of digital footprints analysis (Golder & Macy, 2014; Lambiotte & Kosinski, 2014; Williams & Pennington, 2018), where digital records of a person from many sources are combined to create a user profile, such an approach can be useful since data would be unstructured by nature and difficult to analyze.

The algorithm presented is based on a few stages. The first is a data preparation stage in which data are collected from various log files and organized. A sequence mining approach is used to detect the most frequent sequences of actions and organizes them at the user level. The clustering of these sequences per user is done through exploratory factor analysis (EFA), which results in factors representing user behavior patterns. Last, causal nets are used to construct a representation of these factors graphically. Two data sets from different LMSs were used to test the algorithm. The first dataset comes from a traditional, but gamified, academic course, meaning it had no structured processes. A second case study was based on data from a standard massive open online course (MOOC). The emerging patterns from both cases studies, indicating how different users approached these courses, is presented. Lewis Carroll writes in Alice in wonderworld: "If you do not know where you are going, any road will get you there", therefore we were required to answer the question, how do we know if the results are accurate or random. To validate the results, we generated random user behavior patterns and inserted simulated data representing them into the dataset of the first cases study. The algorithm was executed again – confirming that previous patterns as well as the simulated patterns emerged.

This paper is structured as follows. First, a background on pattern discovery and process mining is provided. A brief background on gamification and the way it can un-structure processes is given, and the limitations of existing process mining methods are outlined. Next, the UBPD methodology is proposed, and relevant considerations are discussed. Two real-life case studies and simulation data are used to demonstrate how the methodology works and how results are achieved. Finally, a discussion of the results, applicability, and limitations of the methodology, as well as future research directions are provided.

Background

Pattern discovery

Understanding user behavior in online systems helps site developers and designers understand how their system is being used, what works well, and what needs to be improved (Srivastava, Cooley, Deshpande, & Tan, 2000). System log files can partially answer these questions as they provide statistics such as the most accessed page, the frequency of visits per user, and the duration of time on a page. Error log files complement this data by providing information such as broken links, unauthorized access attempts, general errors on the website, and more, depending on the richness of these logs.

Understanding the bigger picture hidden within the log files requires going beyond basic statistics. In systems where users are expected to follow a specific process (i.e. completing an online order or purchase request), analysts might want to know if users are indeed following this process, are there deviations from the process and which users are deviating from it. In systems where there is no process to follow (i.e. news web sites or knowledge management systems), analysts might be interested in questions such as what, if any, sub-processes exist, are all users behaving in the same unstructured manner or are there different classes of users that emerge. As information systems are often a mixture of structured and unstructured processes, in most cases, all the above questions are relevant.

Several advanced methods exist to address these more complex questions. Clustering methods (Ferreira, Zacarias, Malheiros, & Ferreira, 2007; Luengo & Sepúlveda, 2012) are used to group user actions with similar characteristics, classification methods (Pennacchiotti & Popescu, 2011) are used to classify user actions into a given set of classes, and association rules methods (Agrawal & Srikant, 1994; Lau, Ho, Chu, Ho, & Lee, 2009) are used to detect user actions that frequently appear together. Beyond user behaviors, it is sometimes interesting to detect hidden processes or parts of processes. Methods such as process mining (van der Aalst, 2011b; van der Aalst et al., 2012; van der Aalst & Günth, 2007) and sequence analysis (Van Helden, 2003) are used in such cases. Most

of these methods use system log files as input and assume a sequential set of activities are recorded in them, indicating there is a process that led to the execution of these sequences of actions, hence, the discovered process.

Sequence mining (Srikant & Agrawal, 1996) and episode mining (Mannila et al., 1997) examine sequences of events and search for recurring usage patterns based on the most frequent sequences of events. They do not necessarily require that an end-to-end process exists, and rather focus on subsets of processes. The Apriori and generalized sequential pattern (GSP) methods (Agrawal & Srikant, 1994; Srikant & Agrawal, 1996) are commonly used for this task by scanning the entire set of sequences and searching for sequences that meet a minimum frequency threshold but may be time consuming when datasets are large (Han et al., 2001). Episode mining (Leemans & van der Aalst, 2014; Mannila et al., 1997) uses the notion of a sliding window based on time or number of events and searches for frequent items within this window. Sequence hierarchy discovery is an algorithm that looks at hierarchies of sub-processes (Greco et al., 2005) and tries to combine them into a full process, assuming it exists. Some of the more recent algorithms use stochastic modeling and a Markov chains approach (Balakrishnan & Coetzee, 2013; Faucon, Kidzinski, & Dillenbourg, 2016; Geigle & Zhai, 2017) to address the fact that not all users interact with the system in the same way and describe how users navigate within the system.

Web server log files are good candidates for sequence mining (Mobasher, Cooley, & Srivastava, 2000; Patel & Parmar, 2014; Sisodia & Verma, 2012; Spiliopoulou, 2000; Srivastava et al., 2000) because pages are accessed sequentially, and there are several links a user can select at any given moment. Studies have shown that sequence mining provides good results and is already in use in generating personalized websites (Ferreira et al., 2007). Sequence mining is also commonly used in genome studies to examine DNA sequences (Kaneko et al., 1996).

The aforementioned methods work well for systems with an underlying business process such as in the case of purchasing (Ingvaldsen & Gulla, 2008), audit processes (Jans, van der Werf, Lybaert, & Vanhoof, 2011), supply chain management (Lau, Ho, Zhao, & Chung, 2009; Trkman, McCormack, De Oliveira, & Ladeira, 2010), and other business processes that have clear start and end points. However, not all systems have an underlying business process. News websites allow users to consume news differently, in Learning Management Systems (LMS) the processes may be extremely short, such as accessing a system to download a presentation, view a video, or submit an assignment, in MOOCs participants can interact with the learning materials in any order and time that they choose, and in social network sites, users can browse content and jump from topic to topic in what may seem like a chaotic behavior.

While process mining methods have shown great success in discovering structured processes, they are less successful with non-structured processes where processes do not have a clear path and any step can follow any step (Rebuge & Ferreira, 2012; van der Aalst, 2011b). Structured processes are processes in which all activities are repeatable and have a well-defined input and output, while unstructured processes are processes where activities have no pre- or post-activity and are determined based on experience, intuition, trail-and-error, and rules-of-thumb (van der Aalst, 2011a). Discovering specific usage patterns in non-streamlined and non-structured processes is a promising research direction (Celino & Dell'Aglio, 2015). Even in cases in which there is a significant underlying process, it may have so many deviations, that the ratio between the deviations and main process is too large, and the existing algorithms would struggle to fully understand what the intended process is and what are the deviations. In such cases, sequence mining methods are typically used to identify sub-processes that may or may not add up to a full process. When there is no clear process, the focus is switched from examining how a system is being used, to how different users are using it, also referred to as user behavior patterns. User behavior patterns are sequences of actions that are performed by a user sequentially (Tseng & Lin, 2006) or almost sequentially. There is no definition to the amount of actions that constitute a pattern, and in some cases, even two activities qualify as a pattern (Kang, Liu, & Qu, 2017).

For the detection of user behavior patterns to be useful, the process of detecting and analyzing behavior patterns must be fully automated, which is missing in current research. In some studies

(Davis, Chen, Hauff, & Houben, 2016; Hou, 2015; Huang, Chen, & Lin, 2019) the analysis process is indeed automated using sequence and clustering methods, but the data collected and the pattern detection processes are based on manual observations and interpretations, or on a set of predefined expected behaviors. The limitations of these methods are both in the manual classification step and in their need for a predefined set of behavior classes. Another issue with many of the existing processes is that they work at the system level and not at the user level. They seek to understand the overall process or sub-processes performed by users, ignoring the inherent differences between users. The above leads to the following research question: Within a non-structured process or system, is it possible to automatically identify recurring user-level behavior patterns, and perform user clustering based on these patterns?

The case of gamification – when a process becomes unstructured

Gamified systems are good examples of loosely-structured processes. Gamification is the use of game design elements in a non-gaming environment (Deterding et al., 2011) with the intent of increasing user engagement (Kankanhalli et al., 2012; Werbach, 2014), hedonic motivation (Lowry et al., 2013; Van der Heijden, 2004), or achieving other business goals (Hamari & Koivisto, 2015). In recent years, gamification is commonly included into LMS (Buckley & Doyle, 2016) as a means to increase motivation. The inclusion of game elements, into a utilitarian environment, such as LMS, is likely to change the way users interact with the system due to the additional options and affordances provided, reducing the structure of existing business processes. Due to the unstructured nature of gamified systems, using process or sequence mining to discover an underlying process would be challenging and can become even more challenging if the system was initially unstructured.

The most common approach to studying the effects of game elements on users is to examine the isolated effects of specific game elements and assess their contribution to the overall objectives of the gamification implementation. The most common game elements studies are points (Mekler, Brühlmann, Opwis, & Tuch, 2013), badges (Anderson, Huttenlocher, Kleinberg, & Leskovec, 2013; Antin & Churchill, 2011; Hakulinen, Auvinen, & Korhonen, 2013), leaderboards (Butler, 2013; Costa, Wehbe, Robb, & Nacke, 2013; Landers & Landers, 2015; Mekler et al., 2013), and levels. The majority of studies focus on effects of a single game element on gamification success (Hamari & Koivisto, 2013; Li, Grossman, & Fitzmaurice, 2012), providing insights at the game element level. In reality, gamified systems do not include just a single game element, and the ability to understand user behavior patterns provides the ability to study the interaction between game elements and their influence on gamification success, which is a line of research only a few scholars pursue (Codish & Ravid, 2014a, 2014b).

The goal in gamification is to trigger user behaviors that support business objectives. Designers may intentionally try to trigger a specific behavior through gamification (e.g. create a cooperative environment or a sharing culture), however, they might also add game elements without fully understanding of how users would relate to them. In any case, even with proper design, it is hard to predict precisely how users would interact with game elements. Due to the unexpected behaviors that may arise (Callan, Bauer, & Landers, 2015; Werbach, 2014), measuring the outcomes of gamification is an important activity that should be performed throughout the implementation phase.

One option for measuring success of gamified systems is to measure the desired business objectives before and after gamification implementation. While such an approach has its benefits, it lacks the ability to provide insight into how individual users are influenced. This latter point is important since not all users would be influenced in the same way, and while some users may be extremely engaged, others may be negatively affected. Understanding how users interact with a system, be it an expected behavior or not, requires systematic detection of these user behavior patterns, which, as mentioned, is not trivial. To date, few authors (Ašeriškis & Damaševičius, 2014; Codish & Ravid, 2015; Sisodia & Verma, 2012) have proposed going beyond the analysis of trivial user behavior patterns in gamified environments and seek emerging patterns through log analysis. However, these

studies do not provide an automated method to perform these tasks and focus on the theoretical conceptual steps that should be taken.

