Mining Useful Information from Big Data Models Through Semantic-based Process Modelling and Analysis

Kingsley Okoye
University of East London, London, United Kingdom

Follow this and additional works at: https://scholarworks.lib.csusb.edu/ciima

Part of the Management Information Systems Commons

Recommended Citation
Available at: https://scholarworks.lib.csusb.edu/ciima/vol16/iss2/1

This Article is brought to you for free and open access by CSUSB ScholarWorks. It has been accepted for inclusion in Communications of the IIMA by an authorized editor of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.
MINING USEFUL INFORMATION FROM BIG DATA MODELS THROUGH SEMANTIC-BASED PROCESS MODELLING AND ANALYSIS

1. INTRODUCTION

This paper examines how effective process mining and analysis methods could be used to discover useful sets of recurrent behaviors within an existing process knowledge base or information system, using the learning process as a case study. The goal is to provide an automated system that can analyse in an abstract manner based on prior knowledge about a user’s learning behavior during a learning process. This is necessary to provide continuous intelligent recommendations of learning patterns and/or improvement on the learner’s performance. As a result, suitable learning paths are determined, which can then be used to address the problem of mining the various learning patterns for the individual process elements (user profiles). Essentially, the work makes use of the semantic modelling approach to support the discovery of previously unobserved learning behaviors or patterns through semantically annotated event logs and models. In other words, the method makes use of the process mining approach in order to discover learning patterns automatically, by means of semantic reasoning. According to Mancera, Baldiris, Fabregat, Vinas, and Caparros (2011), the lack of conformance and/or suitability of automated learning contents has an increasingly debilitating impact on learning, which in turn has a strong influence on expected learning outcomes. Studies have shown that among the challenges to current information systems are not only making information available for learners at any time or in any form, but also offering the right content to the right users and in the right format (Fischer, 2001; Yu, Nakamura, Jang, Kajita, & Mase, 2011). Ideally, flexible content delivery of automated learning should support adaptive learning recommendations that consider prior knowledge about learners (user profiles), learning behaviors, and learning goals. Moreover, meeting the goals of learners in today’s learning management systems might be accomplished by providing a well-designed and fit-for-purpose learning model that adapts to different learning requirements. Thus, the focus of this work is to develop an automated learning process management approach that can adapt and provide useful information based on prior knowledge about a user’s learning behavior, particularly during the execution process. The conceptual knowledge extracted from the captured learner’s data can be semantically annotated to automatically suggest appropriate future learning paths to improve the learning progression.

Process mining allows the extracting of information from captured user data sets. The technique is successfully applied to classical mining of processes where each process execution is recorded in terms of events log sequences (Nooijen, van Dongen, & Fahland, 2013). According to van der Aalst (2016), each event within a process is related to exactly one case and is assigned a case identifier (van der Aalst, 2011), which results in automatic creation of workflows achieved by using a semantic scheme to represent the event logs and models about the users (Murzek, Kramler, & Michlmayr, 2006).

The work in this paper differs from previous works in several aspects. First, it provides an automated learning process through the semantic rule-based approach to adaptively support learning for different learners. We focus on personalizing the learning-process data based on user behavior and executed activities as opposed to most existing learning systems, which provide learning guidance based on the views of a few designers or experts. Second, in addition to managing the learning process and models, this work also supports the discovery of useful information from the models through process mining techniques. Third, in terms of
user interaction patterns, the current work mainly considers how to discover a user’s unobserved learning behavior in relation to prior knowledge about the user. Hence, this work is not only intended to enhance the learner’s ability to learn and meet their learning needs, but is expected to be useful in providing learning paths and guidance based on individual differences. This contribution is achieved by collecting users’ initial capabilities and preferences on interaction and then using semantic modelling and process mining techniques to represent and detect behavioral changes as well as determine which adaptations or further assistive measures are best suited or may be required through time.

The rest of the paper is structured as follows: in Section 2, appropriate related work is analyzed and discussed. This is followed by a description of mining concepts and techniques related to the topic area of the research in the following section. Section 3 describes some of the data mining techniques, concepts, and practices and then shows how we semantically apply the representations for learning processes in order to draw conclusions and make predictions based on further analysis of the data. Section 4 presents the proposed learning model to express both user profiles and learning components. In addition, it discusses the generation of the learning process model and the method by which we semantically annotate the process model, describing in detail its representation (ontology) and reasoning using process description languages such as the ontology web rule language (OWL). Section 5 shows the method for automated discovery of learning patterns. The prototype implementation and preliminary outcomes are discussed in Section 6. Finally, Section 7 identifies directions for future research.

2. BACKGROUND INFORMATION

Presently, many organizations focus on applying process mining (van der Aalst, 2016) technologies to different aspects of their business processes. Interestingly, most of the developed systems make use of the process mining techniques only for representation of their business concepts, knowledge, or data (van der Aalst, 2011). By contrast, the method in this paper utilizes the process mining technique together with the semantic modelling technique to represent learning processes. The purpose is to further enhance this area of research by not only adapting the process mining tools but also presenting a way to relate semantic-based reasoning, especially for computing various process instances found within a learning knowledge base. For instance, given a set of activities or an event log of a process domain (e.g. learning process), the proposed approach automatically constructs process models capable of describing, classifying, and enhancing observed patterns or behaviors.

Fahland and van der Aalst (2012) note that it is difficult to learn useful models from event logs following the characteristics of real-life events. However, process mining has been proved as one of the existing technologies that may be used to extract non-trivial and useful information from event logs (van der Aalst, 2016; IEEE Task Force on Process Mining, 2012). Past studies in this area argue that heuristic mining, genetic mining, and fuzzy mining (Gunther & van der Aalst, 2007) provide case-hardened process discovery techniques capable of constructing simple, intuitive models to explain the most likely or common behaviors. Process discovery, which has been treated as the most important and most visible intellectual challenge related to process mining, aims to automatically construct a process model, e.g. a Petri net (Murata, 1989) or BPMN model (White & Miers, 2015), and describes causal dependencies between the individual activities (Calders, Guenther, Pechenizkiy, & Rozinat, 2009). In principle, one could use process discovery to obtain a model that describes reality.
The second type of process mining is conformance checking (Adriansyah, van Dongen, & van der Aalst, 2011) in which an existing process model is compared with an event log of the same process to check if in reality it conforms to the resulting process (Munoz-Gama & Carmona, 2011; Rozinat & van der Aalst, 2008; Weerdt, Backer, Vanthienen, & Baesens, 2011). Conformance checking could imply that the model does not describe the executed process as observed in reality or that it is being executed in a different order. It could also mean that activities in the model are skipped in the log or that the log contains events not described by the model. Given these drawbacks, the last type of process mining, model enhancement, proves useful. Van der Aalst, Schonenberg, and Song (2011) used the idea of an enhanced existing model to maintain compliance and to quantify deviations using information about the actual process recorded in some event logs from a business process.

