The Application of a Semantic-Based Process Mining Framework on a Learning Process Domain

Kingsley Okoye[a*], Syed Islam[a], Usman Naeem[a], Mhd Saeed Sharif[a], Muhammad Awais Azam[b] and Armin Karami[a]

[a]School of Architecture Computing and Engineering; College of Arts, Technologies and Innovation, University of East London, Docklands Campus, 4-6 University Way, London, United Kingdom. E16 2RD.
{K.Okoye, Syed.Islam, U.Naeem, S.Sharif, A.Karami}@uel.ac.uk

[b]Faculty of Telecom and Information Engineering, University of Engineering and Technology, Taxila, Pakistan.
awais.azam@uettaxila.edu.pk

Abstract — **The process mining (PM) field combines techniques from *computational intelligence* which has been lately considered to encompass artificial intelligence (AI) or even the latter, augmented intelligence (AIs) systems, and the data mining (DM) to *process modelling* in order to analyze event logs. To this end, this paper presents a semantic-based process mining framework (SPMaAF) that exhibits high level of accuracy and conceptual reasoning capabilities particularly with its application in real world settings. The proposed framework proves useful towards the extraction, semantic preparation, and transformation of events log from any domain process into minable executable formats – with focus on supporting the further process of discovering, monitoring and improvement of the extracted processes through semantic-based analysis of the discovered models. Practically, the implementation of the proposed framework demonstrates the main contribution of this paper; as it presents a Semantic-Fuzzy mining approach that makes use of labels (i.e. concepts) within event logs about a domain process using a case study of the Learning Process. The paper provides a method which aims to allow for mining and improved analysis of the resulting process models through semantic – labelling (annotation), representation (ontology) and reasoning (reasoner). Consequently, the series of experimentations and semantically motivated algorithms shows that the proposed framework and its main application in real-world has the capacity of enhancing the PM results or outcomes from the syntactic to a much more abstraction levels.**

Keywords—process mining, process models, ontology, semantic annotation, reasoner, AI, event logs

I. INTRODUCTION

The need for novel approaches in design and integration of computational intelligence into everyday (e.g. business, learning) processes has sprout research investigations on how to exploit such tools for use in improving the ever increasing data about various organizations. In recent years, a common challenge with many of the business processes has been on how to develop intelligent systems that can provide platforms for exploring the additional, and most often, the monotonous tasks of managing the entire operational process and quality of information - by providing understandable and useful insights on the best possible ways to make the envisioned information explicable in reality. Such process-related analysis, often allied to *process mining*, means there is also need for tools and techniques that can extract valuable information from the event logs about the domain processes in view.

Most organizations have devoted a greater amount of resources towards modelling of their everyday processes or operations. Nevertheless, majority of the mapped processes appears to be non-operational, unfitting and/or presents itself in formats that are targeted towards understanding the process flows (workflows) rather than tackling the complexity of the business or domain processes in reality. In other words, a greater number of the resulting models and methods tends to support just *machine-readable* systems rather than *machine-understandable* systems at large. By machine-understandable systems we refer to methods that are developed not just for representing information in formats that can be easily understood by humans, but also for creating applications and/or systems that trails to inclusively process the information that they contain or supports. Thus, the mapped processes (models) are either semantically-labelled (semantic annotation) to ease the analysis process, or represented in a formal structure (ontology) that permits a computer (through semantic reasoning) to deduce or discover new facts/knowledge as a result of the defined relations or assertions within the ontology.

According to the work of [1] - [4] the efficiency or quality of the so-called organizations process can be improved if the process analysts and/or owners could perform an accurate analysis or exploration of the extracted event logs. For instance, the process of visualizing the relations or attributes the process instances share amongst themselves in a knowledge base. Of late, the PM field [3] or better still the latter - Process Querying [5] have turn out to be such valuable method that can be used to discover meaningful information from the recorded event logs. Besides, the process mining (PM) field combines techniques from computational intelligence and data mining to process modelling and analysis, as well as several other disciplines related to the business process management (BPM) [3] and AI to analyze the data. However, [6] and [2] notes that a common problem with majority of the PM methods in current literature is that they tend to rely on the tags or labels within the events log to discover the process models, and as consequence, appears to be vague and limited to some extent when confronted with unstructured data. The aforementioned challenge is as a result of lack of abstraction level of analysis (i.e. conceptual information)

that are most often required from a real world perspective. Apparently, given such kind of challenge, most of the PM techniques appears to not technically gain from the real-time knowledge (semantics) that describes the tags/labels in event logs of the process domains in question [6].

For this purpose, this paper introduces a framework to help bridge the *semantic gap* displayed by the traditional PM techniques - by providing a method that focus on structuring in a formal way the events logs and resultant models about the domains processes (i.e. sematic-based analysis of the datasets). The work utilizes a case study of the learning process to demonstrate the application of the approach. In short, the work addresses the identified challenges, i.e. - (i) the lack of PM tools/methods that exhibits semantic information extraction, retrieval and/or analysis, and (ii) process mining of events log and the derived models at a more abstraction level as opposed to the syntactic method of analysis displayed by the traditional PM techniques. Primarily, the purpose of the proposed semantic-based PM and model analysis framework (SPMaAF) is a way of introducing formal structures for the events logs used to perform process mining as well as improvement of the analysis of the results to a more abstract level that can be easily understood in a real-world settings. Predominantly, as it concerns the process of extraction, retrieval and conceptual analysis of the derived information (or mapped processes). Technically, the SPMaAF focuses on augmenting the value (machine-understandable system) of information derived from the resultant models by semantically-annotating the individual process entities (instances) with concepts which they represent in the real-world sceneries, and then links them to a well-defined domain ontology so as to allow for improved semantic analysis of the mined datasets. Thus, provide process models that are closer to the domain concepts.

To end with, the study illustrates the application of the proposed SPMaAF framework using the learning process domain case study - to show the importance as well as its comparison against other existing benchmark algorithms used for process mining. Practically, the SPMaAF considers the various phases of the PM techniques and implementation - ranging from the primary stage of collection and transforming the captured events logs to discovering of worthwhile models, then expounds the traditional PM to semantically preparation of the derived models for a more conceptual analysis and/or querying. In terms of abstraction level, the work shows that the SPMaAF is able to provide a formal means to analysing the data sets (i.e. events log or models) by allowing the real world (semantic) meaning of the individual process instances to be improved by making use of semantic technologies - such as the ontology web language (OWL) [7], semantic web rule language (SWRL) [8], description logics (DL) [9] etc. The method utilizes those tools in order to provide semantic information (i.e. through semantic reasoning or inference knowledge discovery) that are closer to human understanding, and yet, machine-understandable. In other words, this paper focuses on establishing the extent to which semantic modelling (ontological representations) and analysis is able to improve the results of the process mining through the *conceptualization* method (i.e. explicit specification of the domain concepts) [10][5] which, in turn, helps in identifying appropriate semantic knowledge or information about the process instances (elements).

The rest of the paper is structured as follows: in section (II), background information and related areas are discussed. Section (III) explains the SPMaAF including the main methods, components and motivation towards using the semantic-based approach to perform PM. In section (IV), the paper presents the different phases of implementing the SPMaAF framework, as well as sets of proposed algorithms that allows for ample implementation of the various phases. Section (V) shows how the work utilizes the case study of Learning Process to illustrate the SPMaAF approach and its main application in real world setting. In section (VI), we evaluate the outcome of the experimentations, particularly against other benchmark algorithms used for PM in order to weigh up the impact of the proposed framework, and then, discuss in section (VII) the influence and main achievements of the paper including the limitations and threats to validity, and consequently highlights worthwhile areas that could be investigated by extending the approach in this paper. Finally the work concludes and points out the direction for future works in section (VIII).