Systems and gamification implementations differ from each other, thus, any methodology for detecting user behavior patterns must be completely automated and system independent. We propose the User Behavior Pattern Detection (UBPD) methodology, which is based on sequence analysis methods, as an automated process for detecting differences in behavior patterns between users. We consider a user behavior pattern as a pattern that is common to several users but not to all users, which is the essential difference between a user behavior pattern and a system level usage pattern. To demonstrate and validate the methodology, we use a learning management system, which has no streamlined processes, and include gamification to make it even less structured.

Methodology

Terminology

Extracting user behavior patterns from a system requires examining sets of common usage patterns and looking for user-specific repeating patterns. Unlike methods such as episode mining (Mannila et al., 1997) and sequence analysis (Van Helden, 2003), where the objective is to find frequently recurring patterns, in this case the objective is to find patterns that are frequent for only some of the users. Having such patterns is an essential phase in the ability to cluster users based on their behavior patterns.

Using process mining terminology (van der Aalst et al., 2012), the following terms are defined as summarized in Table 1. An *event* is an archetype action that can be recorded by the system. Events are determined by the system's capability to generate them. Examples of an event are opening a file, visiting a page, or viewing a video. An *activity* is a single event performed by a user and recorded by the system. If a user performs an event many times, each occurrence of performing the event will be recorded as an activity. Not all events need to be analyzed, such as system-generated events, time-based events, or error messages. These can be considered irrelevant to user behavior analysis, and at a certain point during the cleanup phase, they should be removed. However, it is important to note that in some cases, these supposedly non-relevant events may trigger events by the user and should perhaps not be ignored.

Systems often record many types of events that practically represent the same action. For example, suppose there are different events called opening link A, opening link B, and opening link C. If these events represent opening a link with no need to distinguish between them, we should represent the three events as a single action called "open link". This means that an action is a superset of events that, for analysis purposes, represent similar events.

A *session* includes all activities performed by the user between the timeframe of logging into the system and logging out of the system. Thus, there is a need to identify these sessions. In cases where a user logs in and logs out, this is straightforward, but in many cases, such as when systems remember user authentication, the login is automated and is not recorded as an event. Logging out of a system depends on users' habits and awareness of privacy issues. In some cases, users close the system without logging out, and in cases in which a personal device is used, a logout may never happen. To overcome this limitation, it is common to use a threshold of 30 minutes of inactivity to indicate the start of a new session (Clark, Ting, Kimble, Wright, & Kudenko, 2006).

Table 1. Behavior patterns methodology terminology.

Term	Definition
Event	An archetype action that can be recorded by the system
Activity	A single event performed by a user and recorded by the system
Action	A superset of events, that for analysis purposes represent similar events
Motif	Recurring sequences of actions that appear in a network more frequently than expected in a random network
Session	All activities performed by the user between the timeframe of logging into the system and logging out of the system. If a user is not active for more than 30 minutes, a log out activity is automatically defined

Searching for user behavior patterns requires the identification of cases in which a specific sequence of actions re-occurs more frequently for some users than it does for others. Most process mining methods do not focus on user behavior differences, and thus seek frequently performed sequences of actions regardless of who performed them. The focus on user-specific behavior patterns is the key difference between UBPD and existing process and sequence mining methods.

Searching for frequent sub-sequences of actions within a given sequence is the focus of several algorithms, such as the Apriori (Agrawal & Srikant, 1994), the GSP algorithm (Srikant & Agrawal, 1996) that expands the Apriori algorithm, and episodes finding (Mannila et al., 1997), in which episodes are defined as "a collections of events that occur relatively close to each other in a given partial order" (Mannila et al., 1997, p. 259). These algorithms are good at finding overall frequent sequences of actions. They do not, however, directly address our need for detecting user-specific behavior patterns.

Borrowing a term from genetics research, where sequence mining is commonly used, a *motif* is defined as a "recurring pattern that appears in a network more frequently than expected in a random network" (Alon, 2007; Milo et al., 2002). Motif research originally focuses on detecting how proteins regulate genes, but it is used in different domains as well, gaming among them, where they are used to understand how specific actions regulate behavior (Ghoneim, Abbass, & Barlow, 2008). In terms of behavior patterns, motifs are the recurring sequences that appear in user sessions. Figure 1 shows how all the terms defined above relate to each other.

Algorithms dealing with finding frequent subsets of actions, i.e. motifs, differ in how they achieve this. In our case, we seek to find user specific usage patterns we can relate to a user behavior. The most predominant question that needs to be addressed is what qualifies as a frequent motif. Algorithms address this by defining threshold values determining that any value above the threshold is frequent, but how this threshold is calculated has not yet been determined.

User behavior pattern detection process

The following section outlines the UBPD methodology. A graphical overview of the methodology is presented in Figure 2. The methodology is broken into four main parts: Extract transform and load (ETL), sequence mining, clustering, and interpretation phases.

As with all process mining methodologies, the first stage of the methodology is an extract, transform, and load (ETL) process where the data to be analyzed are collected from the various data sources and combined, cleaned, and organized in a format to which an algorithm can be applied.

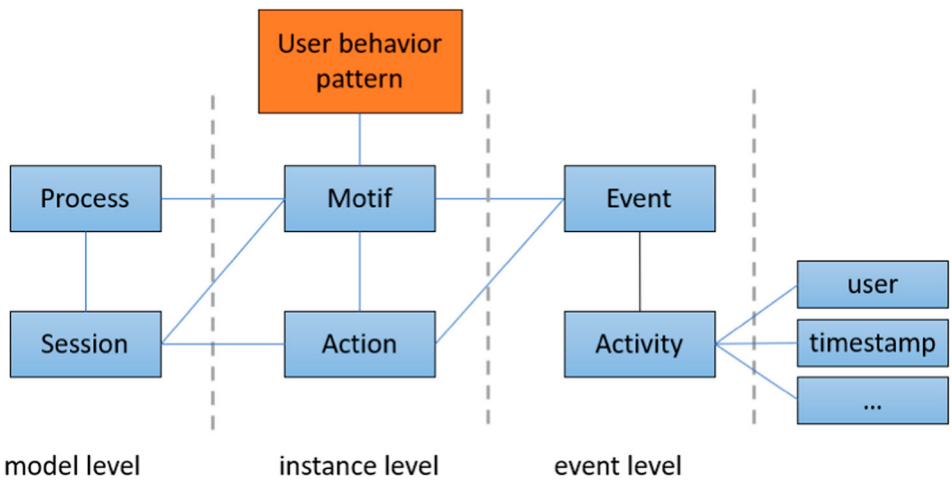


Figure 1. Visual representation and links between methodology terms.

The ETL stage is unique for each system because data is stored and organized differently in each system, but the results need to be in a single dataset that includes, at a minimum, the user id, activity, and time of event. Activities may or may not include additional information allowing for further data analysis, but our methodology does not require it. Each activity represents an event that a user performed, however, not all logged events need to be analyzed as they might represent time-based events, error messages, or administrative tasks, that are not relevant to the understanding of user behavior. As part of the ETL configuration, designers should consider which event to include in the analysis dataset. It should be noted that in cases where a user behavior may be triggered by an event, it should not be deleted.

Designers should determine which events should be clustered together using the same action, and the ETL phase should then rename the activities dataset to include at a minimum, the user id, action, and time of action. For data processing efficiency reasons, it is useful to enumerate each action with a unique identifier to allow for faster data analysis and simplified results presentation. If there is no need to cluster events into actions, this step is not necessary, but in many cases, different events do have similar meanings.

In the second phase, the actions dataset is broken into user sessions. Each user session is prefixed with a login action and postfixed with a logout action, if they did not already exist. The output of this stage is a list of sessions that include a user identification and a time-ordered sequence of user actions within each session (Figure 3[a]). Consecutive identical actions are ignored in this process since we seek to understand the transition behavior between actions. If a user spends a long time doing something, we consider this to be a single action. For instance, if a user is reading content

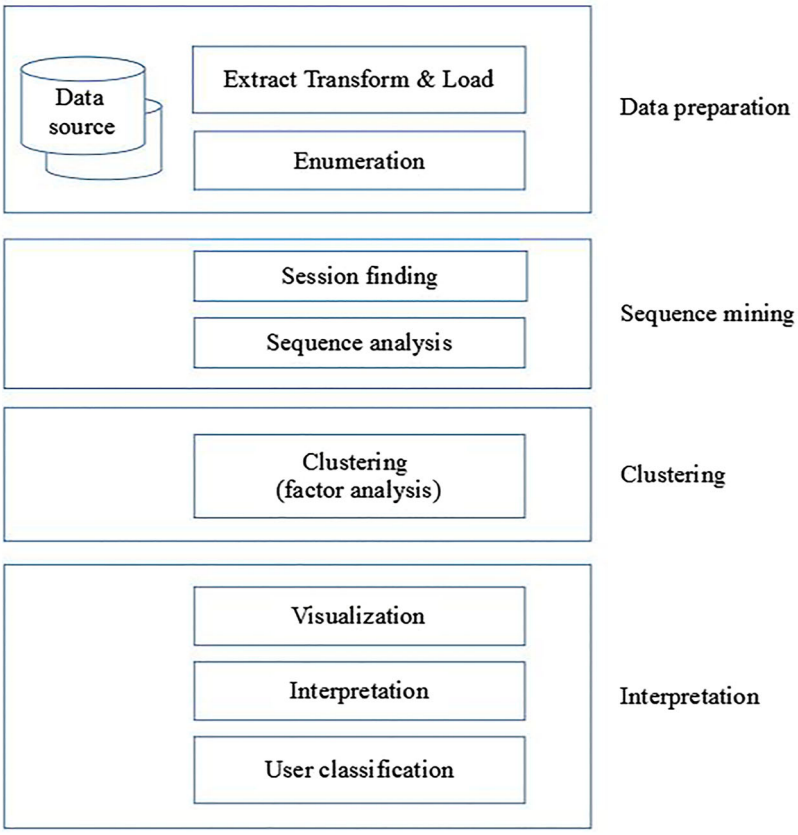


Figure 2. An overview of the UBPD methodology.

on a web page, and continues to read content, this is considered a single activity that does not transition from reading content to reading content.

A sliding window of size W is used to define sequences of actions with a length of W. The size of W can vary from as low as two actions and up to the size of the longest session. Smaller window sizes (e.g. shorter sequences) have an advantage because they can detect short behavior patterns that are masked when looking at wider window sizes. Due to the long tail effect, smaller window sizes also guarantee that the motifs selected are those who are more frequent. Wider window sizes are more likely to represent the true meaning of a sequence of actions, but they also reduce the number of sequences that are extracted from each session, up to the point where the window size is longer than the session length and nothing is extracted. Balancing between shorter window sizes and more meaningful sequences, it is recommended to set the upper limit of the window size to the first quartile of the session length, which means that up to 25% of the sessions are ignored. Allowing larger window sizes would result in loss of information to analyze which can harm the analysis. Analyzing the ratio between the number of unique motifs and total number of motifs, against the window size, would allow to determine the optimal window size which beyond it, increasing the window size would have a minor effect on the ratio. The output of this stage is a list of motifs of length W performed by each user. Figure 3(b) shows the output of this stage for a window size of three using the example in Figure 3(a).