Nooijen et al (2013) recently introduced an automatic technique (artifact life cycle model) for discovering structured processes from any given data source, by using a number of existing techniques to fill in crucial gaps in each concept of the data objects in data-centric systems. In order to fill these gaps, event-types specification is used to construct queries that extract attributes from all event logs, group them into cases, order them by time stamps, and finally write the result into a classical log in separate database columns. According to Verbeek, Buijs, van Dongen, and van der Aalst (2011), the most recent generic approach to event log extraction is XESame, which manually defines mapping between the source data and event logs, sorts them into traces, and then translates their mappings to SQL queries, which are subsequently stored in a database. Stored data can be queried to retrieve the sets of events from the logs in central process data systems, as we express in this paper using semantic rules (Horrocks et al, 2004). Many approaches have been tested to extract event logs from ERP (Enterprise Resource Planning) systems such as SAP (Ingvaldsen & Gulla, 2008; Piessens, 2011) and PeopleSoft (Ramesh, 2006). Consequently, as ERP systems in general provide multiple case identifiers, the majority of these approaches failed. Nooijen et al (2013) argue that event logs can only be successfully extracted when database tables are carefully selected by hand or a better view of data is semantically annotated to ease its analysis. Accordingly, this work uses the semantic rule-based approach to extract data from annotated data objects to create rules capable of detecting changes or similarity in learning patterns and/or behaviors.

Other researchers have also focused on discovering common structures that can be found in a variety of processes by describing Workflow Activity Patterns (WAPS) (van der Aalst, 2011). Indeed, various definitions of workflow have been proposed in literature (Agrawal, Gunopulos, & Leymann, 1998; Ferreira & Thom, 2012; Fischer, 2002; Jablonski & Bussler, 1996; Thom, Reichert, & Iochpe, 2009). Thom et al. (2009) describe WAPS as structures involving the interaction between the user and the control-flow constructs used to model the semantics of the activities being performed. Workflow systems assume that a process can be divided into small, unitary actions called activities (Agrawal et al., 1998). To perform a given process, one must perform the set or a subset of the activities that comprise it. In addition, there may be dependencies between different activities. Hence, an activity is an action that is a semantic unit at some level, which can be thought of as a function that modifies the state of the process in terms of the semantics of the patterns, and which can be discovered automatically by means of semantic reasoning (Okoye, Tawil, Naeem, & Lamine, 2016a, 2016b).

In recent years, the concept of workflow management has been applied in many enterprise information systems (van der Aalst & van Hee, 2004; Fischer, 2002; Jablonski & Bussler,
such as Staffware, IBM, MQSeries, and COSA, which offer generic modelling and enactment capabilities for structured processes. Many other software systems have also adopted workflow technology. For example - ERP systems such as SAP, PeopleSoft, Baan, and Oracle CRM (Customer Relationship Management) software. However, despite its advantages, many problems still emerge when applying the workflow technology in different settings. One such problem is that these systems require a workflow design (van der Aalst, Weitjers, & Maruster, 2004), whereby, a designer must construct a detailed model accurately describing the routing of work, which most often requires deep knowledge of the workflow language and/or management involved. Another example of the challenges encountered: creating a workflow design is a complicated time-consuming process, and typically there are discrepancies between the designated workflow process and the actual processes as perceived by the management.

Huang and Shiu (2012) noted that searching for suitable learning paths and content to achieve a learning goal is time consuming and troublesome, especially on dynamic learning platforms. To solve these problems, the authors propose a User-Centric Adaptive Learning System (UALS) that uses sequential pattern mining to construct adaptive learning paths based on users’ collective intelligence and recorded events, and then employs Item Response Theory (IRT) with a collaborative voting approach to estimate learners’ abilities for recommending adaptive materials.

The following section describes the technologies capable of transforming existing raw data within a learning knowledge base into meaningful and useful information that can be used to enable more effective reasoning and tactical strategies for adaptation and decision making.

3. DATA MINING AND PROCESS MODELLING

Process mining is not limited to automatic discovery of patterns within processes. It builds on data mining and process modelling techniques. However, some of the existing data mining approaches appear to be too data-centric in providing a comprehensive understanding of end-to-end process execution within processes. In this work, we use some data mining techniques to put the captured volumes of data within a learning knowledge base into a process context. The aim is to use the concept of data mining to describe and understand learning process reality based on captured knowledge or historic data. In terms of process mining, the focus is on providing technologies capable of transforming existing raw data into meaningful and useful information that can be used to enable more effective reasoning and tactical strategies for adaptation and decision making (van der Aalst, 2016; Forrester, 2010; van der Aalst, Adriansyah, & van Dongen, 2012). We explore and discuss learning process mining in the context of some basic data mining techniques, particularly as it concerns semantic modelling of the data within a learning process knowledge base.

**Concepts Mining and Data Representation**

Many definitions of data mining have been proposed in the existing literature. Hand, Mannila, and Smyth (2001) define data mining as the analysis of recorded data sets to find unsuspected relationships and to summarize data in novel methods that are understandable, meaningful, and useful to the data owner. According to van der Aalst et al. (2016), the input data is most often given as a table and the resulting data sets may be patterns, equations, graphs, tree structures, clusters, or rules. The discipline of data mining is increasingly characterized by
concrete scientific tactics and has numerous practical applications (Hand et al, 2001; Alpaydin, 2010; Witten & Frank, 2005; van der Aalst, 2011).