II. BACKGROUND INFORMATION AND RELATED AREAS

A. Patterns Discovery, Classifications and Model Analysis

Process mining (PM) belongs to collection of tools within the Business Intelligence (BI) [3] that utilizes data mining techniques to discover patterns or models from event logs and predict outcomes through further analysis of the discovered models. For example, [11] and [12] introduced various approaches for learning pattern control through a workflow management system (WFM) [3] but does not relate a devoted strategy for the process analysis such as the PM technique. On the other hand, [13] sparks the potential benefits of enhancing learning process models particularly within the context of Adaptive Educational Hypermedia Systems (AEHS) [14] by constructing a framework for exploiting learner models. Their method combines PM techniques with the concepts of learning patterns. Similarly, [15] and [16] have worked on approaches that applies the PM techniques in context of e-learning. The authors analyzed and points out tools that are used to perform PM tasks which qualifies better in support of e-learning processes. Whereas, [17] argues that a way of supporting learners within an e-learning setting is to adopt the combine approach of using the PM with concepts of the discovered learning patterns such as the one described in this paper.

According to [17], to perform the classification or assumptions of learner patterns, datasets need to be extracted from a learning system (i.e. the data sources). Moreover, in view of carrying out the process analysis - the extracted datasets needs to be prepared and transformed into formats that allows for the pattern recognition to follow. Consequently, prior to performing any other further analysis or use of the discovered information, an evaluation of the resulting learning patterns is carried out, and then the results are presented and interpreted as shown in Fig. 1.
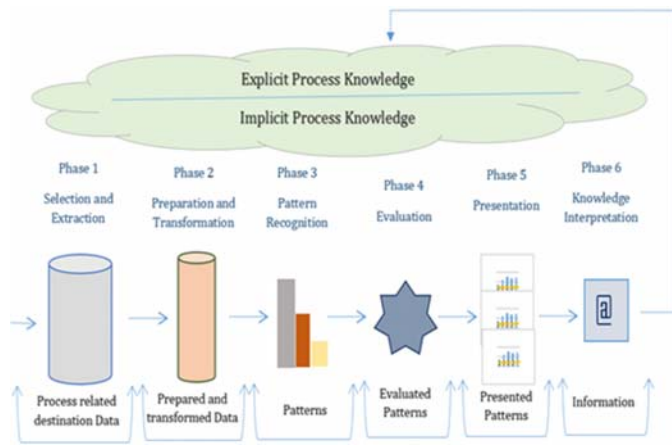
Fig. 1 Process Mining and Analysis Framework [17].

As gathered in Fig. 1 such type of process analysis, allied to the PM technique, improves explicit process knowledge. For instance, processes that are complemented by some implicit knowledge which may also be discovered inadvertently [18]. In short, PM could be adopted to analyse and improve the captured processes (e.g. the learning process) and models, or even more, used to recommend future patterns and/or behaviours.

Furthermore, the authors in [17] introduced the PM notion in e-learning settings to highlight the need of the approach in offering the capability of improving information values of such systems. The authors opines that PM in combination with learning concepts could be a promising tool for process automation and modelling. Their arguments are concerned with how learning processes can be improved through the use of process modelling and rule-based controls, as well as how process models can be generated in details taking into account the concept of the learning styles or patterns. Interestingly, the outcome of their research [17] reveals that the implementation of rule-based controlled methods (e.g. workflow-nets) [3] into systems that support process management remains an important field of further investigation. For that reason, and to address such gap in literature - this work has introduced the SPMaAF which is perceived as a semantic-based PM approach directed towards the discovery and enhancement of the resulting process models with its application on the learning process domain in order to show the influence and usefulness of the approach. We also explain in this Section II *(B)* to *(D)* - some of the intelligent functionalities that are attributed to the existing PM techniques and the need for the *advance* notion of the *semantic process mining* that has inspired the development of the SPMaAF framework and its implementation.

B. Concepts and Entities Relationship Prediction

A significant feature of Intelligent and Adaptive Educational Learning Systems (IAELS) [14] is the predictive aspect of the systems they support. Reference [19] notes that a pre-emptive educational model seeks to predict possible future activities in order to accomplish better student's training as well as overcome likely issues. Interestingly, the authors propose a Casual and Fuzzy Student Model (CFSM) which describes several user attributes to anticipate the sequencing of activities for the users

through the acquisition of domain knowledge about the learners. The main component derived here and pertinent in context of this paper, is the user attributes which are semantically described in form of concepts within an ontology. In essence, the described concepts sets casual relations amongst the individual elements, and as such represents a certainty on how an attribute utilizes the object/datatype properties and/or triggers another attribute.

C. Fuzzy Mining and Reasoning

Furthermore, instead of using a *neuro fuzzy* or *probabilistic reasoning* method, the authors in [19] makes use of a *fuzzy casual reasoning* approach to deal with qualitative observations or information. The main difference here, is that the approach organizes *ontologies* which are used in identification and definition of the meanings of the concepts, casual relations, fuzzy rule bases, universe of discourse and other relevant items using the OWL [7] declaration sentences such as *Classes, Datatype Properties, Functional Properties,* and *Process instances* etc. Moreover, ontology firmly sits at the heart of the proposed SPMaAF approach, and a description of how we have utilized and implemented the schema is explained in the later Sections (IV) and (V) of this paper.

D. Learning Process Automation

Automated learning process [17] means supplying simulations, contents and interactive maps in a unified and well-structured manner. Moreover, this is where semantic process modelling in combination with the PM techniques is capable of providing useful method for better identification and analysis of learning problems. Hence, the main reason why the proposed SPMaAF framework in this paper is grounded on *process mining* and *semantic process modelling*. The authors in [20] suggests that process automation irrespective of the settings in which they are being used in (be it within the educational settings or any other domain), should be offered as specialized computer-generated services and the analysis have to also integrate some form of business sub-processing tools such as the PM techniques.

E. The shift from Process Mining to Semantic Process Mining and Model Analysis

Process mining (PM) has proven to be one of the existing techniques that can be exploited to analyse event data about any domain process. One of such areas that the PM technique is currently being applied and is gaining attention in recent years is the Educational Process Mining (EPM) [21]. In fact, the EPM is a new domain area within the wider context of business process management (BPM) [3] that aims to apply process mining (PM) techniques to find out user patterns or models from the captured sets of educational data [15][22]. According to [21], EPM means the application of PM to raw educational data by considering the end to end processes rather than local patterns, as opposed to the Educational Data Mining (EDM) [1] which tends not to be process-centric and do not focus on event data [1][3] e.g. the rows (instances) and columns (variables) of a typical data file which does not have any meaning. Nonetheless, [21] notes that both the EDM and EPM apply specific algorithms to data in order to discover hidden patterns

and/or relationships. In fact, whichever tool one chooses to adopt, the key focus should be on achieving the purpose of adopting such techniques.