A single motif represents a very short sequence of actions. In systems where users can easily navigate between different actions, we would like to understand which sequence of actions (i.e. motifs) lead to which sequence of actions most frequently. A set of motifs which are frequently performed together by some users more than others, represent a user behavior pattern. Detecting these groups of user behavior patterns is done through clustering groups of similar behaviors using an exploratory factor analysis (EFA) with the most frequent motifs as input. Each of the most frequent motifs is assigned to a dummy variables and a count of the number of occurrences of that motif for each user is done. The matrix of users and the number of occurrences for each motif (i.e. the dummy variable) by user is used as the input to the EFA. The output of the EFA is a set of constructs that represent user behavior patterns as they cluster motifs which load high on some users and low on others. The selection of EFA as the clustering method was done after using different clustering methods such as hierarchical clustering (Murtagh & Contreras, 2017), dendrograms, and K-means. All algorithms produced similar results but the EFA was the most efficient in terms of performance and the number of configuration parameters.

a.

User ID	Session Data
nnn1	A B C E G B D F Z
nnn1	A B E C B G Z
nnn2	A E C B D Z
nnn2	A D F B G E C B Z

b.

W=3	
User ID	Subsequence
nnn1	ABC
nnn1	BCE
nnn1	CEG
nnn1	EGB
nnn1	GBD
nnn1	BDF
nnn1	DFZ
nnn1	ABE
nnn1	BEC
nnn1	ECB
nnn1	CBG
nnn1	BGZ

Figure 3. Schematic output of the session identification stage: (a) session data and (b) motifs for a given user.

The exact number of the frequent motifs to include in the EFA is not straightforward, as researchers are not in agreement about the required ratio between variables and subjects. Ratios of 1:3 (Cattell, 2012), 1:5 (Bryant & Yarnold, 1995; Gorsuch, 1997), 1:10 (Everitt, 1975) and higher have been recommended as rules of thumb. Other scholars have noted that this ratio depends on the data characteristics and number of subjects, meaning that it is up to the researchers running the analysis to decide the correct ratio based on communalities, sample size, and the number of factors (MacCallum, Widaman, Zhang, & Hong, 1999). In cases where there is a clear-cut between frequent and non-frequent motifs, only the frequent ones should be used, however, in cases where frequency distinctions are not easy to make, system designers need to make a reasonable decision about the ratio by optimizing the number of frequent sequences included in the analysis, the number of factors generated by them, and the explained variance gained by adding more sequences to the analysis.

The details of running a factor analysis are beyond the scope of this paper – for a detailed analysis see Cattell (2012); however, the result of this process is a set of constructs that includes motifs that users perform together. The exact number of constructs to expect depends upon the complexity of the system analyzed. The standard cut-off criteria of eigenvalues smaller than one can be used, unless it is possible to clearly define the number of expected behavior patterns. Since each construct includes a set of motifs (e.g. sequences of activities), the best visual representation of a construct is a causal net. Causal nets are directed networks showing the flow of activities from node to node Figure 4 shows how drawing the relations between all motifs in a construct provides a view to the user behavior pattern.

Factor analysis provides a score for each subject on each construct. A high score on a specific construct means that the behavior represented by the construct is more salient for that user. The combination of scores given to each user on each construct represent the users' overall behavior classification. For instance, if a system has two constructs being interpreted as competitiveness and curiosity, and we can define a high-medium-low scale for each construct, nine different classes of users can be drawn from these two constructs.

The last phase in the process is interpreting the meaning of the construct. Factor analysis effectively detects when there are commonalities between the behaviors in a construct but cannot interpret their meaning, which is something that system designers and analysts should determine. System designers should also be the ones to determine the course of action to take as a result of these findings.

The methodology presented so far is based on a myriad of existing methods in process and sequence mining that are combined to interpret usage logs and detect specific recurring user behavior patterns. Executing this methodology requires the extraction of sequences of activities, which is typically a system-specific manual process, and a standard statistical software package to perform the factor analysis. While these methods are all grounded in theory, combining them to identify user

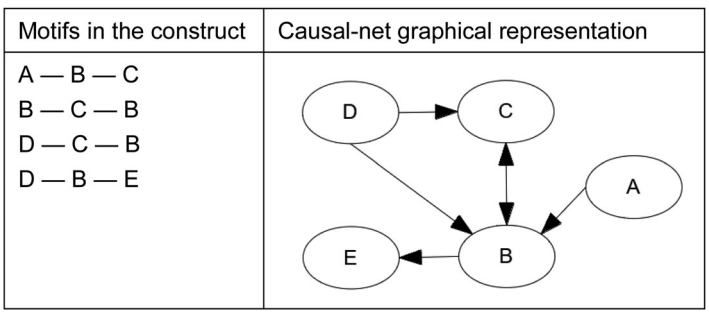


Figure 4. A sample representation of motifs of size three belonging to the same construct.

behavior patterns is a novel approach. In the next section, we demonstrate the use of this methodology using two different real-life examples.

Case studies and simulation

Both case studies presented in this paper are based on the Moodle LMS but represent different learning scenarios. The first case study is based on a standard academic course where various gamification elements were added causing the usage of the LMS to be more chaotic. The second case study is based on a MOOC with users mostly viewing videos and submitting assignments. The behaviors expected in both case studies are different. In the MOOC case study, we expect to discover users with different learning strategies, while in the gamified course we expect to find behaviors that are impacted by the gamification. Existing research already uses behavior patterns to

Figure 5 provides a visual representation, using a Petri-net structure, of the two case studies showing their actual data, along with a standard academic course with no modifications. This representation highlights the differences between courses and the inability of producing meaningful insights based on such a representation.

LMSs carry a major promise for adaptive learning and enriched learning experiences (Costa, Alvelos, & Teixeira, 2012); however, in many cases, student interactions with them are centered around downloading class material, handing in assignments, and reading announcements (Costa et al., 2012). Such tasks are atomic, or very short processes that are less interesting from a process mining lens because each task is only two or three steps long (see Figure 5-II).

Case study A – gamified academic course

This first case study is based on an existing learning environment which was gamified by adding different game elements. The data used for the analysis are from four consecutive semesters in which the course was offered in the same format. Students participating in the course were undergraduate students in their third year out of four with more than 95% of the students majoring in industrial engineering and management.

Course setting

The main objective of the gamified course was to increase student engagement with course materials by encouraging more frequent and meaningful interactions. The main functionalities of the standard LMS were kept, and game mechanics were added. First, a discussion board was added where

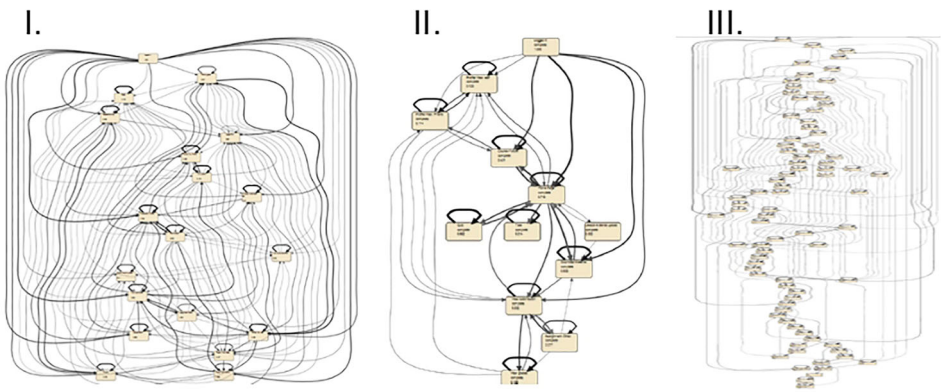


Figure 5. Network representation based on actual data of three types of courses. (I) Gamified course – Case study A, (II) Reference structure – Standard academic course, and (III) MOOC – Case study B.

students and staff could discuss items relevant to the course material. Discussion boards include good design principles for the incorporation of games in education (Aviv, Erlich, & Ravid, 2005; Li et al., 2012; Lieberoth, 2015) providing interaction opportunities between students and staff, allowing students to create content, build online identities, explore ideas, and take risks (Gee, 2005a, 2005b). For each contribution to the discussion board, students received a default value of 10 credit points, and for more meaningful contributions, participants received up to 50 points. Meaningless contributions, such as "I agree with the comment above", did not grant points. Each post was graded automatically and in real-time using software developed for this purpose. The number of points each participant had was visible to all students through a leaderboard. Contribution to the discussion board was partially mandatory, as students were required to reach 600 points over the semester. However, there were other mechanisms of earning points available to those who did not feel comfortable posting their thoughts online. The average number of points achieved by students ($n = 303$) was 792, with a standard deviation of 502, and a median of 700. The minimal amount of points was 300, and the maximum was 4418, indicating that some of the participants were extremely engaged while others were not. Many of the students continued discussions way after having reached the mandatory 600 points. Students were granted badges for completing certain activities in the discussion boards, such as contributing posts (1, 5, 10, 20, 50, or 100), responding to questions, and participating in various activities online.

Additional game mechanics aimed to increase engagement included voluntary weekly quizzes about the material taught that week. The weekly quiz scores were summed and presented in a dedicated leaderboard that ranked students. Logic riddles or small game-theory experiments in which students could voluntarily participate were made available at certain points throughout the course.

The use of points, badges, and leaderboard game mechanics is often criticized by gamification scholars, who claim that they are trivial implementations that harm long-term intrinsic motivation (Barata, Gama, Jorge, & Goncalves, 2013; Hanus & Fox, 2015; Mekler et al., 2013). While this may be true in some cases, for students whose intrinsic motivation is weak to begin with, these mechanics have been found to be successful for short-term tasks (Anderson et al., 2013; Butler, 2013; Hakulinen et al., 2013; Landers & Landers, 2015; Mekler et al., 2013) and were thus used in this study.

Data preparation

The log file used for analysis included 504,040 activities performed by 381 students participating in the course. The number of unique activities was 127 out of which 57 were deemed as system events such as emails sent and password reset requests or other redundant activities, leaving 70 activities in the analysis. These activities were mapped to 29 distinct actions – combining, where appropriate, similar activities into a single action.

A Perl program developed for this purpose takes the base dataset and processes it, separating the base dataset into sets of sessions. Using the sessions dataset, a separate dataset is created for different window sizes, which will later assist in the selection of the appropriate window size for the specific case. The window size selection is a key factor that must be determined at the beginning of the analysis. Analyzing the effect of increasing the window size on the average number of motifs per unique motif is shown in Figure 6. We would like to increase the window size up to the point where increasing it further, simply creates many unique motifs with very few instances in each. Based on the knee demonstrated in Figure 6 it is possible to determine that the right window size is three and that beyond that window size, the number of motifs per user does not change much.