The use and adaptation of the input data described by van der Aalst (2016) for process mining builds on two pillars: data mining and exploration of process models. In this paper, we use some basic data mining techniques to describe how one can extract useful process models by focusing on the data resources, since process mining builds on classical data mining techniques. Our goal is to build on the useful information that comes from the data mining field to evaluate the result of the learning process models. Hence, a basic understanding of data mining is indispensable for a comprehensive use of the process mining technique as described in this paper. These techniques and they ways in which they are connected to the semantic modelling of the learning process case study are discussed in detail in the following sub-sections.

**Association Rule Learning**

Association Rule Learning aims to find rules that can be used to predict the value of some response variables that have been identified as important in the same way as the decision systems (Yarandi, Jahankhani, & Tawil, 2013), but without focusing on a particular response variable. This rule learning aims at creating rules of the form:

\[
\text{IF } X \text{ THEN } Y
\]

Where \( X \) is often called the **antecedent** and \( Y \) the **consequent**.

Hence, \( X \Rightarrow Y \)

This work treats the preceding rule as similar to the semantic web rule language (SWRL) (Horrocks et al, 2004), which we used in providing a more ontological description and enhancement to the learning process model.

The SWRL rule has the form: \( \text{atom} \land \text{atom} \text{ (antecedent) } \rightarrow \text{atom} \land \text{atom} \text{ (consequent)} \).

According to van der Aalst (2011), **association rule learning** strongly supports the use of metrics frequently expressed in terms of **support** and **confidence**. These expressions help measure the strength of the association rule. **Support** determines how often a rule applies to a given data set, which means the fraction of instances for which both antecedent and consequent hold. Therefore, a rule with high **support** is more useful than a rule with low **support**. A rule that has low **support** may occur simply by chance and is likely to be irrelevant from a learning perspective because it may not be profitable to monitor, recommend, and promote learning activities or learning patterns. **Support** can be used to evaluate learning process models and its execution where:

\( N_x \) is the number of instances for which, \( x \), learning activity holds;

\( N_y \) is the number of instances for which learning activity \( y \) holds; and

\( N_{x,y} \) is the number of instances for which learning activity \( x \) and \( y \) holds.

Consequently, **support** for the rule \( X \Rightarrow Y \) is described as: Support, \( s(X \Rightarrow Y) = \frac{N_{x,y}}{N} \)

where \( N \) is the total number of instances.

**Confidence**, by contrast, measures the reliability of the inference made by a rule over a learning process model. Hence, for a given rule with the form, \( X \Rightarrow Y \), the higher the
**Confidence**, the more likely it is for the consequent $Y$ (learning pattern extension) to be represented within the learning process that contains $X$ (learning patterns). In other words, **Confidence** measures the conditional probability that the extension $Y$ will happen, given $X$.

Confidence, $c(X \Rightarrow Y) = \frac{Nx_y}{Nx}$

Overall, inferences made by an association rule learning suggest co-occurrence of relationships between items in the antecedent ($X$) and consequent ($Y$) of the rule. Therefore, for every given set of activities or item set, there exist rules having:

- $support \geq minSup$ and
- $confidence \geq minConf$

where: $minSup$ and $minConf$ are respectively the corresponding $support$ and $confidence$ thresholds.

In terms of the learning process knowledge base (ontology) that was created for the experiment in this paper, these metrics were used to dramatically reduce the exploration or drilling down space when constructing the set of the frequency or sequence of the activity logs. The simple requisite is that $X$ and $Y$ are non-empty and any variable appears at most once in $X$ and $Y$. For instance, the following association or relationship applies:

**IF Learner(X) AND hasLearning_Activities(Y) THEN hasLearning_Process(X, Y)**

Thus, $Learner(?X) ^ hasLearning_Activities(?X,?Y) ^ Activity(?Y) \rightarrow hasPartLearning_Process(?X,?Y)$

Indeed, such an approach has been used to provide process specifications and descriptive languages that are logically comprehensive, e.g., by using the OWL (Bechhofer et al., 2004) and the Knowledge Interchange Format (KIF) (Obitko, 2007) based on Description Logics (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2003), which are fundamental to knowledge representation formalisms. The KIF makes it possible to understand the meaning of unique logical expressions through declarative semantics. From the learning process example, it can be expressed that:

"Every Learner has a Learning_Activity."

Hence,

$$( \forall X ) \quad \rightarrow \quad ( Learner X ) \quad \exists Y \quad ( and \quad ( Learning_Activity X Y ) ) )$$

Likewise, "Every Learning_Activity is part of a Learning_Process" and must have some kind of a Learner. Hence the expression:

$$( \forall X \quad \exists Y ) \quad \rightarrow \quad ( Learning_Process X Y ) \quad ( and \quad ( someLearner X ) \quad ( someLearning_Activity X Y ) ) )$$
The rule as expressed using the KIF format suggests that a strong relationship exists between Learning_Process and Learner. This is because Learner(X) has_Activities described as a Learning_Activity, and Learning_Activity has been described as PartOfLearning_Process etc.

In short, designers of large knowledge base systems (e.g. big data) and analysts can use these types of rules to help identify new opportunities, especially for enhancement of the discovered process models. The association rule is currently being used in application domains such as web mining and scientific data analysis, much as we used it in this paper. In the defined process model, the association patterns reveal interesting connections among domain entities, individual classes, objects, and data properties. Such information may help the model to provide and create a better understanding of how the different elements within the Learning Process knowledge base relate and interact with each other.

**Sequence Mining**

Sequence Mining addresses the problem of analyzing sequences of item sets, unlike some other algorithms that do not consider the order in which events are performed. This approach to data mining was first proposed by Srikant and Agrawal (1996), where each event within a process has an event identification, time of execution (timestamp), and a set of activities (item set).

The focus of sequence mining is to discover frequent sequences defined by a pattern. Given a set within a learning process. For instance, where:

\[
X = \{A, C, D, B, E\} \text{ and } Y = \{A, B, C, F, E, G, H, I\} \]

exist, the frequent set of activities \{A, B, C, E\} can be discovered because of the predefined proportion of the data within the sequence set.