A number of researchers have directed their work towards the use and application of this new advanced aspect of PM within the educational settings [22][23][24][25][15][4]. According to [22] EPM emerges from the educational data mining (EDM) discipline, and the drive for its incentive is primarily to discover, analyse and improve the educational processes based on the hidden or unobserved informations within the events log recorded by the IT systems. References [4] and [22] are even more specific about the existing methods for extracting models within the EPM which are limited to some extent because the approaches depend on traditional PM techniques that are purely syntactic in nature (i.e. based on labels in event logs) to discover process models. Technically, in so doing, the developed systems tend not to gain from the real knowledge (semantics) that describes the processes as performed in reality, and as consequence, the actual semantics behind the event logs remains missing and sprouts the additional need for domain experts and even time to interpret them.

Therefore, in practice, PM poses some certain issue of semantics that limits its efficiency when handling large volume of events log from the complex educational systems [22][24][2]. The authors in [22] argue that the *semantic process mining* [24][2] approach appears to be a promising area that can be explored in order to resolve those issues of understanding the different patterns or trace heterogeneity, and as such, to extract streamlined models that fits or represents the actual processes as performed in reality. Moreover, [22] believes that *semantic annotation* of the captured datasets can be utilized to address the challenge of interpreting the processes. Thus, to benefit from the actual semantics behind those event logs tags or labels, the *semantic process mining* which enforces mining and analysis of processes at the conceptual levels have to be employed.

In view of that, this paper focuses on providing a method for semantic PM - by proposing the SPMaAF framework which is directed towards the discovery and semantic enhancement of the process models derived by applying the traditional PM techniques. The work shows through the *learning process domain* case study - How events logs from any given process domain is being extracted, semantically-prepared, and transformed into minable formats that supports the process of discovering, monitoring, and improvement of the extracted processes through semantic-based analysis of the discovered models.

Truly, the semantic-based approach proves to allow the analysis of the extracted event logs based on concepts rather than the event tags of the process. Besides, the semantic-based planning and analysis allows the meaning of the learning objects/models to be enhanced using *property characteristics* and *classification of discoverable entities*, to generate inference knowledge which are then used to determine useful patterns and improve analysis of the resulting models based on the domain concepts.

III. CONCEPTUAL OVERVIEW OF THE SEMANTIC PROCESS MINING AND ANALYSIS FRAMEWORK (SPMaAF)

In this section of the paper, we introduce the proposed SPMaAF and its main design components, and subsequently, show how the integration of the different phases of the framework and the resultant algorithms are utilized for ample implementation of the approach. The design of the SPMaAF framework and its main components is shown in Fig. 2.

In short, the proposed Semantic-based Process Mining and Analysis Framework (Fig. 2) constitutes the following processes as gathered in [26]

❖ *extraction of process models from event data logs:* the derived models 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:* helps 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 task and consistency checking to validate the resulting models 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 amongst themselves within the knowledge base.
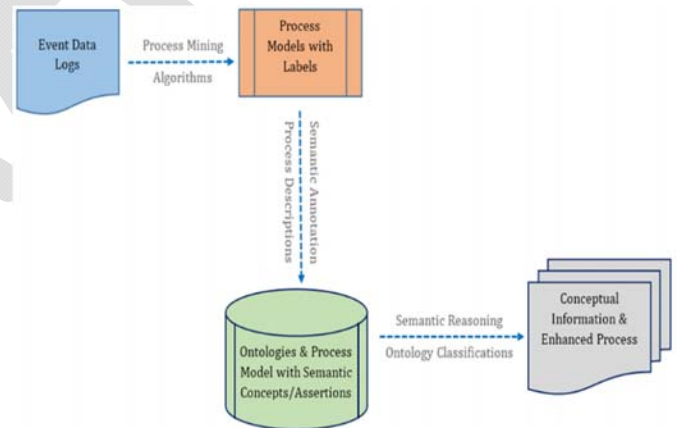

Fig 2. SPMaAF: Semantic Process Mining and Model Analysis Framework

In fact, the key step to application of the semantic-based process mining and model analysis approach is to focus on connecting the mining algorithm with 2 primary mechanisms:

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 [26].

Furthermore, the application of the semantic-based approach have received a major amount of interest as it concerns the PM field. On one hand, the SPMaAF focuses on using the semantic information (i.e. metadata) that are embodied in the events log of the domain processes to produce a new method for PM, or better still, supports improvement of the existing methods to assist humans in gaining a novel and more accurate results at a higher conceptual level that maps with the domain context as opposed to the traditional PM techniques (that tends to analyse data at the syntactic level). On the other hand, because of the semantic level of analysis, the outcome of the SPMaAF method appears to not just perceived as a machine-readable system but also a machine-understandable system. Whereby, the results of the process is not just in formats that the computer can read and analyse; but also the process analyst and/or owners or IT experts can easily understand the outcomes of the process. Such method of big data analysis is very important especially in current literature - where most often, the classical PM algorithms or tools fail to make use of the semantics (i.e. domain specific information) that are embedded in the events log of the processes which they represent.

To this end, this paper illustrates through the instantiation of the SPMaAF, sets of algorithms formalizations and the resulting semantic fuzzy mining approach - that through encoding of events log and models with rich semantic annotation, and the integration of semantic reasoning aptitudes that it is possible to specify useful domain semantic-based analysis capable of bridging the semantic gap conveyed by the traditional PM techniques [1][2][26]. Moreover, when considering such kind and level of semantic-based PM method, the work notes that meaningful information on how the different process elements (or components) fits and relies on each other within the process knowledge-base is made possible, and especially important towards the extraction of models that have the capability to produce newly, valuable, and conceptual information.

In short, the main difference between the SPMaAF (Fig. 2) and the traditional PM framework (Fig. 1) is that - whilst the framework in Fig. 1 tends to analyse the extracted events logs to derive some explicit and/or implicit information about the processes they support without considering the *semantic* aspects of the information that are contained in the events log. On the other hand, the SPMaAF (Fig. 2) focuses on semantical integration and extension of the method in Fig. 1 by considering the semantic gap that is missing with the traditional PM framework in terms of the extracted events log and the derived process models. In other words, whilst the traditional PM techniques analyses the event data logs at *syntactic levels* (i.e. labels or tags in the event logs), the SPMaAF pursues to extend and analyse the available events log and derived process model at a much higher *conceptual levels* (i.e. based on concepts defined within the model).

IV. IMPLEMENTATION OF THE SPMaAF: PROCESS MODELS DISCOVERY, SEMANTIC-BASED ANALYSIS AND ALGORITHMS FORMALIZATIONS..

In this section, we describe the different phases of implementing the SPMaAF, as well as, provide the sets of algorithms to instantiate each phase of the framework.

A. Different Phases of Implementing the SPMaAF & Design

The application of the SPMaAF is in 3 phases as detailed in [26]:

In Phase 1: the PM technique is applied in order to make available the process mappings for the learning process, and check its conformance with the event logs based on the Fuzzy Miner [27] as described in [4] and [6]. The main reason is that the resulting process map allows us to quickly, and interactively explore the processes into multiple directions and to show the individual activities *workflow*, and then provide platform for semantic annotation of the different process elements within the knowledge base. The proposed *Algorithm 1* in this Section IV (*B*) describes the procedures on how we have implemented this phase of the process.

In Phase 2: semantic modelling of the resulting process mappings in terms of the annotated terms is done. Thus, the semantic model represents the domain knowledge about the various learning activities and sequence workflows including the concepts as defined in an Ontology which sits at the core of the approach by using process description languages such as the OWL [7] and SWRL [8]. In addition, the process also makes use of the Reasoner i.e. Pellet - to infer the different instances and the ontological representation (taxonomy) of the learning process model in reality [6]. The proposed *Algorithms 2* in this Section IV (*C*) explains the steps we have taken to implement this stage of the framework.