Table 2 summarizes the impact of the window size on the number of motifs extracted and the number of unique motifs extracted. As window size grows, fewer motifs are extracted, and more of them are unique making them harder to analyze. A smaller window size means fewer actions are included, making the results less robust.

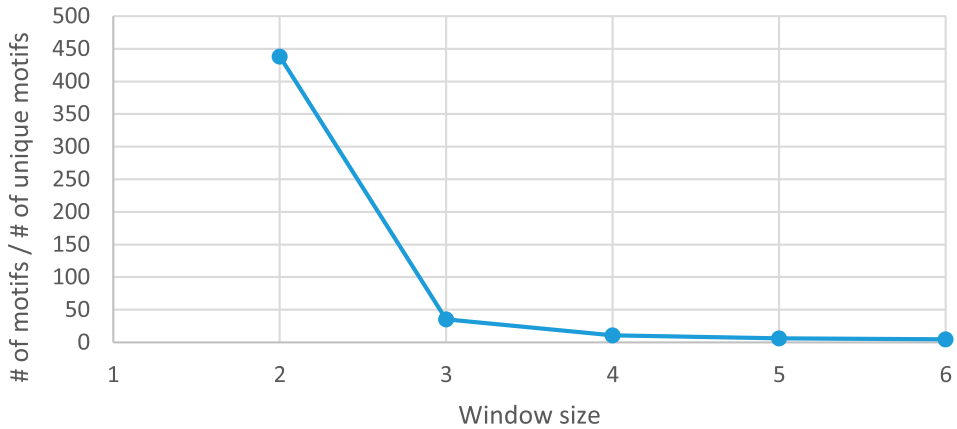


Figure 6. The ratio between the number of motifs and unique motifs – Case Study A.

Table 2. Window size calculations for case study A.

Window size	# of motifs	# of unique motifs	# of motifs / # of unique motifs
2	1,19,662	273	438.32
3	68,187	1931	35.31
4	56,534	5203	10.87
5	47,953	7581	6.33
6	41,683	8801	4.74

Pattern detection

Next, the motif dataset for a window size of three was processed by an R program developed for this purpose using the psych package and the embedded factanal procedure. The program summarizes the different motifs per user and performs an EFA based on the most frequent motifs using a varimax rotation. Since there is no prior assumption as to the number of factors to extract, the eigenvalue lower or equal to one criterion (Kaiser, 1960) was used. While additional methods exist for making this decision, such as parallel analysis (Horn, 1965), the method we use examines many different combinations of motifs and factors, allowing us to determine the optimal number for this problem. Eigenvalue was selected due to it being computationally simple and commonly used in research.

The results of this analysis are Petri nets representing user behavior patterns. Petri nets in this context, are used as a graphical tool similar to flowcharts, block diagrams, and networks (Murata, 1989) and are commonly used to represent processes (De Medeiros & Weijters, 2005). Defining what counts as most frequent is not straightforward. Ideally, the entire population of motifs would be included in the analysis, but since there are significantly more motifs than users, there is a limit on the ratio between motifs and users. A high ratio of 1:100 would result in fewer factors that do not explain variability, while a low ratio of 1:3 may result in an unreliable model since EFA is sensitive to such cases (MacCallum et al., 1999). The model was executed several times with different ratios, to assess the optimal ratio. As more motifs are included in the analysis, it is expected that the number of factors discovered will increase, and this is indeed what happened. However, more factors do not necessarily mean a better result, as factors may either be meaningless or repeat themselves with slight variations if the model is overfitted.

The frequency and variability of motif occurrences may also influence the ratio selection. As shown in Figure 7, there is a significant long tail effect, and the top 20 motifs account for nearly 65% of all motifs. However, the ratio between the frequency of appearance and variability is noisy, meaning that some of the less-frequent motifs create more variability, indicating that a higher number of motifs should be used to include more variability in the analysis.

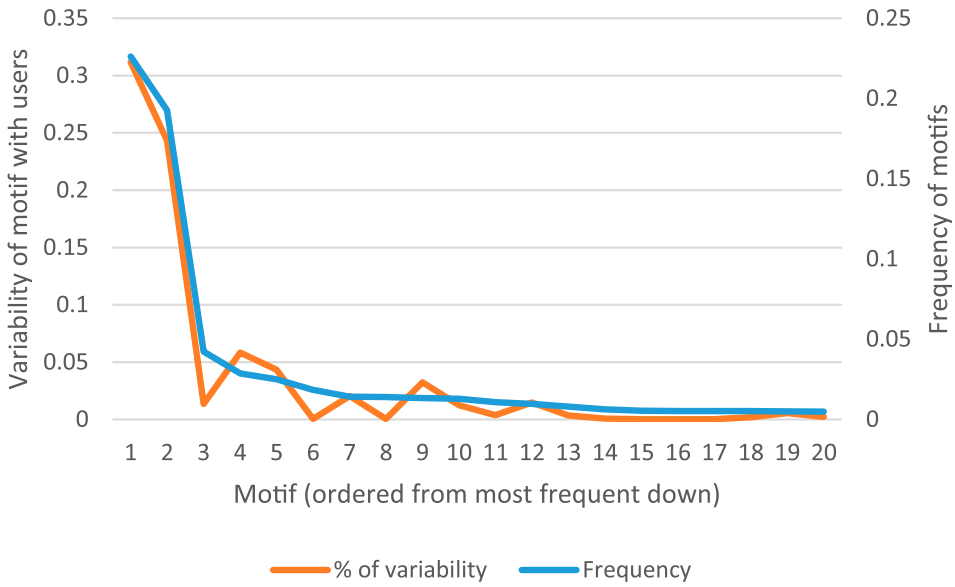


Figure 7. Variability and frequency of top 20 motifs.

Determining the right number of motifs to include in the analysis was done by running the analysis several times with different numbers of motifs and optimizing between the explained variance of the model and the number of motifs used. The results of this analysis are summarized in Figure 8. The x-axis shows the number of motifs introduced into the model. Left y-axis shows the number of factors discovered by the model, and the right y-axis shows the actual ratio used by the model after removing motifs that do not significantly load on any factor. The right y-axis also show the explained variance of the model. Ideally, a parsimonious model is preferred allowing for a minimal number of motifs and factors, explaining the maximum variance in the data. Taking this into account, a

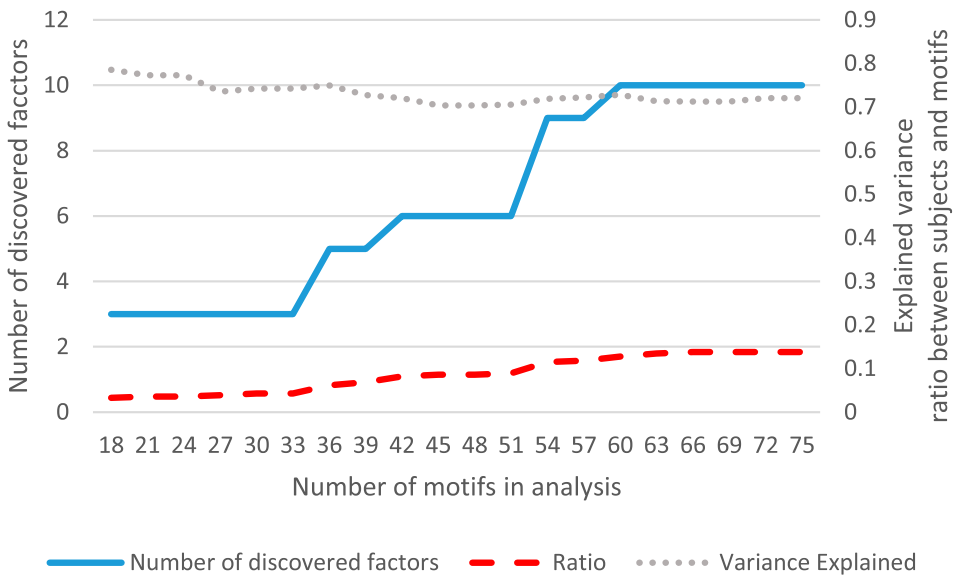


Figure 8. Summary of executing the model several times using different ratios – case study A.

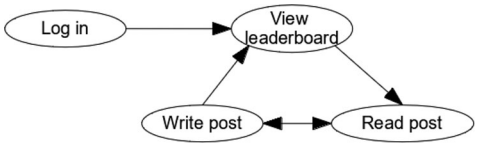
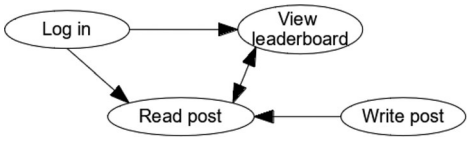
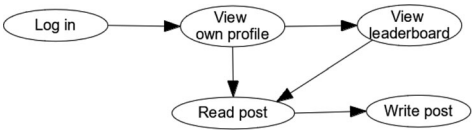
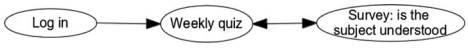
model using 36 motifs representing a 1:16 ratio was selected, explaining 75% of the variance, generating five distinct usage behavior patterns.

The model using 36 motifs was finally executed resulting in five factors. Patterns are presented as Petri nets, making them easier to understand visually. While EFA provides the understanding that a certain behavior is salient, the reason for the pattern being salient is a matter of interpretation. Table 3 shows the emerging patterns and a subjective interpretation based on our understanding of the environment in case study A.

While the results of case study A are plausible, we wanted to test the validity of the results by supplementing the actual data with simulated data of patterns that do not exist in the original dataset. If the methodology can detect these new patterns, our confidence in the correctness of the results is higher. In addition, if the results, can reproduce the same patterns as the data prior to simulation, our confidence in the validity of results is higher.

The data generated through the simulation process included the two patterns shown in Figure 9. The procedure for generating the data for pattern A was such that for each user, a random number of motifs representing actions that appear in the new patterns was generated, using a normal distribution. To include some variability, 30% of the motifs were set to be positive-false, i.e. represent a sequence of actions that involve the additional actions but do not match the pattern. Pattern B was simulated such that 40 motifs that match the patterns were randomly generated for every third user, ensuring significant variation between users. While adding variability to the patterns is necessary as the methodology is based on detecting variability, the value of 30% was arbitrarily chosen. As the

Table 3. Usage patterns – case study A.