Hence, the expression that sequence \(<a_1, a_2, .... an>\) is a subsequence of another sequence \(<b_1, b_2, .... bn>\) if there exist integers \(<i_1, i_2, .... in>\) such that:

\[
a_1 \subseteq b_{i1}, a_2 \subseteq b_{i2}, .... an \subseteq b_{in} \quad (\text{van der Aalst, 2011}).
\]

According to van der Aalst (2011), the support of a sequence, \{s\}, is the fraction of sequences in the data set that has \{s\} as a subsequence. This means that a sequence is frequent if its support fulfills some threshold minsup.

For instance, the sequence \{(A), (A, D), (D)\} is a subsequence of \{(B), (A), (B), (A, D, B), (D, B), (B)\} because:

\[
\{(A)\} \subseteq (A) \\
\{(A, D)\} \subseteq (A, D, B) \\
\{(D)\} \subseteq (D, B)
\]

Also, it is important to note that: \{(A), (B)\} is not a subsequence of \{(A, B)\} etc.

The example reveals that in sequence mining, it is possible to proficiently discover and generate patterns, which further can be used to create rules of the form \(X \Rightarrow Y\), as mentioned earlier in the discussion of association rule learning. \(X\) is regarded as an existing learning sequence (antecedent) and \(Y\) is the possible learning subsequence extension (consequent). In the developed learning model in this paper, the idea of sequencing is used to semantically annotate and create attributes for the various activities within the learning knowledge base.
The method helps in classification of the entities (process instances) within the knowledge base based on the frequency of pattern (entities, classifications, or taxonomy) in which the activities are performed and the relationship they share with other sets of classes.

**Clustering**

The concept of clustering as a data mining technique aims at grouping the set activities (referred to as instances) into clusters. A set of activities (instances) in a group should be similar to each other, and in many circumstances may be unrelated to the set of instances in other clusters. Many clustering algorithms have been proposed in literature (Bramer, 2007; Hand et al, 2001; Witten & Frank, 2005; van der Aalst, 2011). For the purpose of this paper, we focus on *k*-means clustering (van der Aalst, 2011).

![Clustering of Process Instances/Activities](image.png)

Figure 1: Clustering of Process Instances/Activities.

Figure 1 shows the importance of the clustering technique toward performing process mining. Assume we have a set of activities with the variables *Learning_Actions* and *Classes* as described in detail in the model developed in the next section of this paper (Figure 13). It is possible to discover the four clusters shown in Figure 1 using the *k*-means clustering technique. The different clusters correspond to individual entities that have similar attribute(s) within a class. Basically, the entities within a particular cluster are close to one another and are closely related or share similar object/data properties, and as such support further enhancement and are useful in addressing the problem of analyzing relationships among item sets within the learning process, rather than analyzing sequences of item sets as with sequence mining. According to van der Aalst (2011), *k*-means clustering is a *distance-based* algorithm that has been proposed in literature by assuming a *distance notion*. The method considers each instance within the process to be an *n-dimensional vector*, where *n* is the number of variables that modestly takes the *Euclidian* distance.

Hence, the ordinal *n*-values needs to be either true = 1, false = 0 or cum laude = 2.

By referring to *k*-means clustering, we envisage an approach that streamlines a learning process by discovering classes with the value *k* = *n* and their associated instances. The *k*-means approach starts with an arbitrary centroid denoted by + symbol. In our developed model, these + symbols refer to the individual activities that make up the learning process, and by using the executed distance metrics (true = 1, false = 0, cum laude = 2) we assign all instances (i.e. learning activities) that are closest to a defined *centroid* (class). The result of the method leads to the effective modelling of the classes within the learning process model.
by ontologically describing the domain entities, subclasses, and the associated instances of a class. In fact, the quality of a particular clustering is defined by the average relationship of activities within the process model to the model’s corresponding class. In the work of Witten and Frank (2005), one of the problems that arises when using the k-means algorithm is determining the number of clusters, $K$, as these clusters are fixed from time of creation. However, van der Aalst (2011) argues that as $K$ is increased, the average relationship of an instance to its corresponding class decreases, which is not very useful. The solution is to start with a small number of clusters and then progressively increase $K$ as long as there exists a substantial improvement.

**The Agglomerative Hierarchical Clustering (AHC)**

AHC is another popular clustering technique that addresses the problems identified with the k-means clustering by generating a variable number of clusters (Fernández & Gómez, 2008; van der Aalst, 2011). The method works by assigning each activity within the process to a devoted singleton cluster and then searching for the two clusters that are closest to one another. The goal is to merge these two clusters into a new cluster. On one hand, the resulting model or class hierarchy is more like a Decision tree learning model (Han, Wang, & Bryant, 2008). On the other hand, the AHC is a very simple but useful technique for process mining. In Figure 2, the initial clusters consisting of $K1$ and $K2$ are merged into a new cluster, namely $K1K2$. Additionally, the approach searches for other clusters that are closest to $K1K2$ and merges them as well. This process is repeated until all activities are merged in the same cluster, i.e. $K1… K10$.

Figure 3 illustrates the clustering that results from the hierarchical process of the AHC.

![Figure 3: The Agglomerative Hierarchical Clustering Technique.](image-url)
As shown in Figures 2 and 3, *clustering* can be used as a pre-processing step for modelling processes in real-time settings or executions (Bose & van der Aalst, 2009; Greco, Guzzo, Pontieri, & Saccà, 2006). With this technique, it is possible to construct fractional process models that are novel and easy to understand by grouping similar cases together, which can then further be simplified by discovering related cases in each cluster.

Figure 4, for instance, illustrates some of the fractional learning clusters within the developed learning model. The illustration shows how we represent our data at two levels: *process level*, which consists of the sub-processes within the learning knowledge base, and *data level*, which consists of all the various units (learning activities) that group into different individual sub-processes. This approach is what we first use to define the *Named Classes* in the learning process domain ontology, and subsequently use as a reasoner (semantic reasoning or classification) to create inferences capable of discovering newly unobserved classes or relationships based on the defined rules (object/data type assertions) or underlying expressions within the learning knowledge base.
Figure 4: High-level Definition of Process Model at Process and Data Levels.