In Phase 3: extraction and automated querying of the learning concepts is carried out. The work uses the Eclipse Java Runtime Environment to create the methods and interface for loading the Process Parameters (i.e. the ontology concepts). Essentially, we make use of the OWL Application Programming Interface (OWL API) to extract and load the inferred concepts from the defined ontology. Example of the inferred underlying concepts using the OWL API and the resulting ontologies is as shown in [23] and [24] and described in Section (V) of this paper. Reference [26] also shows the general architecture for implementing such a semantic-based process mining technique. The *Algorithm 3* describes the instantiation of the semantic reasoning phase.

B. Algorithm 1

Here, we describe the proposed *Algorithm 1* and how we have used the method to perform the process mining and model discovery (Phase 1) in order to discover useful process models from the events data log. The process proves useful towards generation and mapping of individual traces that makes up each of the process executions. Reference [6] describes how the proposed algorithm is implemented using PM tools such as the *Disco* based on the Fuzzy Miner Algorithm [27] to generate and map the process models from the event logs for conformance checking which allows one to analyze the individual cases (i.e. the classified traces and the sequence of activities executions).

In essence, *Algorithm 1* explains how the work discovers and produce process models and traces from any given event data log as follows:

Algorithm 1: Discovering Process Models and Individual Traces from Event Logs.

1: For all Recorded and Captured Event Data Log **EDL**
2: **Input:** **PM** – Process mining tool used to extract model, **M**
   e – Classifier for the event logs, **EDL** and traces, **T**
3: **Assign:** **Case_id(e)** i.e. the Case associated to event, **e** within the **EDL**
   Act_name(e) i.e. Activities associated to event, **e** within **EDL**
   Other_attributes e.g. Event ID, Timestamp, Resources, Roles etc. associated to event, **e** within the **EDL**
4: **Output:** Process maps for the discovered models, **M** & individuals traces, **T** classifications for the event log, **EDL**
5: **Procedure**: Produce Models, **M** from **EDL** for cross-validation to determine how well **M** reflects the performed activities in reality i.e TraceFitness, **TF** and for further analysis
6: **Begin**
7:   **For all** Event Data Log **EDL**
8:     **Extract** Process Maps, **M,** & Traces, **T** ← from Event Log **EDL**
9:     **while** no more process element is left **do**
10:       **Analyze** Model, **M** and Traces, **T** to determine tracesFitness, **TF**
11:       **If** $T \leftarrow$ Null **then**
12:         obtain the occurring $act\_name(e)$ sequence sets from Log **EDL**
13:       **Else If** $T \leftarrow 1$ **then**
14:         cross-validate resulting Trace, **T** from **EDL** with Model, **M**
15:       **If** trace, **T** exist **then**
16:         **For** each event Classifier, **e** output ← return as True_Positive, **TP** i.e fits the Model, **M**
17:         **Else If** trace, **T** does not exist **then**
             Return event Classifier, **e** output as True_Negative, **TN** i.e does not fit the Model, **M**
18: **Return**: Classification Results of the Process Mining approach and Process Mappings
19: **End** If statements
20: **End** while
21: **End** For

Algorithm 2: Developing Ontology from Process Models & Event Logs

1: For all defined models **M** and event log **EDL**
2: **Input:** **C** – different classes for all process domain
   R – relations between classes
   I – sets of instantiated process individuals
   A - sets of axioms which state facts
3: **Output:** Semantic annotated graphs & an ontology-driven search for process models and explorative analysis
4: **Procedure**: create semantic model with defined process descriptions and assertions
5: **Begin**
6:   **For all** process models **M** and event log **EDL**
7:     **Extract** Classes **C** ← from **M** and **EDL**
8:     **while** no more process element is left **do**
9:     **Analyze** Classes **C** to obtain formal structures
10:       **If** $C \leftarrow$ Null **then**
11:         obtain the occurring Process instances (**I**) from **M** and **EDL**
12:       **Else If** $C \leftarrow 1$ **then**
13:         create the Relations (**R**) between subjects and objects // i.e between classes **C** and individuals (**I**)
14:       **If** relations **R** exist **then**
15:           **For** each class **C** ← semantically analyse the extracted relationships (**R**) to state facts i.e Axioms (**A**)
16:       create the semantic schema by adding the extracted relationships and individuals to the ontology
17: **Return**: taxonomy
18: **End** If statements
19: **End** while
20: **End** For

Indeed, the *Algorithm 1* shows that:

▪ A typical process model, **M** consist of Traces, **T** (i.e. Cases)

▪ A Trace (Case), **T**, consist of events, **e,** such that each event relates to precisely one case.

▪ Events, **e**, within a Trace are ordered, most often in a sequential order

▪ Events for any process mining task must have atleast a Case identification ID (*Case_id*) and Activity Name (*Act_name*) attributes to allow for the process model discovery.

▪ Other additional information may be required for ample implementation of process mining e.g. Event ID, Timestamp, Resources, Cost, Roles, Places etc.

C. Algorithm 2

Semantic *representation* of the process (i.e. ontology) - *Phase 2* is vital for implementation of the SPMaAF framework. The ontological representation is focused on integrating information about the various entities that are embodied in the events data log (**EDL**), and the derived models (**M**). This is especially important when the aim is to discover previously unknown links and/or useful information amongst the process instances. Besides, the implementation of the procedures in Phase 3 (i.e. reasoning over the ontologies and classification of individual outputs) relies totally on the capability to represent the derived knowledge in a formal manner (taxonomies).

Thus, *Algorithm 2* shows how the work creates ontologies from the events log and models as described in [26] as follows:

Truly, the annotated logs/models and ontological representations are very useful in carrying out the semantic-based analysis of the process domains - because at this stage, the input data are presented in a formal structure (taxonomies) that can connect to concepts which are referenced in the underlying ontology [26]. In return, from *Algorithm 2*, the work shows that the ontology, Ont, is a quadruple, i.e.

$$Ont = (C, R, I, A)$$

which consists of different classes, **C**, and relations, **R**, between the classes [28] [29][26]. Perhaps, a relation, **R**, trails to connect a set of class(es) with either another class, or with a fixed literal and is capable of also describing the sub assumption hierarchy (e.g. taxonomy) that exists between the various classes and their relationships. In addition, the classes are instantiated with a set(s) of individuals, **I**, and can likewise contain a set(s) of axioms, **A**, which states fact (e.g. what is true and fitting within the model, or what is true and not fitting within the model). In other words, the ontologies can be defined as connected sets of taxonomies (i.e RDF + Axioms) or yet still, structuring in a formal way (Triple + Facts) where the subject includes the *Classes*, predicate represents the *Relations*, and the object includes the *Individuals*, in addition to the sets of axioms which states facts. The authors in [30] are even more specific about the importance of the ontological concepts (particularly the *classes*) in semantic representation of real-time processes. The authors note that "classes are the central item of the ontology" and further explain that a well-defined class may represent all types of procedures e.g. running tasks, data transmission, data flow control, activity workflows etc.

Without a doubt, the procedures in *Algorithm 2* shows the need for a well-defined *semantic annotation* process especially in developing such an ontology-based framework (SPMaAF)

that supports semantic-based PM and analysis - by producing in an automatic manner the underlying taxonomies and/or properties the process instances share within the knowledge base [29] [24]. Thus, the semantic annotation process is necessary for concrete implementation of the SPMaAF including the further steps of enhancing the model as explained in [24]. According to [29] a semantically annotated graph is described as follows:

$$SemAn :: N \cup E \rightarrow COnts$$

where: *SemAn* describes all kinds of annotations which can be input, output, meta-model annotation etc.