Behavior pattern	Pattern	Possible interpretation
A1		Content contribution. The user logs in and is curious about his leaderboard position. He contributes and reads posts checking its influence on his position compared to others.
A2		Content reading. The main focus of this behavior is viewing content. It may include viewing the leaderboard or checking the status of the user's or other users status.
A3		Badge collection pattern. Badges were given for contributing data and were presented on the user's profile page. The key reason for a user to visit his profile page was to view their badges. In this behavior, the user logs in and looks existing or newly received badges, which leads him to explore additional status items such as the leaderboard, and to contribute more content.
A4		Knowledge points collection pattern. Two mechanisms were available for collecting knowledge points and in this pattern, users performed both sequentially. Knowledge points were the second type of points available for collection.
A5		Social networking pattern. Users reading content that other users posted would be curious about the users' postings and visit their profile pages to read about them and view their badges.

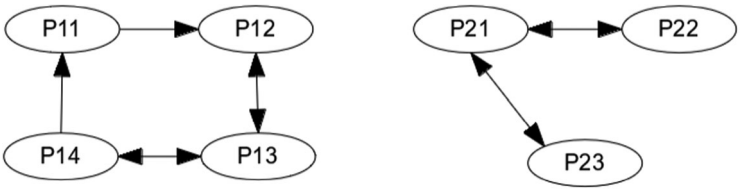


Figure 9. Simulated patterns.

variability increases, there would be no pattern to detect while on the other hand, with very low variability clustering method based on variability would not detect these patterns.

A window size of three was used for both the simulated model and the actual model, allowing better comparison between them. The simulated data included 80,943 motifs, out of which 2001 were unique motifs. These values are comparable with those found in Table 2 for the non-simulated data. A descriptive view of the data is shown in Figure 10 showing comparable results to Figure 7.

Next, the model was executed several times using different numbers of motifs as input to the EFA to determine the correct number of motifs to include in the analysis. The selection criteria were as before: fewer motifs, higher explained variability, and fewer factors. While Figure 11 indicates that a simple model of 18 motifs can be used, we selected a model with 36 motifs, which provides close results to that of 18 motifs but richer behavior patterns. As expected, the simulated model successfully identified the simulated patterns and behaviors A1, A2, and A4, as shown in Table 3. Increasing the number of motifs above 51 resulted in identifying behaviors A3 and A5 as well.

To summarize case study A, the UBPD algorithm detected five key behaviors performed by students in a gamified academic course using an LMS. The detected behaviors were related to the gamification of the course and how different students interacted with them. Unlike existing algorithms, there was no prior knowledge required about the existence of these behaviors, and their discovery and relating them to students was fully automated. The discovered pattern supports prior research indicating that different people are engaged differently by gamification (Codish & Ravid, 2014b; Hamari, Koivisto, & Sarsa, 2014).

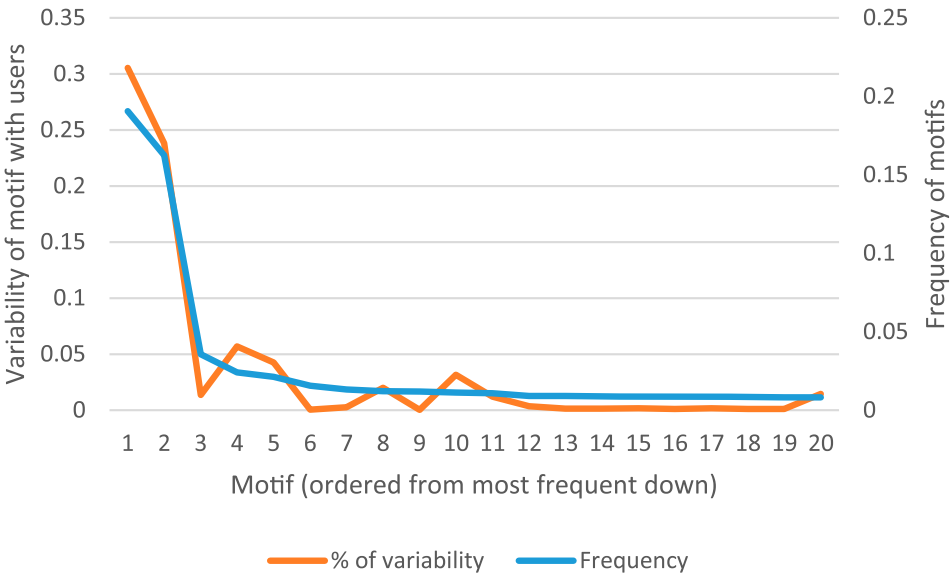


Figure 10. Frequency and variability of top 20 motifs – simulated data.

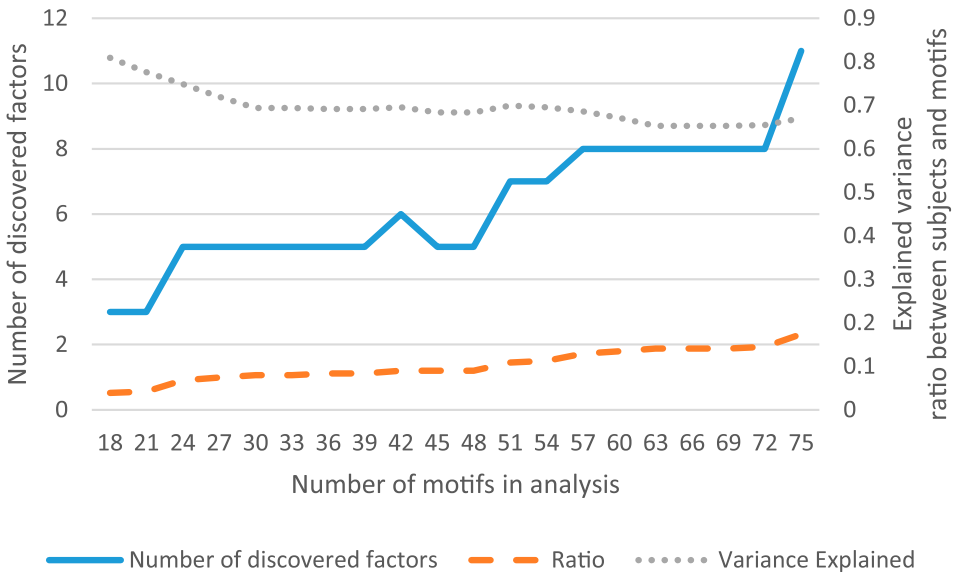


Figure 11. Summary of executions using different input variables – simulated data.

Including simulated data into the original data makes it possible to examine the validity of the algorithm. Original patterns were reproducible but required the inclusion of a larger number of motifs in the model, which is reasonable considering that instead of generating the original five behavior patterns, the simulation data were required to generate at least seven patterns. The simulated patterns appeared as they were expected to appear, despite the inclusion of positive-false motifs to the data indicating the algorithms ability to deal with noise.

Case study B – MOOC

In the second case study, data derived from system logs of a mid-sized MOOC on the recent history of the Middle East delivered in Hebrew were examined. The MOOC was offered by the Open University of Israel between 4 April 2015, and 7 July 2015. Students considered in this analysis were those who enrolled in the MOOC to get access to all the course materials and teachers (Kalz et al., 2015) and did at least one activity in the course. The course was freely available to the public without any prerequisites on knowledge or any other obligation and did not offer an academic recognition for completion of the course. During the course, participants' activities were recorded in a log-file.

MOOCs have specific characteristics that make them excellent candidates for learning analytics (Clow, 2013; Coffrin, Corrin, de Barba, & Kennedy, 2014; Kizilcec, Piech, & Schneider, 2013). They typically include many participants, have detailed log files, a good diversity of participants, and a process which is loosely defined. In most MOOCs, learners are expected to follow a standard process of watching video lectures in a specific order, answer quizzes and participate in online discussions. The key benefit of a MOOC is that it allows users to follow different paths that suit their learning styles, objectives from the course, time constraints, and other factors influencing their decisions. Therefore, while a main process does exist, learners will often deviate from it. Figure 5(c) shows a process map for a standard MOOC where it is clear there is an overall process, but various deviations are apparent.

Data preparation

The data file included data from 367 out of 1942 participants in the course, who agreed to have their data included in this analysis. Participants age ranged between 18 and 85 years ($M = 61$, $SD = 14.01$). Fifty-six percent were males. For most (63.7%), this MOOC was their first online learning experience,

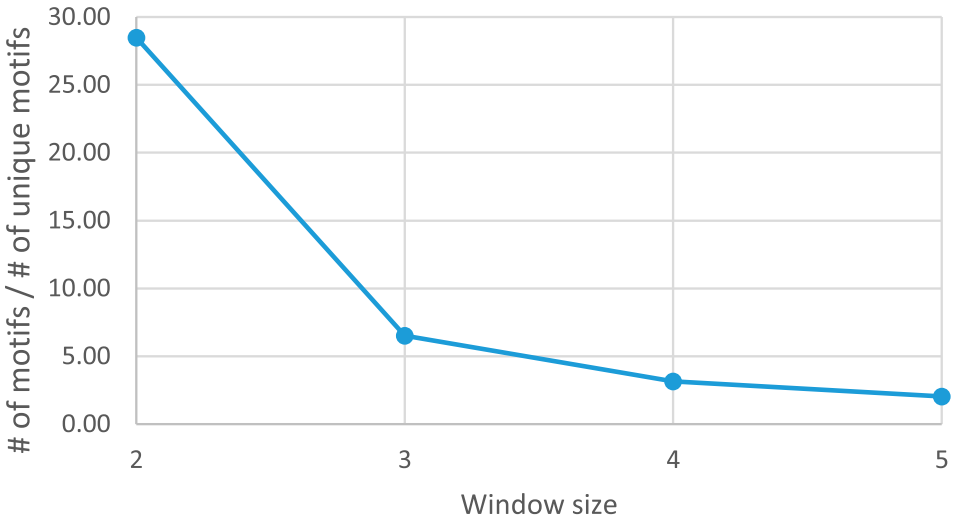


Figure 12. The ratio between the number of motifs and unique motifs – case study B.

and they indicated themselves as having high Internet skills ($M = 6.23$, $SD = .65$, in a scale range from 1 "Has very low Internet skills" to 7 "Has very high Internet skills").

The data file was clean of non-relevant data and included 93,942 log entries with 86 unique activities. As done in the first case study, an analysis to determine the best window size was executed. The results of this analysis appear in Figure 12 and show that as before, beyond a window size of three, the ratio between motifs and unique motifs becomes very low, which would result in low variability, making EFA less effective.

Pattern detection

Based on the window size analysis, motifs of window size three have been included in the pattern detection algorithm, and the model was executed 20 times with a different number of motifs each time to determine the best model. The results of this analysis can be viewed in Figure 13. Forty-two motifs were included in final analysis based on the observation that at this number, the explained variance was almost the highest while keeping a low ratio and fewer factors. Finally, patterns were extracted through the EFA process, and interpretations of the factors are shown in Table 4. The visualization of patterns through Petri nets are available in Appendix A.

To assess the impact of selecting more motifs into the analysis, the same model was executed with 57 motifs, which as shown in Figure 13, provide a similar level of explained variance while producing two additional behavior structures. For the analysis to be sound, it is expected that adding more motifs into the analysis will produce a similar set of behaviors, with richer data, which indeed happened. All behaviors detected with 42 motifs. The additional motifs detected appear in Table 4 as behaviors C8 and C9.