**Decision Tree Learning (DTL)**

DTL, unlike clustering, focuses on the classification of activities within the learning knowledge base correlated to discoverable variables that are predictable (Han et al, 2008). The method uses uncompromising response variables by classifying the learning activities and arranging the resulting value in the form of a tree. DTL consists of nodes that correspond to the possible values (leaf nodes) and the predictive variables (non-leaf nodes) that can be referred to as sets of *classes*. Each class in the tree splits a given set of nodes into two or more subsets (subclasses). With DTL, each instance within the tree is represented as a subclass of a domain class referred to as the *root node*. For instance, based on the attribute of the activities for the domain class (root node) within the learning model shown in Figure 5, the learning process splits into some that are leaf nodes and others that are non-leaf nodes.
According to the work of van der Aalst (2011) and Han et al (2008), DTL uses a recursive top-down algorithm expressed in terms of the root node, r, and all associated instances to the root node.

Where: \( x = \{r\} \) i.e. the set of nodes to be traversed. Hence

\[
\text{IF} \ x = \emptyset \ \text{THEN END} \\
\text{Else} \\
\quad \text{// Select and extract all subset of } x \text{ based on entropy} \\
\quad X: = X/\{x\}, \\
\quad \text{where } x \in X \text{ (} x \text{ is a subset of } X \text{ based on entropy)} \\
\quad \text{// Check if splitting is possible?} \\
\quad \text{IF} \ X: = \emptyset \ \text{THEN END} \\
\quad \text{Else} \\
\quad \quad \text{// create a set of Child nodes } Y \text{ and Add } Y \text{ to } X \\
\quad \quad Y: = X \cup Y \\
\quad \quad \text{// and Connect } x \text{ to all Child nodes in } Y \\
\quad \text{End}
\]

**Figure 6a: Decision Tree Learning Algorithm.**
As shown in Figures 6a and 6b, one of the basic functions of the DTL algorithm is to help designers and systems developers to define and decide when to stop adding nodes. This function is achieved by using the formula:

**IF X: = Ø THEN END.**

Thus, the formula determines when splitting is no longer possible. Until that point, the nodes continue to split through the following enabling function:

**Else //create a set of Child nodes Y and Add Y to X, (Y: = X ∪ Y), until the value of X: = Ø.**

In short, DTL is beneficial in improving the resulting nodes or in restricting the decision tree to a certain level, as indicated in Figure 5. This is necessary to determine the variation of the smallest unit (node) within the provided data by splitting the set(s) of activities into subsets using the property of entropy (van der Alast, 2011), as is shown in the following formula: 

\[ E = \sum_{i=1}^{K} P_i \log_2 P_i \]

Apparently, the more we split a node, the lower the entropy, until the overall root node reaches a definition value equal to zero. According to van der Aalst (2011) entropy represents a measure that is used to define and quantify the diversity in a leaf node to determine if splitting is possible or needed. In turn, DTL as we have gathered, is a technique useful in learning process mining to help locate all decision points within the process. For example, the paths taken or the attributes of the data sets known at, or prior to the decision point.

In the next section, we introduce the concept of process mining aimed at the discovery of worthwhile process models and automated analysis of the learning process domain.
4. PROCESS MINING TOWARD AUTOMATED LEARNING

Process Modelling and Automation.

One of the key challenges in developing automated systems for learning is to build effectively represented user profiles, learning styles, and/or behaviors to help support reasoning about each learner (Huang & Shiu, 2012; Nganji, Brayshaw, & Tompsett, 2011). It should also be possible to dynamically update the representation to account for the changing state of learners and the variations in the information that is relevant to each user over time. There remains the additional task of matching such learners (user profiles) with solutions that best fit their particular learning needs or requirements (Nganji, Brayshaw, & Tompsett, 2013) through the generation of rules during the learning process. A semantic rule-based approach toward automated learning is expected to collect routines and monitor changes in a user’s behavior during the learning process to determine which adaptation technique may be progressively required. The process is illustrated in Figure 7.

![Workflow Diagram for the Learning Process Activities.](image)

The method presented in this paper takes into account the user’s profile (prior knowledge of learner’s background), learning behavior, and actions (activities) when using the system. The workflow model in Figure 7 focuses on information about the learning process used to create the semantic model in this paper (Figure 13) consisting of the classes, properties, or individuals and how they are related. Moreover, the semantic modelling and analysis process is not only relevant during the design and requirement stage, but also for monitoring and enhancing the entire learning process. Typically, to perform process mining, data categories need to be captured. First, the identification of process instances is necessary, which is the modelling of learning process units or data about the different users. Further, these data must be selected from the learning knowledge to carry out the analysis on the captured data sets. For instance,
Figures 8 and 9 show how data about each learning process is extracted, prepared, and transformed into a machine-readable (minable), yet, machine-understandable (semantic-based) format. The results are analyzed and compared to a prior knowledge test to see if they correspond to the information within the learning knowledge base. Consequently, to enhance the information value of the resulting process models and usefulness of the whole system, it is necessary to enrich the mentioned instances or process objects. A process object may refer to the learner, for instance, and the outcome may refer to the purpose of performing the process mining. Thus, process modelling discussed here is the application of the fundamental concepts of process mining and semantic modelling. Additionally, one of the major benefits of semantic-based process modelling is that it increases capacity of large knowledge bases by improving resources through model enhancement. It also reduces cost by removing waste and helps mitigate risk through conformance checks.

Figure 8: Process Mining of Learning Process Event Data Log.

Figure 9: Example of a Process Model With Semantic Annotated Markup Language.
As shown in Figure 9, semantic process modeling related to this work is described as a set of *semantic rules* that is used to model different activities within a learning process by using the events logs, which are transformed into minable formats, e.g. the Semantically Annotated Mining eXtensible Markup Language (SA-MXML) (de Medeiros & van der Aalst, 2009), and eXtensible Events Streams (XES) (van der Aalst, 2016). Moreover, these rules or associations incorporate references between elements in the events logs and concepts in an ontology based on the OWL (Bechhofer et al, 2004) and SWRL (Horrocks et al, 2004) that is layered on top of the existing information asset to provide additional enhancements to the learning processes.