In general, [26] notes that the semantical planning of any ontology-based system such as the SPMaAF - must take into account all process *actions* (activities) within the ontology and must consist of some form of semantic annotation. Moreover, according to the definition in [29] if we Let *A* be the set of all process actions. A process action $a \in A$ is characterized by a set of input parameters $Ina \in P$, which is required for the execution of *a* and a set of output parameters $Outa \subseteq P$, which is provided by *a* after execution. All elements $a \in A$ are stored as a triple (*namea, Ina, Outa*) in a process library *libA*.

D. Algorithm 3

Accordingly, the last essential component in realizing the SPMaAF and its main application is the aptitude to perform *semantic-based reasoning* in order to infer (i.e. ontology classifications) and check for consistency in the semantic model. To perform the automatic inference and population of the individual ontologies, the system makes use of the *reasoner* to produce the necessary outputs and/or provide the results as executed during the information retrieval process (or search queries).

To this end, *Algorithm 3* shows how the work uses the reasoner (e.g. Pellet) in carrying out the ontological classifications and semantic reasoning following the definitions in [26]

Algorithm 3: Reasoning over Ontologies and Classification of Entities and Outputs

---

1: For all defined Ontology models **OntM**
2: **Input:** classifier e.g. Pellet Reasoner
3: **Output:** classified classes, process instances and attributes
4: **Procedure**: automatically generate process instance, their individual classes and Learning concepts
5: **Begin**
6:   **For all** defined object properties (**OP**) and datatype properties (**DP**) assertions in the model (**OntM**)
7:   **Run** reasoner
8:   **while** no more process and property description is left **do**
9:   **Input** the semantic search queries **SQ** or set parameter **P** to retrieve data from **OntM**
10:   **Execute** queries
11:     **If SQ** or **P** ← Null **then**
12:     re-input query or set the parameter concepts
13:     **Else If SQ** or **P** ← 1 **then**
14:     infer the necessary associations and provide resulting outputs
15: **Return**: classified Concepts
16: **End** If statements
17: **End** while
18: **End** For

Basically, *semantic reasoning* as described in *Algorithm 3* – associates meaning to the labels (i.e attributes) as well-defined within the knowledge-base by referencing the set of rules and/or assertions (e.g Object/DataType properties) that are defined in the ontologies in order to answer some common question (or queries) and perform a more abstraction analysis of the process elements.

In summary, methods for semantic PM and analysis such as the SPMaAF in this paper - focuses on information about resources hidden within a process knowledge-base, and how they are related [2][23][24][31]. The semantic-based analysis allows the meaning of the domain entities and/or object properties to be enhanced through the use of *property characteristics* and *classification of discoverable entities*, to permit analysis of the extracted event logs based on concepts rather than event tags or labels about the process. Interestingly, there are not too many algorithms that supports such semantic level of analysis currently in literature, and there are few existing applications that demonstrates the capabilities of the semantic approach [2][23][24][31]. References [23][24][32] shows how semantic annotations and reasoning can be used to provide more improved analysis/enhancement to process models and event logs through concept matching (i.e. ontology classifications). Specifically, [24] and [32] perform semantic modelling and integration of the resulting process maps with annotated terms and then present the domain knowledge for the activities (i.e attributes or concepts) as defined within the ontology using process description languages such as the OWL [7] and SWRL [8]. Indeed, reasoning on ontological knowledge plays an important role in semantic representation of processes [33] by allowing for extraction and conversion of explicit information into some implicit information. For example, the intersection or union of classes, description of relationships and concepts assertions. Thus, those main components (as described in Algorithms 1 – 3) namely; *annotated logs/models*, *ontology*, and *semantic reasoning* is the foundation upon which the SPMaAF is developed.

V. USE CASE SCENARIO AND IMPLEMENTATION OF THE SEMANTIC FUZZY MINING APPROACH

This section presents the practical implementation of the SPMaAF referred to as the Semantic fuzzy miner - including integration of the different stages of its development, and then, looks at the use case scenario of the *Learning Process* to show how the different components fit and is capable of analysing process models and event logs at a more abstract level.

A. Case Study of the Learning Process

This paper makes use of the case study example of the learning process (specifically the Research Process) to illustrate the application of the proposed method. The paper uses event logs about the research process to analyse and answer some common queries or real-time questions about the learning process. The case study as represented in [6] illustrates the workflow of a typical research process from the identification of the research area of topic to conclusion of the research investigation and award of degree. Apparently, in order to complete the research journey, it is expected that the researcher

must complete a set of step(s), i.e. activities or milestones - in order to find answers to the research questions [6]. To this end, this work shows by way of the SPMaAF framework and sets of semantically motivated algorithms that PM results can be presented at a much higher abstraction levels through semantic-based analysis of the learning process activities.

B. Implementation: Semantic Modelling and Analysis of the Research Learning Process.

To demonstrate the application of the SPMaAF framework and the proposed sets of algorithms – the work analyzes the learning process in order to determine patterns and/or attributes that differentiates some process instances from another. For example, to establish what *attributes* or *paths* particular learners follow or have in common, or what attributes distinguishes the *successful learners* from the *uncomplete* ones [26]. Henceforth, we focus not just on answering the identified questions, but also, to determine to what extent the application of the SPMaAF framework is able to perform semantic reasoning and classify in an effective manner - the different group of learner (i.e. cases) within the learning knowledge base. Practically, the paper focus on the use case distinction of the *successful* and *uncomplete* learners within the knowledge base.

Furthermore, the authors in [6] and [24] shows that the workflow for a research process consist of four milestones, i.e., establish context → learning stage → assessment stage → validation of learning outcome [24][6]. Thus, from defining the topic area -to- reviewing literatures -and- addressing the problem -then- defending the solution [6]. According to [26] the order in which the different milestones (which comprises of various individual learning activities) are carried out has the capability of determining the research outcome [24].

For example, in order to resolve the aforementioned learning questions in this section with regards to the *Successful* and *Uncomplete* learners – the work refers that a "Successful Learner" is a subclass of, amongst other NamedLearnerCategory, a Person that performs some LearningActivityConcepts, who has a universal object property restriction or relationship with the four milestones of the ResearchProcessClass (i.e. from defining the topic area -to- reviewing literatures -and- addressing the problem -then- defending the solution) as explained in [6][24][26] and defined below:

Successful Learner Class:

1: **ontology** ResearchProcess
2:   **concept** SuccessfulLearner
3:   hasCompleteMilestone **ofType** {DefineTopicArea, ReviewLiterature, AddressProblem, DefendSolution}
4:   isPerformerOf **some** LearningActivity
5:   is ofType Person
6:   hasInstance **members** {Mattew, Isaac}
7:   **axiom** DefinitionOfSuccessfulLearner

Clearly, as shown in the description above: the necessary condition is that - if something is a SuccessfulLearner, it is necessary for it to be a participant of the LearningActivity Concept class and necessary for it to have a kind of sufficiently defined condition and relationship with the ResearchProcess class and its attributes, namely: Define Topic Area, Review Literature, Address Problem and Defend Solution [6][24][26].