Case study B demonstrated the ability to extract the behavior patterns of students participating in a MOOC. A total of seven behaviors were extracted using a minimal set of motifs, and an additional two behaviors were extracted when using a larger number of motifs. While some of the behaviors were expected, such as in the case of B4 in Table 4, others were more surprising, such as in C8 where there are users who focus mostly on the first video lectures for every week.

Discussion and conclusion

Process mining is typically used to uncover underlying business processes and deviations from them by discovering actual user behavior and comparing it with the expected behavior (van der Aalst et al.,

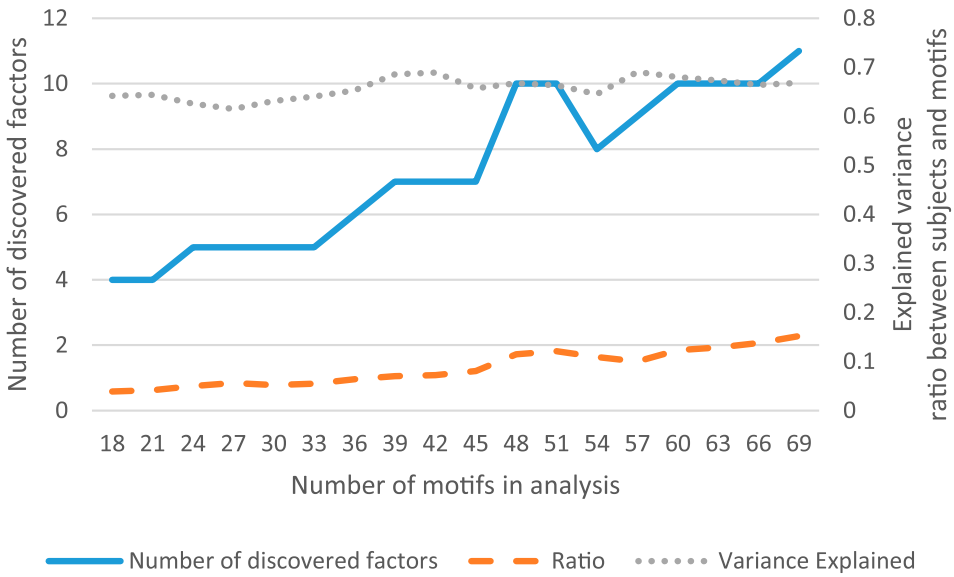


Figure 13. Summary of executions using different input variables – case study B.

Table 4. Usage patterns – case study B.

Behavior Pattern	Possible interpretation
B1	Users were motivated to complete all weekly quizzes. The weekly quiz is a self-evaluated activity that enables learners to evaluate their knowledge base on materials covered in the previous week.
B2	Sporadic first-week behavior. Users expressing this behavior viewed the first videos of the course one at a time and not sequentially.
B3	Users who mostly viewed the first videos of weeks 2–4 non-sequentially. This is a sporadic behavior that can be interpreted as an exploration behavior of merely checking on each week's topic, but not completing it.
B4	Users who viewed each week's lectures in sequential order. This is the expected behavior of a learner.
B5	Users who viewed lectures 1.4 and 1.5 not sequentially. Unlike B2 where users viewed lectures 1.1, 1.2, and 1.3 – Users who are strong on this behavior also viewed lectures 1.4 and 1.5 in a non-sequential manner. Users low on this behavior are those who did not continue to view the remaining lectures of the week.
B6	Users who accessed the site to view announcements in the general discussion forum. The general discussion forum was used as a social tool enabling learners to receive updates about the course progress and to introduce themselves to the learners` community.
B7	Users accessing the site to view week four forum. This behavior received no plausible explanation from the course staff.
C8	Users viewing the first lectures for each week. People with this behavior viewed the first and sometimes also the second lecture of each week non-sequentially. These might be people who are interested in the introduction to each topic without going into more detail.
C9	Users who viewed all of the first weeks' lectures sequentially. These would be people who were fully engaged only at the beginning.

2012; van der Aalst & Weijters, 2004). While successful at discovering well-structured processes, it is less successful in less structured processes where users have the freedom to execute the process in different ways. The challenge in the latter case is to detect these differences and understand if there is a reason for different users to behave differently. Our research question in this paper is: Within a non-structured process or system, can we automatically identify recurring user-level behavior patterns and perform user clustering based on these patterns? Specifically, as we focused on learning environments, these user behavior patterns can be viewed as learning processes.

This paper presents the user behavior pattern detection (UBPD) methodology along with two case studies based on LMS implementations, demonstrating its usage, and thus, answering this research question. Simulation data were included to present the effectiveness of the methodology in

discovering patterns that were injected into the data. In the first case study, a simple academic course was used, but after adding several game elements into it, it has become a complex, unstructured system. The second case study was based on a MOOC, where users have the freedom to decide what to do and how to do it. The differences between these two cases are evident when looking at Figure 3.

UBPD is unique in its focus on finding user behavior patterns that exist for only some users. It uses EFA to detect groups of activities performed together that explain the variability in the system. However, in processes with no variability in which all users perform a process in the same way, UBPD would not be of use. In systems where some of the processes are structured, and some are not, UBPD would detect the unstructured processes, ignoring the structured processes. In such cases, UBPD does not replace existing methods but rather complements them. The user clustering, which has been described above, is another key benefit of the methodology, as it provides insight into different user behavior patterns.

Several parameters and decisions were included in the methodology and are discussed in the order they appear within the methodology. The selection of actions to include in the analysis has a direct influence on the resulting patterns. Grouping activities into actions is often a straightforward task since it should be clear which activities should be grouped; however, it is important to ensure that the grouped activities represent a clear action. For instance, in the educational setting used in this study, all activities related to the submission of an assignment were grouped into an assignment submission action since they all have the same meaning. In both cases studies, activities such as resetting a password or downloading a presentation were not included in the analysis, however, this does not always have to be the case. Resetting a password is an administrative task, and thus not included, but if it is included, and UBPD detects it as a user behavior pattern (i.e. enough variability exists between users with regards to that activity), perhaps it indicates that some users are more forgetful than others. If actions only have a few occurrences, they will be removed later as part of the EFA process since they would not be considered frequent motifs.

In both case studies and the simulation data, a window size of three was selected. Figure 6 and Figure 12 show that beyond this size, the number of unique motifs grows significantly, resulting in many motifs with only a few occurrences per user. This window size might differ in other systems, and it is recommended validate this number for different situations and dataset sizes. In case study A, the dataset was larger and included fewer users and fewer actions. This resulted in a stronger tail effect than in case study B, which had a smaller dataset, significantly more users, and more actions analyzed. An additional reason for keeping a smaller window size is that using a large window size carries the risk of missing short usage patterns of two or three actions.

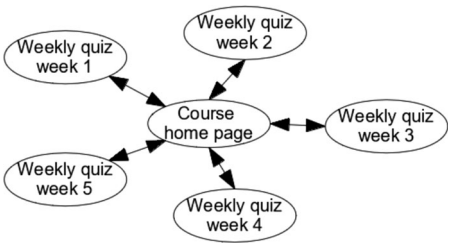
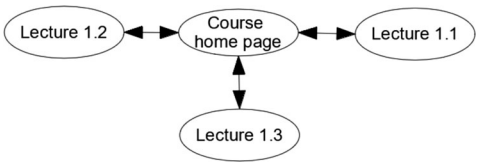
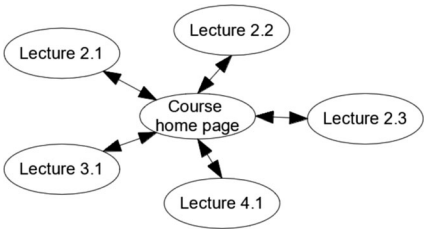
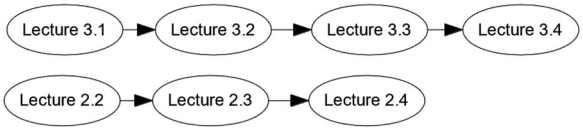
It can be assumed there is no known number of factors to expect during the EFA stage. Typically, EFA tries to maximize the explained variance, which in both case studies resulted in a minimal number of motifs to include as variables, and as a result, extracted factors. Including too many motifs into the analysis can result in overfitting and extracting meaningless patterns. Also, in cases in which there are few subjects, as in case study A, there is a limit on the ratio between subjects and motifs that must be kept (MacCallum et al., 1999). Since we are interested in extracting rich behavior patterns, we executed the model several times with a different number of motifs and selected a point that balanced these limitations. In case studies A and B, we demonstrated how adding motifs to the analysis does not change the discovered factors and can only result in additional factors. While this step was executed manually, it is possible to automate this step to determine the right number of motifs to include.

The validity of the resulting factors has been tested in several ways. First, simulation data have shown that when known patterns were injected into the existing dataset, the methodology was able to detect them correctly without impacting the existing patterns. This ability provides the confidence that the detected patterns are correct. Additionally, while increasing the number of motifs in the analysis increased the number of factors, only new patterns were added without impacting existing patterns, emphasizing the stability of the discovered patterns. The objective of process mining is

to discover an underlying process, but the meaning or reasons for a discovered process are left in the hands of system analysts to explain. In both of our case studies, the resulting patterns were presented to analysts and their interpretation of the results is included in Tables 3 and 4. Case study B, however, includes a pattern that was repeated in the two executions of the UBPD that had no plausible explanation by designers. It is possible that such a pattern indeed exists, but designers are unaware of it. It is also possible that it is a factor that should have been removed since it is based on a single motif (Streiner, 1994). Even if we ignore the unexplained patterns, UBPD was capable of automatically detecting user behavior patterns within unstructured processes, which is a task with which existing methodologies struggle (Rebuge & Ferreira, 2012; van der Aalst, 2011b).

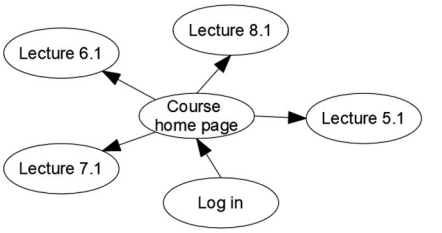
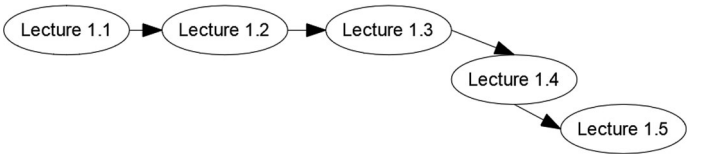
This paper presents three key contributions to the world of process mining, as well as several contributions to the development and analysis of interactive learning environments. From a process mining perspective, it provides the ability to discover different usage patterns of different users. While existing methodologies focus on the detection of the processes or sub-processes of a system, UBPD seeks to find the variance in how users interact with the system. To demonstrate

Table 5. Usage patterns – case study B.