Therefore, the following subsection describes the semantic-based process mining and analysis framework, which the work has proposed for ample implementation of the method in this paper.

**Semantic-based Process Mining and Analysis Framework (SPMaAF)**

The design of the SPMaAF, detailed by Okoye, Naeem, & Islam, (2017) is primarily constructed on building blocks shown in Figure 10.

![Figure 10: The Semantic-based Process Mining and Analysis Framework (SPMaAF).](image)

Figure 10 illustrates the proposed framework for the semantic-based process mining and analysis technique in this paper which are constituted by the following processes:

- **Extraction of process models from event data logs** are represented as a set of annotated terms that links and relates to defined terms in an ontology, and in so doing, encodes the process logs and the deployed models in the formal structure of ontology (semantic modelling).

- **The inferred ontology classifications** help associate meanings to labels in the event logs and models by pointing to concepts (references) defined within the ontology.

- **The reasoner** (inference engine) is designed to perform automatic classification of tasks and consistency, checking to validate the resulting model as well as clean out inconsistent results, and in turn, presents the inferred (underlying) associations.
The conceptual referencing supports semantic reasoning over the ontologies to derive new information (or knowledge) about the process elements and the relationships they share among themselves within the knowledge base.

To summarize the design framework, the key step to the application of the semantic-based process mining and model analysis approach is to connect the mining algorithms with two key core elements:

1. Event logs and process models where the labels have references to concepts in an ontology; and
2. Reasoners that are invoked to reason over the resulting ontologies for the logs and models.

The use of this type of semantic-based framework and its application has gained significant interest within the field of process mining. The SPMaAF framework uses the semantics captured in event data logs (i.e. metadata) to create new methods for process mining, or better still support the enhancement of existing ones. The framework may assist humans in gaining novel and more accurate results at a higher conceptual level, mapping to the domain context, as opposed to the traditional process mining techniques that tend to analyze data at the syntactic level. Further, because of the semantic level of analysis, the outcome of the technique can be understood easily by the process owners, process analysts, or IT experts. Event logs from the various process domains usually carry domain-specific information (semantics), but quite often, the traditional process mining techniques and algorithms lack the ability to interpret or make use of such semantics across the different process domains. In other words, while the traditional process mining technique trails to analyse the events data logs at syntactic levels (i.e. labels or tags in the event logs), the SPMaAF extends and analyzes the available events data logs and derived process models at a much more conceptual level.

**Process Mining and Learning Pattern Discovery.**

Process discovery is simply a technique used to discover, monitor, and improve real processes by extracting knowledge from the event logs about the domain process in view. The method allows for the discovery of traces not present in a given process (Bose & van der Aalst, 2009; Greco et al, 2006; Gunther, 2009). In this paper we used a semantic rule-based process mining approach to classify instances based on discoverable variables to show the processes in a more detailed way. We show that given an events log about any process (case study of the learning process), one can determine the dependent variables in terms of independent ones, which complement the way we look at processes. Thus, the data and the discovered models can be analyzed based on the process instances and variations.
In Figure 11, we imported the learning process data into Disco tool (Rozinat & Gunther, 2012) to derive a fuzzy model that allows us to visualize in detail how the processes have been performed. The process mappings allow us to focus on the stream of behaviors and to see the paths they follow in the process. Case id tags were used to assign the identifier for process instances and Activity tags for the set of tasks that are performed during the learning process. We associate Timestamp tags with activity instances for the purpose of sequencing. The time performance shows how often each task is executed in terms of frequency of each activity in the process model. This is achieved by using the Frequency Analysis to determine how often a given process is performed. The variants show the process in a more detailed manner by revealing all the cases that have been created during the process execution. Accordingly, the most frequent variants can also be determined. While the Map view gives a visual understanding about the process flows and the Statistics view provides detailed performance metrics about the process, the Cases view actually goes down to the individual case level and shows the raw data. Thus, in order to inspect individual cases, it is important to verify the findings and see concrete examples, particularly for unexpected behavior that may be revealed during the process analysis.

In fact, the process mining approach proved to provide reliable and trustworthy results for data sets of arbitrary complexity and can be understood efficiently by domain experts with no prior experience in process mining.

Although the Disco tool is based on the proven framework of the fuzzy miner (Gunther, 2009; Gunther & van der Aalst, 2007), in this paper we developed a completely new set of process metrics and modelling strategies using the semantic rule-based approach. These additional metrics prove useful for semantic-based process analysis because they hold relevant contextual information (domain-specific characteristics) as opposed to the traditional process mining techniques. For instance, the fuzzy miner shows these metrics and statistics, but we need to focus our analysis and to split out and compare the processes with respect to these...
characteristics. In total, there are six powerful filter types available in Disco (Rozinat & Gunther, 2012), and they can be combined and stacked in any order. However, in this paper we focus on the Attribute filter (Rozinat & Gunther, 2012), which describes as well as excludes certain activities, resources, or process categories based on data attributes. In addition to the analysis views, the filtering capabilities allow us to quickly and interactively explore processes in multiple directions and to answer concrete questions about the process in a flexible manner. Insofar as the filtering is capable of transferring the data and information within a short period of time, we can also hold inference reasoning and generate process improvement ideas along the way. This is what semantic rule languages such as the OWL described in this paper allow us to achieve.

**Learning Process Event Data**

The minimal requirement for process mining shows that any event within a process can be related to both a case and an activity. Additionally, these events within the different cases are ordered (Zaki & Wong, 2003; Liu, 2005). Typically, from the set of event logs, we use the information in each row to obtain a more compact representation of each case by using the process mining concepts to support the ontological reasoning of the sequence of activities, also referred to as actions. Figure 12 below shows the ontological representation of the structure of a typical events log.

![Ontological Description of Events Data Log for a Typical Process](image)

Figure 12: Ontological Description of Events Data Log for a Typical Process.