The same *object property restriction* [26] applies to the "Uncomplete Learner" class as described below:

Uncomplete Learner Class:

1: **ontology** ResearchProcess
2:   **concept** UncompleteLearner
3:   hasOnlyCompleteMilestone **ofType** {DefineTopicArea, Or ReviewLiterature, Or AddressProblem, Not DefendSolution}
4:   isPerformerOf **some** LearningActivity
5:   is ofType Person
6:   hasInstance **members** {Paul, Danny, Mark, Gregory, John}
7:   **axiom** DefinitionOfUncompleteLearner

Ideally, we observe from the descriptions above that the *object* or *data type properties/restrictions* are mainly employed when classifying or inferring previously unobserved class(es) or entities that satisfies the restrictions (assertions). For example, those individuals that could be classified as *Successful* or *Uncomplete* leaner who are required to meet the condition (restriction) of the specified class in order to become members or participants of the class. In fact, such kind of structured organisation (i.e object/data type restrictions) which integrates *semantic annotation* (labelling) and *reasoning* presents itself as a good way of formally representing the various entities (process instances) that can be found within any kind of process knowledge base or domain.

VI. EVALUATION OUTCOME AND EXPERIMENTAL RESULTS

The proposed framework in this paper shows that the semantic-based approach sparks methods that supports:

(i) the application of process mining techniques to domain processes, and

(ii) provision of real-world semantic knowledge and understanding about domain processes (e.g. case study of the learning process), which are useful towards the development of PM algorithms that are more intelligent with high level of effective conceptual reasoning capabilities.

The work in this paper proves that ontologies help in harmonizing the various process elements that are found within the process models and event logs. Moreover, the semantic annotations and reasoning helps to extract and add useful conceptual knowledge to the mining results.

Specifically, in [24] the work makes use of the event logs in [34] and [35] to weigh up the performance and to demonstrate; how the proposed Semantic-based Fuzzy miner is able to

perform a more accurate classification of the individual traces within the process base. This includes the capability to integrate the ontological concepts and perform semantic reasoning capable of discovering worthwhile models given the datasets (with *training set* and a *test set*) [34][35] for the cross-validation experiments.

Practically, in the experimentations in [24], the Semantic Fuzzy mining approach references a number of different OWL ontologies (e.g. the training model ontology, test set ontology, traceFitnessClassification ontology etc.) which were all created for the purpose of the experiment. For each ontology created for the experiment, all concepts in their turn were considered by the *reasoner* and are checked for consistency by referencing the process parameters. Based on behavioural characteristics of the provided datasets [34][35], a cross validation method was adopted in order to overcome the variability in the composition of the *training sets* and *test sets*. The traces were computed and recorded according to the *reasoner's* response, and the classification process was tested on the resulting individual outcomes by quantitatively assessing its performance with respect to correctly classified traces. For each result of the classification process, the replayable, i.e, true positives (TP) and non-replayable, i.e, true negatives (TN) traces were learned.

Indeed, the work observes that for every run set of parameters, the commission error, i.e. *false positives (FP)* and *false negatives (FN)* was null, thus equal to 0. This clearly means that the classifier did not make critical mistakes. For example, settings where a trace is deemed to be an instance of a specific class whilst it really is an instance of another class. At the same time, it is important to note also that the trace accuracy rate was very high i.e. for the correctly classified *true positives* (TP) and *true negatives* (TN), and were consistently observed for all the test sets. Further details about the outcome of the experiment with regards to the discovered process models and results of the classification process for the corresponding individual traces occurring in each *test* set are as reported in [24].

In short, the semantic-based fuzzy mining approach is aimed at improving the process analysis and system performance. Besides, the integration of the different ontologies, conceptual reference models, and reasoner; enables the definition of a more universal analysis question, and then focus on finding answers for those questions in an automated, thus, computerized manner. Even more, since the analysis is carried out at a higher level of abstraction (i.e. conceptualization), the results can be easily understood (i.e. closer to human comprehension) and the process of adding new concepts in the ontology, or yet still, changes or modifications to the attributes (i.e. labels or tags) do not necessarily entails or requires updating the analysis questions or queries.

Therefore, the SPMaAF framework, the resulting Semantic Fuzzy miner and its algorithms formalizations is a significant contribution to the state-of-the-art; where many existing PM techniques requires some form of reconstruction to bring the process analysis to a greater conceptual level or in many cases lacks the capability of identifying and/or making use of the semantics that can be found within the various process domain.

In Table 1, the work have carefully analyzed the aspects of the proposed framework and its main application (i.e, the Semantic Fuzzy Miner compared to other existing benchmark algorithm for semantic process mining). The Table 1 highlights the differences and similarities in properties of our approach with the only well-known, if not the only one, semantic process mining algorithm (the Semantic LTL Checker) [2] currently in literature as shown in Table 1. Thus, the Table 1 represents the Semantic-Fuzzy miner and its application properties evaluated against the Semantic LTL Checker [2].

	Semantic LTL Checker	**Semantic-Fuzzy Miner**
Data Input	Takes event logs concepts as input to parameters of Linear Temporal Logic (LTL) formulae	Takes process models derived from fuzzy mining of the event log as input to learn and reason about the domain process
Ontology	Ontologies are defined in WSML format	Ontologies are defined in OWL and SWRL format
Reasoning	Integrated using the WSML2Reasoner (W2RF)	Integrated using the Pellet Reasoner
Functionality	Uses LTL properties or formulae defined in LTL Template files (i.e. contains the specification of properties written in the special LTL language)	Uses process description properties (*CLASS_ASSERTIONS; OBJECT_PROPERTY_ASSERTIONS;* and *DATA_PROPERTY_ASSERTIONS)* defined using OWL and SWRL Language/schema.
GUI	There is option to select *concepts* for the parameter values	There is option to select *concepts* for the parameter values
Support	Supports *concepts* as a value (i.e. when a concept is selected, the algorithm will test whether the attribute is an *instance of* that concept, and concepts can only be specified for set attributes).	Supports *concepts* as a value (i.e. when a concept is selected, the algorithm will test whether the attribute is an instance of that concept, and concepts can only be specified for set attributes).

Table 1. The Semantic-Fuzzy miner and its application properties evaluated against existing benchmark algorithm.

Firstly, the Semantic Fuzzy mining approach based on the distinctive properties described in Table 1 proves to be more robust and effective than the traditional PM techniques. This is due to the fact that the method also takes the semantics perspectives of the event logs and process models into account. Moreover, as opposed to the existing semantic LTL checker [2]

which only considers and takes event logs concept as input to parameters of a Linear Temporal Logic (LTL) formula to analyse process, the semantic fuzzy mining approach also takes the process models as input. Besides, because those models are automatically created from the actual event logs of the process domains, the system tends not to unnecessarily lose or leave out important information or missing data.

Secondly, even though both approaches make use of ontologies, a major difference between the existing semantic LTL checker algorithm and the Semantic-Fuzzy miner is the fact that ontologies are defined in Web Service Modelling Language (WSML) [36] format with the semantic LTL checker, whereas on the other hand, ontologies are defined using the OWL [7] and SWRL [8] format in this paper. In essence, whilst there are limitations with WSML ontologies with respect to the exchange of syntax over the web, OWL ontologies aim to bring the expressive and reasoning power of description logic (DL) [9] to the semantic web. Moreover, it is the state of the art logical layer upon which semantic architectures are currently built in literature [37]. In short, OWL ontologies allows one to specify far more about the properties and classes which are defined within a process domain and/or knowledge base. In other words, the OWL ontology as utilized in this paper is developed not just for representing information in formats that can be easily understood by humans, but also for building applications that tends to inclusively process the information that they contain or supports. Thus, supports *machine-understandable* systems rather than just *machine-readable* systems.