Behavior Pattern	Pattern
B1	
B2	
B3	
B4	
B5	

(*Continued*)

Table 5. Continued.

Behavior Pattern	Pattern
B6	
B7	
C8	
C9	

this point, assume that all learner in a LMS perform a specific task similarly, such as reading an essay and immediately answering some questions about it. Methodologies such as episode finding, Apriori, or GSP would easily detect this pattern; however, UBPD would not since it would be performed by all users similarly. On the other hand, if different users performed that process differently (e.g. some read and answer questions immediately while others read part of the essay, answer a question, leave, and then come back to complete the task), the algorithms above might not detect any process at all, whereas UBPD would detect the process and the different ways people performed it. UBPD will even provide insight into which users are doing what. This was evident in the simulation we performed in case study A, where UBPD did not detect a pattern that was included to all users. However, when adding a pattern to only a few users, it was immediately detected.

The second contribution is the ability to deal with noise even within a sub-process. Existing methodologies seek stable processes or, such as in the case of association rules, stable relations between activities. UBPD detect similar motifs and through EFA, groups them into meaningful patterns represented as Petri-nets in Table 3. Finally, the methodology produces factor scores from the EFA to each user for each pattern, indicating how salient this behavior is for each user. A user can receive a high score on several behavior patterns, indicating those are the behaviors they perform most, or a low score on all behaviors meaning the discovered patterns do not represent their behavior. Using these scores to produce on-the-fly user clustering, is a unique capability that UBPD introduces and can be further explored.

From an educational point of view, UBPD detects how different learners interact in a learning environment. When designing a learning environment, educators often have a specific course of action that learners would follow, such as view all lectures sequentially, yet many do not follow that path. Being able to understand learner preferences, can help designers ensure that their design addresses these different preferences. In this study, we examined learner behaviors across a full semester, but it is possible to use shorter time frames such as a week or a month, and understand how learning preferences evolve.

Being able to provide close to real-time feedback on individual learning processes and comparing these processes with other learners and learning objectives carries a great potential for future developments in the field of personalized learning and adaptive learning. Detecting the learning processes currently being used and giving each learner a score on them can be used in many ways. Learners can see their learning process compared to other, which can be further used to modify or enhance certain behaviors. Teachers can use this data to assist specific learners and adapt their teaching styles, system designers can use this data to redesign or improve learning environments, and last, adaptive systems can automatically modify themselves based on actual usage data to encourage required changes in learning behaviors.

Limitations and next steps

Although simulation has been used to demonstrate the ability of UBPD to detect processes successfully, additional simulations should be done to determine the sensitivity of the methodology to variability. If there is no variability, processes would not be detected, and if the process is too variable processes would not be discovered since EFA would remove the actions from the analysis. This additional analysis was not included in this study to keep the focus on the paper on the methodology and should be further examined.

The process of selecting activities for analysis and combining activities into actions requires additional analysis. In the proposed methodology, this is part of a manual ETL process, but ideally, it can be automated using clustering methods. Additional manual steps, such as determining the correct number of motifs to include in the EFA, should be automated.

The clustering method used in this study was EFA which loads most of the variance on the first cluster. While different clustering methods have been examined throughout the study, a more in-depth comparison of different methods should be done, acknowledging that for different domains, different clustering methods might be more suitable. In addition, once a clustering is validated, different machine learning methods can be applied to further improve the clustering.

The two case studies came from a similar domain of LMS. Data from other types of systems should be analyzed to ensure the external validity of the methodology. Finally, in LMS and MOOCs specifically, user behavior changes over time. In future studies, a temporal model should be included checking user behaviors over time and providing meaningful data to system analysts as to what is happening right now in the system, not merely an overall of how the system is being used. The stability of behaviors can be tested as well over time since some behaviors might be salient at the beginning of a course and not at the end.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by Paul Ivanier Center for Production Management.

Notes on contributors

David Codish Ben-Gurion University of the Negev, Beer-Sheba, Israel (codishd@post.bgu.ac.il). Dr. Codish competed his Ph.D. and M.Sc. in the Industrial Engineering and Management department at Ben-Gurion University, Israel. For the past 20 years he managed several Information Systems organizations for a variety of hi-tech organizations and is now focusing most of his time on research. His key research area is gamification and its inclusion in various information systems as a means of making tedious tasks more fun, increasing user acceptance, improving the learning processes and achieving higher performance.

Eyal Rabin The Open University of the Netherlands (eyal.rabin@gmail.com). Mr. Eyal Rabin is a PhD student at the faculty of Management, Science and Technology at the Open University of the Netherlands. His M.A. in Social Psychology from the Hebrew University, Israel and his B.A. in Psychology from Ben-Gurion University of the Negev, Israel. Eyal is working as a statistical counselor and tutorial in the Education and Psychology department at the Open University of Israel. His research focuses on the relations between learners` characteristics, learning processes and study outcomes at massive, online, open courses (MOOCs) and other forms of online learning.

Gilad Ravid Ben-Gurion University of the Negev, Beer-Sheba, Israel (rgilad@bgu.ac.il). Prof. Ravid is a senior faculty member in Information Systems at the Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Israel. His Ph.D., titled "Information Sharing With CMC in Small Groups: Communication Groups and Tasks" is from Haifa University, his MBA from the Hebrew University, and his B. Sc. in Agricultural Engineering from the Technion. He was a postdoctoral fellow at the Annenberg Center for Communication, University of Southern California, Los Angeles. Dr. Ravid's main interests are focused on the relationship between social structure and human behavior, games and gamification and computer-mediated communication systems. His work includes such research as information overload phenomena, social structure in web-based educational forums, wiki-based education and wikipedia as a social space, celebrity formation, the "social structure" of lightning synchronization, civil information needs and information sharing in groups, learning with games and gamification solutions. He has published in top peer-reviewed journals including *Information Systems Research*, *First Monday*, and *Information Systems Journal*.

ORCID

David Codish http://orcid.org/0000-0001-8510-9256

References

Agrawal, R., & Srikant, R. (1994). *Fast algorithms for mining association rules*. Proc. 20th int. conf. very large data bases, VLDB.

Alon, U. (2007). Network motifs: Theory and experimental approaches. *Nature Reviews Genetics*, *8*(6), 450–461.

Anderson, A., Huttenlocher, D., Kleinberg, J., & Leskovec, J. (2013). *Steering user behavior with badges*. 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil.

Antin, J., & Churchill, E. F. (2011). *Badges in social media: A social psychological perspective*. CHI 2011 Gamification Workshop, Vancouver, BC, Canada.

Ašeriškis, D., & Damaševičius, R. (2014). Gamification patterns for gamification applications. *Procedia Computer Science*, *39*, 83–90.

Aviv, R., Erlich, Z., & Ravid, G. (2005). Response neighborhoods in online learning networks: A quantitative analysis. *Journal of Educational Technology & Society*, *8*(4), 90–99.

Balakrishnan, G., & Coetzee, D. (2013). Predicting student retention in massive open online courses using hidden Markov models. *Electrical Engineering and Computer Sciences University of California at Berkeley*.

Barata, G., Gama, S., Jorge, J., & Goncalves, D. (2013). *Engaging engineering students with gamification*. 5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES),, Bournemouth, UK.

Bryant, F. B., & Yarnold, P. R. (1995). *Principal-components analysis and exploratory and confirmatory factor analysis*. Washington, DC, US: American Psychological Association.

Buckley, P., & Doyle, E. (2016). Gamification and student motivation. *Interactive Learning Environments*, *24*(6), 1162–1175.

Butler, C. (2013). *The effect of leaderboard ranking on players' perception of gaming fun*. 5th International Online Communities and Social Computing Conference, Las Vegas, NV, US.

Callan, R. C., Bauer, K. N., & Landers, R. N. (2015). How to avoid the dark side of gamification: Ten business scenarios and their unintended consequences. In *Gamification in education and business* (pp. 553–568). Cham, Switzerland: Springer.

Cattell, R. (2012). *The scientific use of factor analysis in behavioral and life sciences*. New York, NY: Springer Science & Business Media.

Celino, I., & Dell'Aglio, D. (2015). Capturing the semantics of simulation learning with linked data. In *Gamification: Concepts, methodologies, tools, and applications: Concepts, methodologies, tools, and applications* (pp. 273). Hershey, PA, USA: IGI Global.

Chinces, D., & Salomie, I. (2015). *Optimizing Spaghetti process models*. 2015 20th International Conference on Control Systems and Computer Science.

Clark, L., Ting, I.-H., Kimble, C., Wright, P. C., & Kudenko, D. (2006). Combining ethnographic and clickstream data to identify user web browsing strategies. *Information Research: An International Electronic Journal*, *11*(2), 14.

Clow, D. (2013). *MOOCs and the funnel of participation.* Proceedings of the Third International Conference on Learning Analytics and Knowledge.

Codish, D., & Ravid, G. (2014a). Academic course gamification: The art of perceived playfulness. *Interdisciplinary Journal of e-Skills and Lifelong Learning*, *10*, 131–151.

Codish, D., & Ravid, G. (2014b). *Personality based gamification: How different personalities perceive gamification*. Paper presented at the European Conference of Information Systems (ECIS) 2014, Tel-Aviv.

Codish, D., & Ravid, G. (2015). Detecting playfulness in educational gamification through behavior patterns. *IBM Journal of Research and Development*, *59*(6), 6:1–6:14.

Coffrin, C., Corrin, L., de Barba, P., & Kennedy, G. (2014). *Visualizing patterns of student engagement and performance in MOOCs*. Proceedings of the fourth international conference on learning analytics and knowledge.

Costa, C., Alvelos, H., & Teixeira, L. (2012). The use of Moodle e-learning platform: A study in a Portuguese University. *Procedia Technology*, *5*, 334–343.

Costa, J. P., Wehbe, R. R., Robb, J., & Nacke, L. E. (2013). Time's up: Studying leaderboards for engaging punctual behaviour. Gamification 2013 Conference, Stratford, ON, Canada.

Davis, D., Chen, G., Hauff, C., & Houben, G. (2016). *Gauging MOOC learners' adherence to the designed learning path*. In Proceedings of the 9th International Conference on Educational Data Mining (EDM). Raleigh, NC, USA.

De Medeiros, A. A., & Weijters, A. (2005). *Genetic process mining*. Applications and Theory of Petri Nets 2005, volume 3536 of Lecture Notes in Computer Science.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). *From game design elements to gamefulness: Defining gamification*. 15th International Academic MindTrek Conference: Envisioning Future Media Environments, Tampere, Finland.

Everitt, B. (1975). Multivariate analysis: The need for data, and other problems. *The British Journal of Psychiatry*, *126*(3), 237–240.