In Figure 12, we gather that:

1. An Individual process consists of cases.
2. A case consists of events in a sequence and that each event is related to a particular case.
3. Events within a case are sequentially ordered.
4. Events are made up of attributes e.g. Event_ID, ActivityType, Date_TimeStamp etc.

The following section describes and implements the ontological modelling and reasoning of the learning activities capable of deducing inference knowledge. We provide a semantic rule-base system that serves as a conceptual model for implementation of the proposed semantic approach to automated learning.
Semantic Modelling and Reasoning of the Learning process

Ontology is one of the most efficient tools that can be used to model different kinds and structure of objects, events, processes, and behaviors as they happen in reality (Okoye, Islamm & Naeem, 2018) such as it is used in the learning process in this paper. It is a discipline that is not only devoted precisely to the representation of events as they happen, but is also useful in the formulation of robust and sharable descriptions of a given process domain for an enhanced reasoning capability. Such enhancement of the process domain results in an increased knowledge awareness and performance. Moreover, ontology provides the schema and common vocabularies for integrating across diverse data sets (Bechhofer et al., 2004; Tawil, Lithhouvongs, Cevalier, & Taweel, 2011).

In this section, we demonstrate the resultant process model used for the purpose of this work. Figure 13 shows an OWL ontology model for the learning process model, which we implemented in Protégé and reasoned using Pellet 2. Protégé OWL editor supports Description Logic (DL) Queries (Baader et al., 2003) and SWRL rules (Horrocks et al, 2004). The reasoner Pellet also supports the SWRL rules and better still, logical and/or taxonomical classifications.

Figure 13: The Learning Ontology Model in Protégé Editor.

Figure 13 demonstrates the use of the Protégé Editor to construct an ontology that allows us to express the functionality of the learning model in terms of the individual learning patterns. The cases and actions (learning activities) within the learning process were defined as subclasses of the main class DomainEntity. The classes and individual property expressions are based on the OWL syntax - fundamentally focused on collecting all the information about a particular class, object/datatype property, or individual into a single construct called a frame. Additionally, the DL Query provides the platform for searching the underlying ontology. For our model, DL Queries were used to reason about the OWL individuals, primarily in terms of the OWL classes; Object and Data Properties were used to infer the learning activities of any named individual. For instance, Figure 14 represents the process of executing the query refer explicitly to OWL.
individual **Ben_Steward_27**. The result of the DL Query produces the instance value of Ben_Steward_27 Actions within the learning model, as shown in Figure 15.

![Figure 14: DL Query Describing the Actions of a Named Individual.](image)

![Figure 15: Result of the DL Query.](image)

We use the result of the logic expression and reasoning of Actions for **Ben_Steward_27** as information for automated discovery of learning patterns as implemented in subsequent sections. In addition, the expressivity of our OWL is extended by adding SWRL rules to the implemented ontology based on the concrete syntax of the SWRL proposal, which are similar to rules in Prolog or Datalog languages. The SWRL rules provide similar strong formal guarantees to our performed inferences. The motivation is to propose a process model based on such a designed rule base, which serves as a conceptual model for building our proposed semantic rule-based method, which validates our approach and supports future, more reliable developments.

## 5. AUTOMATED DISCOVERY OF LEARNING PATTERNS/BEHAVIOR

The method in this paper allows for traces not present in the existing process model to be discovered by using semantic rules to generalize and allow for the observation of behaviors unrelated to the ones within the knowledge base. The model interpretations are further enhanced by revealing the most likely underlying model that is not invalidated by the next set of observations. The $\alpha$-algorithm is one of the many algorithms used in process mining that aims at reconstructing connectedness from sets of events sequences. It was first proposed by van der Aalst et al. (2004). Since its proposal, several extensions of the algorithm have been presented and used, as in the description below.

If the set of learning activities as implemented in Figures 13 to 15 are analyzed where:
Activity: notation

\[ A = \text{Enrollment} \]
\[ B = \text{Lesson} \]
\[ C = \text{Assessment} \]
\[ D = \text{Feedback} \]

These Activities, as they occur sequentially, take a workflow process log, \( W \subseteq T^* \) as input and result in a workflow net being constructed.

The Workflow Logs (\( W \subseteq T^* \)) is a definitive relationship management algorithm where:

- \( W \) is a finite set of Events.
- \( T \) is a finite set of transitions such that \( (W/T)^{1/4} \)

Hence, \( W \subseteq T^* \) is a set of directed pattern, called the flow relation.

Figure 16: Control Flow of Process Model Corresponding to the Workflow Log.

Figure 16 illustrates that it is easy to check that all the traces described within the model are possible. The initial marking (A) is enabled because of the token at the start of the learning activities. The control flow for its execution (\( X \Rightarrow Y \)) results in the marking that a learner, for instance, performs an activity (A) that is followed by either activities (B and C) or E, and then D (van der Aalst, 2011; Zaki & Wong, 2003). Thus, the execution of the activities from the start event (A) to the final event (D) in the path is modeled in this way.

Information about the activities (actions) and semantic representations and reasoning are used to add more performance-related information to the knowledge base. For instance, from the model in Figure 13, user profile (Ben_Steward_27) learning behavior will be described by the sequence of his or her activities in each case as follows:

Table 1: Example of Performed Activities Within the Learning Process

<table>
<thead>
<tr>
<th>LearnerProfile: (Ben)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action(1) = {Login}</td>
</tr>
<tr>
<td>Action(2) = {Registration}</td>
</tr>
<tr>
<td>Action(3) = {Search, Content_View}</td>
</tr>
<tr>
<td>Action(4) = {Typing, Editing}</td>
</tr>
<tr>
<td>Action(5) = {Save, Upload}</td>
</tr>
<tr>
<td>Action(6) = {Start_Lesson}</td>
</tr>
</tbody>
</table>

Hence, the sets of activities using the \( \alpha \)-algorithm [32] for the subject’s learning behavior discovered from the frequent sequence of the learning profile, as shown in Table 1, is as follows:
(Action[1-6]) = {Login, Registration, Search, Content_View, Typing, Editing, Save, Upload, Start_Lesson}