Thirdly, from a *reasoning* or *classifier* point of view; whilst the semantic LTL checker makes use of the WSML2Reasoner [38][36] to perform a more complex inferences that are beyond subsumption reasoning by only benefiting from the inclusion of semantic annotations, the Semantic Fuzzy mining approach is integrated with Pellet reasoner [39] which typically in addition to semantic annotations has been proved to incorporate optimizations for nominals, conjunctive rules and query answering, and incremental reasoning capabilities that supports process descriptions and logic (i.e. class assertions and object/data property assertions).

Nonetheless, the semantic LTL checker and the Semantic Fuzzy miner both have option to select *concepts* for the parameter values, and indeed, supports concepts as a value, i.e, when a concept is selected, the algorithm will test whether an attribute is an instance of that concept (i.e class), and concepts can only be specified for set attributes. For example, with the Semantic-Fuzzy miner application, one can test whether: For all **Persons** (i.e. Performer instances) does always (**condition check?** - exist four milestones?) implies eventually (**class description**: Successful Learner). In other words, does any named **Person P**: hasCompleteMilestones **A** and **B** and **C** and **D**, where: **A** = DefineTopicArea, **B** = ReviewLiterature, **C** = AddressProblem, and **D** = DefendSolution - represents and points to the concepts within the domain ontology.

VII. DISCUSSION

The method in this paper focus on ascertaining by the series of validation experiments - how the outcome of PM techniques can be enriched through semantic representation and analysis of the derived models. As a result, the Semantic-based Fuzzy miner was developed, in addition to the various sets of semantically motivated algorithms and the design framework – SPMaAF proposed in this paper.

In short, the paper shows that the three main tools, namely: annotated logs/model, ontologies, and reasoner - constitute the building block upon which semantic process mining technologies such as the SPMaAF framework are constructed. Thus, while the *semantic-based annotation* is dedicated to providing *metadata* about the different process elements within the knowledge-base, *ontologies* are used to bind together (integrate) the various classes (concepts) in the model. Without a doubt, the population and/or classification of the underlying relationships that exist amongst the process elements is performed using the *reasoner*.

Indeed, such an approach (e.g. SPMaAF) does not only support the implementation of some of the process description languages, e.g. SWRL, OWL, DL etc., but it also allows the re-use of any type of ontology by another. In addition, the process tends to minimalize the number of human errors which are, every now and again, present particularly during the extraction, interpretation and/or analysis of the multiple entities/concepts that can be found within the domain processes in question.

Therefore, the unabridged notion of the proposed Semantic Fuzzy mining approach, design framework – SPMaAF, sets of algorithms *1, 2 & 3*, and the experimental results proves that semantic concepts (i.e. annotation, ontology, and reasoning) can be layered on top of existing information asset (i.e. process models, event data logs etc.) to provide a much more easy and accurate way of analysing the real-time processes capable of providing real world insights and/or answers that can be more easily grasp by the process analysts or owners.

On the other hand, whilst the research believes and prove that such method is practically suitable for mining processes at a more conceptual level, there could also exist a number of limitations and threats to validity. Thus, whilst the paper have introduced a design framework, and sets of descriptive algorithms and method to resolve the sets of identified problems and question that motivates the research investigations, there could be potentially many ways to address those problem, or even, bigger areas that have not been yet addressed. Owing to the fact that the semantic process mining is a new area within the process mining field, and there are not too many tools or algorithms that support such an approach currently in literature. Therefore, we assume that this work is only an incentive of more robust and intensive research within the context of the semantic-based process mining (PM).

Also, the procedures and method for semantically analysing the domain processes which this paper introduces is one of the main important contributions of this work. Besides, the technique has the capacity to extract conceptual knowledge (information) from the events log and process models. However, the correlation and integration of the key core elements/components (i.e. annotated logs/models, ontologies, and reasoner) that underlies the proposed method assumes that the work has presented an approach which can possibly be re-introduced or extended in other resourceful ways. Perhaps, a worthwhile extension of the approach may include more

sophisticated increment of the ontology schemas and reasoning capabilities that has already been well-defined in this paper. This may include the development of authoring tools capable of augmenting the stated achievements of this paper, or yet still, the mentioned process domains and/or other operational processes in the real world settings in general.

VIII. CONCLUSION

This paper introduced the SPMaAF to support the development of semantic-based PM techniques and algorithms that exhibits a higher level of semantic reasoning and/or capabilities as opposed to the syntactic nature of the traditional PM techniques. The instantiation of each phase of the framework was presented in terms of the proposed *Algorithms 1, 2, & 3*. The evaluation results shows that the SPMaAF approach can be perceived as an ontology-based system that is able to perform query answering and/or information retrieval in a much effective and efficient manner in comparison to existing standard logical procedures - particularly in terms of the other state-of-the-art approaches used for semantic process mining.

Future work will be to implement the proposed framework to analyse data from other domain areas of interest. This will allow for further validation and will generalise the findings of the research and case study presented in this paper. Another extension will be to complement the approach with a platform for completely automatic discovering and integration of the semantic information or knowledge.

REFERENCES

[1] D. Dou, H. Wang, and H. Liu, "Semantic Data Mining: A Survey of Ontology-based Approaches" In: 9th IEEE International Conference on Semantic Computing, California, USA, pp. 244 – 251, 2015.

[2] A.K. A. deMedeiros, W. M. P. Van der Aalst, and C. Pedrinaci, "Semantic Process Mining Tools: Core Building Blocks. In: ECIS, Galway, Ireland, pp. 1953-1964, June 2008.

[3] W. M. P. Van der Aalst, "Process Mining: Data Science in Action" 2nd [ed]. Springer-Verlag Berlin Heildelberg, 2016.

[4] K. Okoye, A. R. H. Tawil, U. Naeem, S. Islam, and E. Lamine, "Semantic-based Model Analysis towards Enhancing Information Values of Process Mining: Case Study of Learning Process Domain" In: A. Abraham et al, Eds. Advances in Intelligent Systems and Computing book series (AISC, volume 614) Springer International Publishing AG., pp. 622-633, 2017.

[5] A. Polyvyanyy, C. Ouyang, A. Barros, and W. M. P. Van der Aalst, "Process querying: Enabling business intelligence through query-based process analytics" Decision Support System, vol. 100(1), pp. 41-56, 2017.

[6] K. Okoye, A. R. H. Tawil, U. Naeem, S. Islam, and E. Lamine, "Using semantic-based approach to manage perspectives of process mining: Application on improving learning process domain data" 2016 IEEE Int. Conf. on Big Data (BigData), Washington, D.C, pp. 3529-3538, 2016.

[7] W3C. "OWL Web Ontology Language" [Internet]. 2004. Available from: http://www.w3.org/TR/owl-ref/ [Accessed: September, 2017].

[8] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML" W3C Member Submission [Internet]. 2004. Available from: http://www.w3.org/Submission/SWRL/ [Accessed: September, 2017]

[9] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, "Description Logic Handbook: theory, implementation, and applications" 1st [ed]. New York, USA, Cambridge Uni. Press, 2003.

[10] N. Balcan, A. Blum, and Y. Mansour, "Exploiting Ontology Structures and Unlabeled Data for Learning" In: 30th Int. Conference on Machine Learning, Atlanta Georgia, USA, pp. 1112-1120, 2013.