Faucon, L., Kidzinski, L., & Dillenbourg, P. (2016). *Semi-Markov model for simulating MOOC students*. Proceedings of the 9th International Conference on Educational Data Mining.

Ferreira, D., Zacarias, M., Malheiros, M., & Ferreira, P. (2007). *Approaching process mining with sequence clustering: Experiments and findings*. 5th International Conference, Business Process Management, Brisbane, Australia, September 24-28.

Gee, J. P. (2005a). *Good video games and good learning*. New York, NY, USA: Phi Kappa Phi Forum.

Gee, J. P. (2005b). Learning by design: Good video games as learning machines. *E-Learning and Digital Media*, *2*(1), 5–16.

Geigle, C., & Zhai, C. (2017). *Modeling MOOC student behavior with two-layer hidden Markov models*. Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale.

Ghoneim, A., Abbass, H., & Barlow, M. (2008). Characterizing game dynamics in two-player strategy games using network motifs. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *38*(3), 682–690.

Golder, S. A., & Macy, M. W. (2014). Digital footprints: Opportunities and challenges for online social research. *Annual Review of Sociology*, *40*, 129–152.

Gorsuch, R. L. (1997). Exploratory factor analysis: Its role in item analysis. *Journal of Personality Assessment*, *68*(3), 532–560.

Greco, G., Guzzo, A., & Pontieri, L. (2005). *Mining hierarchies of models: From abstract views to concrete specifications*. International conference on Business Process management.

Hakulinen, L., Auvinen, T., & Korhonen, A. (2013). *Empirical study on the effect of achievement badges in TRAKLA2 online learning environment conference*. Learning and Teaching in Computing and Engineering (LaTiCE), Macau, China.

Hamari, J., & Koivisto, J. (2013). *Social motivations to use gamification: An empirical study of gamifying exercise*. 21st European Conference on Information Systems, June 5-8, 2013, Utrecht, The Netherlands.

Hamari, J., & Koivisto, J. (2015). Why do people use gamification services? *International Journal of Information Management*, *35*(4), 419–431.

Hamari, J., Koivisto, J., & Sarsa, H. (2014). *Does gamification work? -- A literature review of empirical studies on gamification*. 47th Hawaii International Conference on System Sciences, Hawaii, USA.

Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., & Hsu, M. (2001). *Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth*. Proceedings of the 17th international conference on data engineering.

Hanus, M. D., & Fox, J. (2015). Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Computers & Education*, *80*, 152–161.

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*, *30*(2), 179–185.

Hou, H.-T. (2015). Integrating cluster and sequential analysis to explore learners' flow and behavioral patterns in a simulation game with situated-learning context for science courses: A video-based process exploration. *Computers in Human Behavior*, *48*, 424–435.

Huang, T.-C., Chen, M.-Y., & Lin, C.-Y. (2019). Exploring the behavioral patterns transformation of learners in different 3D modeling teaching strategies. *Computers in Human Behavior*, *92*, 670–678.

Ingvaldsen, J. E., & Gulla, J. A. (2008). *Preprocessing support for large scale process mining of SAP transactions*. Business Process Management Workshops.

Jans, M., van der Werf, J. M., Lybaert, N., & Vanhoof, K. (2011). A business process mining application for internal transaction fraud mitigation. *Expert Systems with Applications*, *38*(10), 13351–13359.

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, *20*(1), 141–151.

Kalz, M., Kreijns, K., Walhout, J., Castaño-Munoz, J., Espasa, A., & Tovar, E. (2015). Setting-up a European cross-provider data collection on open online courses. *The International Review of Research in Open and Distributed Learning*, *16*(6), 62–77.

Kaneko, T., Sato, S., Kotani, H., Tanaka, A., Asamizu, E., Nakamura, Y., … Tabata, S. (1996). Sequence analysis of the genome of the unicellular cyanobacterium Synechocystis sp. strain PCC6803. II. Sequence determination of the entire genome and assignment of potential protein-coding regions. *DNA Research*, *3*(3), 109–136.

Kang, J., Liu, M., & Qu, W. (2017). Using gameplay data to examine learning behavior patterns in a serious game. *Computers in Human Behavior*, *72*(7), 14. doi:10.1016/j.chb.2016.09.062. Retrieved from http://www.sciencedirect.com/science/article/pii/S0747563216306975

Kankanhalli, A., Taher, M., Cavusoglu, H., & Kim, S. H. (2012). *Gamification: A new paradigm for online user engagement*. 33rd International Conference on Information Systems, Orlando, US.

Kizilcec, R. F., Piech, C., & Schneider, E. (2013). *Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses.* Proceedings of the third international conference on learning analytics and knowledge.

Lambiotte, R., & Kosinski, M. (2014). Tracking the digital footprints of personality. *Proceedings of the IEEE*, *102*(12), 1934–1939.

Landers, R. N., & Landers, A. K. (2015). An empirical test of the theory of gamified learning the effect of leaderboards on time-on-task and academic performance. *Simulation & Gaming*, *45*(6), 17.

Lau, H. C., Ho, G. T., Chu, K., Ho, W., & Lee, C. K. (2009). Development of an intelligent quality management system using fuzzy association rules. *Expert Systems with Applications*, *36*(2), 1801–1815.

Lau, H. C., Ho, G. T., Zhao, Y., & Chung, N. (2009). Development of a process mining system for supporting knowledge discovery in a supply chain network. *International Journal of Production Economics*, *122*(1), 176–187.

Leemans, M., & van der Aalst, W. M. (2014). *Discovery of frequent episodes in event logs*. International Symposium on Data-Driven Process Discovery and Analysis.

Li, J., Bose, R. J. C., & van der Aalst, W. M. (2010). *Mining context-dependent and interactive business process maps using execution patterns.* International Conference on Business Process Management.

Li, W., Grossman, T., & Fitzmaurice, G. (2012). *Gamicad: A gamified tutorial system for first time autocad users*. 25th annual ACM symposium on User interface software and technology, Cambridge, Massachesetts.

Lieberoth, A. (2015). Shallow gamification testing psychological effects of framing an activity as a game. *Games and Culture*, *10*(3), 229–248.

Lowry, P. B., Gaskin, J. E., Twyman, N. W., Hammer, B., & Roberts, T. L. (2013). Taking "fun and games" seriously: Proposing the Hedonic-Motivation System Adoption Model (HMSAM). *Journal of the Association for Information Systems*, *14*(11), 617–671.

Luengo, D., & Sepúlveda, M. (2012). *Applying clustering in process mining to find different versions of a business process that changes over time*. Business Process Management Workshops.

MacCallum, R. C., Widaman, K. F., Zhang, S., & Hong, S. (1999). Sample size in factor analysis. *Psychological Methods*, *4*(1), 84–99.

Mannila, H., Toivonen, H., & Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, *1*(3), 259–289.

Mekler, E. D., Brühlmann, F., Opwis, K., & Tuch, A. N. (2013). *Do points, levels and leaderboards harm intrinsic motivation?: An empirical analysis of common gamification elements*. Gamification 2013, Stratford, Ontario, Canada.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., & Alon, U. (2002). Network motifs: Simple building blocks of complex networks. *Science*, *298*(5594), 824–827.

Mobasher, B., Cooley, R., & Srivastava, J. (2000). Automatic personalization based on web usage mining. *Communications of the ACM*, *43*(8), 142–151.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, *77*(4), 541–580.

Murtagh, F., & Contreras, P. (2017). Algorithms for hierarchical clustering: an overview, II. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(6), e1219.

Patel, P., & Parmar, M. (2014). Improve heuristics for user session identification through web server log in web usage mining. *International Journal of Computer Science and Information Technologies*, *5*(3), 3562–3565.

Pennacchiotti, M., & Popescu, A.-M. (2011). A machine learning approach to twitter user classification. *ICWSM*, *11*(1), 281–288.

Rebuge, Á, & Ferreira, D. R. (2012). Business process analysis in healthcare environments: A methodology based on process mining. *Information Systems*, *37*(2), 99–116.

Sisodia, D. S., & Verma, S. (2012). *Web usage pattern analysis through web logs: A review*. Computer Science and Software Engineering (JCSSE), 2012 International Joint Conference on.

Spiliopoulou, M. (2000). Web usage mining for web site evaluation. *Communications of the ACM*, *43*(8), 127–134.

Srikant, R., & Agrawal, R. (1996). *Mining sequential patterns: Generalizations and performance improvements.* 5th International Conference on Extending Database Technology, Avignon, France, March 25–29.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P.-N. (2000). Web usage mining: Discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter*, *1*(2), 12–23.

Streiner, D. L. (1994). Figuring out factors: the use and misuse of factor analysis. *The Canadian Journal of Psychiatry*, *39*(3), 135–140.

Trkman, P., McCormack, K., De Oliveira, M. P. V., & Ladeira, M. B. (2010). The impact of business analytics on supply chain performance. *Decision Support Systems*, *49*(3), 318–327.

Tseng, V. S., & Lin, K. W. (2006). Efficient mining and prediction of user behavior patterns in mobile web systems. *Information and Software Technology*, *48*(6), 357–369.
Van den Beemt, A., Buijs, J., & Van der Aalst, W. M. (2018). Analysing structured learning behaviour in massive open online courses (MOOCs): An approach based on process mining and clustering. *International Review of Research in Open and Distrbuted Learning*, *19*(5), 36–60.
van der Aalst, W. M., & Weijters, A. (2004). Process mining: A research agenda. *Computers in Industry*, *53*(3), 231–244.
van der Aalst, W. M. (2011a). *Process mining: Discovering and improving Spaghetti and Lasagna processes*. Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on.
van der Aalst, W. M. (2011b). *Process mining: Discovery, conformance and enhancement of business processes*. Berlin, Germany: Springer Science & Business Media.
van der Aalst, W. M., Adriansyah, A., de Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., … Buijs, J. (2012). *Process mining manifesto*. Business Process Management Workshops.
van der Aalst, W. M., & Günth, C. (2007). *Finding structure in unstructured processes: The case for process mining*. 7th International conference on Application of Concurrency to System Design, Bratislava, Slovakia. 10-13 July.
Van der Heijden, H. (2004). User acceptance of hedonic information systems. *MIS Quarterly*, *28*(4), 695–704.
Van Helden, J. (2003). Regulatory sequence analysis tools. *Nucleic Acids Research*, *31*(13), 3593–3596.
Werbach, K. (2014). *(Re) defining gamification: A process approach*. 9th International Conference PERSUASIVE 2014, Padua, Italy, May 21–23.
Williams, L., & Pennington, D. (2018). *An authentic self: Big Data and passive digital footprints*. International Symposium on Human Aspects of Information Security & Assurance (HAISA 2018).

Appendix A. Factor analysis results for different window sizes.

Behavior patterns B1–B7 shown in Table 5 are patterns that appeared when using 42 motifs in the EFA phase. These behaviors occurred again when using 57 motifs, mostly with richer patterns.