Clearly, the definitions allow for the prediction of the value of some other probable variables regarding the patterns or behaviors that have been identified as being important. The identified patterns are then modelled using the association rule (Han, Kamber, & Pei, 2011; Liu, 2005; Zaki & Wong, 2003). Likewise, the method in this paper aims to discover similar rules without focusing on a particular variable to discover user interaction patterns during the learning process. The main goal is to discover and create rules of the form:

\[ X \rightarrow Y \]

i.e. IF \( X \) THEN \( Y \),

Where \( X = Learning\) pattern (Antecedent) and \( Y = Learning\) pattern extension (Consequent)

This rule is similar to the SWRL Syntax: \( atom \wedge atom \ldots \rightarrow atom \wedge atom \)

e.g Learner (?X) \wedge hasLearningActivity (?X, ?Y) \wedge LearningActivity (?Y) \rightarrow hasLearningProcess (?X, ?Y)

Driven by these variables, from the Action (activities) logs in Table 1, it is possible that the following learning path can be suggested to improve the performance of unobserved or unnamed users.

Rules like “Learners that have similar instances as Student_Ben are most likely to come across Case(Ben)” can be derived. Thus the expression: Student(?X), isPartOf (?X, ?Course), hasActionSimilar (?Actions, “Ben”) \rightarrow hasSimilarLearningPathTo(?X, “Ben”)

Similarly, the association rule states that when \( X \) occurs, then \( Y \) occurs with certain probability by using the frequent item set \( (I) \) to generate rules (Liu, 2005). This implies that for each process of non-empty Activity/Action \( (A) \) from the learning model in Figure 13:

\( X \Rightarrow Y \) is an association rule of the form; \( Y = (X - A) \) where, Confidence \( (X \Rightarrow Y) \geq \) minimum Confidence

\[
\begin{align*}
\text{Support} (X \Rightarrow Y) &= \text{support} (X \cup Y) = \text{support} (A) \\
\text{Confidence} (X \Rightarrow Y) &= \text{support} (X \cup Y) / \text{support} (A)
\end{align*}
\]

Consequently, if \( X = Case(Ben) \) and \( Y = Ben(\text{Action}[1-6]) \) and \( X \) and \( Y \) are represented according to utilization factor of frequency (van der Aalst, 2011) ranging from 0.01 to 0.09, where \( X \) has a support of 0.04 and \( Y \) has a support of 0.03 from the represented variables, then the expression of certainty \( X \Rightarrow Y \) is 0.03 / 0.04 = 0.75.

This means the prediction variable that 75% of Learners that have instances of \( X \) as in Case(Ben) may later come across \( Y \) as in Ben (Action[1-6]).

Hence, the rule expression is determined: Learner(?X), hasSimilarLearningPathTo(?X, “Ben”) \rightarrow hasCase(?X, “Ben”).
6. DISCUSSION

The development of semantic process mining tools entails three building blocks, *Annotated Event Logs, Ontologies, and Semantic Reasoning* that are all aimed at discovering, monitoring, and enhancing any given domain process. Indeed, any pattern or learning behavior can be discovered as a consequence or condition of such a method of conceptual analysis. It uses the frequency of the action sets within the process to generate rules and events relating to a task to automatically discover the process models, create ontologies, and support the semantic annotation and reasoning of the elements in the information systems. Moreover, ontology can be layered on top of these existing information assets to provide more enhancements to real-time processes in the same manner as process mining. Rather than displacement of prior knowledge, ontology provides benefits in discovery, flexible access, and information integration due to its inherent connectedness (inference), concept matching, and reasoning. This characteristic is the ability to match an idea as well as use the coherence and structure itself to inform and answer questions about relationships the process instances share among themselves within the process knowledge base. Thus, by specifying one concept (Learning_process) one knows that we are also referring to another concept (Learners), and thus Learners learn through engaging in a Learning_process. Technically, in the model developed in this paper, we describe the class Learner as a subclass of the LearningProcess. The necessary condition is: if something is a Learner, it is necessary for it to be a participant of the LearningProcess and necessary for it to have a sufficiently defined condition and relationship with another class, LearningActivity. The method allows the meaning of Learning objects/properties to be enhanced through the use of property characteristics and classification of discoverable entities. It uses the main function offered by the reasoner to help classify the entire model; it checks for consistency in the model, testing whether a specific class is a subclass of another class, and checks whether it is possible for a class to have any instances. This means that a class is said to be inconsistent if it does not have any instances. By performing such a test, i.e. classification, it becomes possible for the reasoner to correctly compute the inferred activity hierarchies (taxonomy).

In general, the semantic-based approach described in this paper has been used to develop semantic process mining plug-ins. In this paper, we used an OWL version 2 to model the learning process, which we implemented using Protégé 4 and Pellet 2. The work also used the Disco process management system (Rozinat & Gunther, 2012) to process the raw data. The expressivity of our OWL model was extended by adding SWRL rules to the implemented ontology based on the concrete syntax of the SWRL scheme.

7. CONCLUSION AND FUTURE WORK

In this paper, the process mining approach is used to discover, monitor, and improve the set of recurrent behaviors that can be found within the learning process. The technique is introduced in order to address the problem of determining the presence of different learning patterns within the learning process base. The semantic-based process mining and analysis method is perceived to be of great importance and significance in this area of research due to its ability to discover worthwhile process models by using the three main building blocks (i.e. annotated logs/models, ontology, and semantic reasoner) and its adoption of process description languages such as the OWL and SWRL. The method is especially useful in bridging the gap between the levels of learning for different users by providing them with the same learning opportunity through a system that adaptively supports the personalization of contents for learning based on data regarding the users’ learning behaviors.
Future work will focus on covering the whole spectrum of the approach presented in this paper to provide more general validation and better support for automated learning using domain-specific knowledge about any business process.

ACKNOWLEDGMENT
The author is most grateful to the tutors and colleagues at the School of Architecture Computing and Engineering, College of Arts Technologies and Innovation, University of East London, UK for their wonderful support throughout the time of this research work. The author would also like to thank the reviewers for their valuable comments.

REFERENCES