[11] M. Cesarini, M. Monga, and R. Tedesco, "Carrying on the e-learning process with a workflow management engine" Proc. of ACM Symposium on Applied Computing, Nicosia Cyprus, pp. 940-945, 2004.

[12] R. Perez-Rodriguez, M. Caeiro-Rodriguez, and L. Anido-Rifon, "Supporting PoEML educational processes in Moodle: A middleware approach" In: SPDECER, Universidad Pontificia de Salamanca, 2008.

[13] L. Nguyen and D. Phung, "Learner model in adaptive learning" World Academy of Science, Engineering and Tech., vol. 45, pp. 395-400, 2008.

[14] A. Peña-Ayala, "Intelligent and Adaptive Educational-Learning Systems: Achievements and Trends" 1st [ed]. Springer Berlin, 2013.

[15] N. Trčka, M. Pechenizkiy, and W. M. P. van der Aalst, "Process Mining from Educational Data" In: C. Romero et al, Eds. Handbook of Educational DM. Chapman & Hall/CRC Data Mining & Knowledge Discovery Series, CRC Press, Boca Raton, Florida, pp. 123-142, 2010.

[16] M. Pechenizkiy, N. Trcka, E. Vasilyeva, W. M. P. van der Aalst, and P. de Bra, "Process Mining Online Assessment Data. Proceedings of EDM, Cordoba, Spain, pp. 279-288, 2009.

[17] M. Holzhüter, D. Frosch-Wilke, and U. Klein, "Exploiting Learner Models Using Data Mining for E-Learning: A Rule Based Approach" In: A. Peña-Ayala, Ed. IAELS: Achievements and Trends, Heidelberg: Springer Berlin Heidelberg, pp. 77-105, 2013.

[18] H. L. Grob, F. Bensberg, and A. Coners, "Regelbasierte steuerung von geschaftsprozessen-konzeption eines ansatzes auf basis des process mining" In: Die Wirtschaftsinformatik, Heidelberg, 2008.

[19] A. Peña-Ayala and H. Sossa, "Proactive Sequencing Based on a Causal and Fuzzy Student Model" In: A. Peña-Ayala, Ed. IAELS: Achievements and Trends, Springer Berlin Heidelberg, pp. 49-76, 2013.

[20] D. Dżega and W. Pietruszkiewicz, "Intelligent Decision-Making Support within the E-Learning Process" In: A. Peña-Ayala, Ed. IAELS: Achievements and Trends. Springer Berlin, pp. 497-521, 2013.

[21] A. Bogarín, R. Cerezo, and C. Romero, "A Survey on Educational Process Mining" Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery (WIRES), Wiley Periodicals, pp. e1230, 2017.

[22] A. H. Cairns, J. A. Ondo, B. Gueni, M. Fhima, M. Schwarcfeld, C. Joubert, and N. Khelifa, "Using Semantic Lifting for Improving Educational Process Models Discovery and Analysis" SIMPDA, vol.1293 of CEUR Workshop Proc., CEUR-WS.org, pp. 150-161, 2014.

[23] K. Okoye, A.R.H. Tawil, U. Naeem, and E. Lamine, "Discovery and Enhancement of Learning Model Analysis through Semantic Process Mining*" International Journal of Computer Information Systems and Industrial Management Applications, vol. 8(2016), pp. 093-114, 2016.

[24] K. Okoye, U. Naeem, and S. Islam, "Semantic Fuzzy Mining: Enhancement of process models and event logs analysis from Syntactic to Conceptual Level" Int. Journal of Hybrid Intelligent Systems, IOS Press, vol. 14 (1-2), pp. 67–98, 2017.

[25] A. Bogarín, C Romero, R. Cerezo, and M. Sánchez-Santillán, "Clustering for improving educational process mining" ACM, NY, pp. 11 – 15, 2014.

[26] K. Okoye, S. Islam and U. Naeem. "Ontology: Core Process Mining and Querying Enabling Tool". In: C. Thomas, Editor. Ontology in Information Science, InTechOpen Publishers. Chapter 7, pp. 145-168, 2018.

[27] A. Rozinat, and C. Gunther, "Disco User Guide - Process Mining for Professionals" Fluxicon.com., Eindhoven, The Netherlands, 2012.

[28] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing" Int. J. of Human-Computer Studies. Vol. 43(5-6), pp. 907-928, 1995.

[29] F. Lautenbacher, B. Bauer, and S. Forg, "Process Mining for Semantic Business Process Modeling" In: 13th Enterprise Distributed Object Computing Conference Workshops; Auckland. pp. 45-53, 2009.

[30] O.O. Petrenko and A.I. Petrenko. A Model-driven Ontology Approach for Developing Service System Applications. Journal Computer Science Application Information Technology, 2017; 2(4): 1-7.

[31] W. Jareevongpiboon and P. Janecek, "Ontological approach to enhance results of business process mining and analysis" Journal of Business Process Management, vol. 19(3), pp. 459 – 476, 2013.

[32] K. Okoye, A. R. H. Tawil, U. Naeem, and E. Lamine, "Semantic Reasoning Method Towards Ontological Model for Automated Learning Analysis" In: Pillay, N., Engelbrecht, A., Abraham, A., du Plessis, M., Snášel, V., Muda, A., Eds. Advances in Intelligent Systems and Computing, NaBIC Conf. 2015, Springer Switzerland, pp. 49-60, 2016.

[33] D. Calvanese, T. E. Kalayci, M. Montali, and S. Tinella, "Ontology-Based Data Access for Extracting Event Logs from Legacy Data: The onprom Tool and Methodology" In: Abramowicz W. Eds, Business Information Systems. BIS 2017. Lecture Notes in Business Information Processing, Springer, Cham, vol. 288. pp. 220-236, 2017.

[34] J. Carmona, M. de Leoni, B. Depair and T. Jouck, IEEE CIS Task Force on Process Mining Process Discovery Contest @ BPM 2016. [1st Edition]

(2016) Available at: http://www.win.tue.nl/ieeetfpm/doku.php?id=shared:edition_2016

[35] K. Okoye, U. Naeem, S. Islam, A. R. H. Tawil and E. Lamine, "Process Models Discovery and Traces Classification: a Fuzzy-BPMN Mining Approach". Journal of International Technology and Information Management (JITIM), IIMA 2018, Vol. 26(4) Article 1, pp. 1-50, 2018.

[36] J. de Bruijn, H. Lausen, A. Polleres, and D. Fensel, "The Web Service Modeling Language WSML: An Overview". In: Y. Sure & J. Domingue, eds. The Semantic Web: Research and Applications. ESWC 2006. LNCS, vol 4011. Springer, Berlin, Heidelberg, pp. 590-604, 2006.

[37] F. Lisi, "Building Rules on Top of Ontologies for the Semantic Web with Inductive Logic Programming" Theory and Practice of Logic Programming, vol. 8(3), pp. 271–300, 2008.

[38] B. Bishop, F. Fischer, U. Keller, N. Steinmetz, C. G. Fuchs, and M. Pressnig, "WSML Reasoner", MA: IRIS Reasoner, Boston, USA, 1999.

[39] E. Sirin and B. Parsia, "Pellet: An owl DL reasoner" International Workshop on Description Logics (DL2004), Whistler, British Columbia, Canada, vol. 104, CEUR-WS.org, 2004.