International Journal of Hybrid Intelligent Systems 14 (2017) 67–98
DOI 10.3233/HIS-170243
IOS Press

Semantic fuzzy mining: Enhancement of process models and event logs analysis from syntactic to conceptual level

Kingsley Okoye*, Usman Naeem and Syed Islam
School of Architecture Computing and Engineering, University of East London, London, UK

Abstract. Semantic-based process mining is a useful technique towards improving information values of process models and analysis by means of *conceptualization*. The conceptual system of analysis allows the meaning of process elements to be enhanced through the use of property characteristics and classification of discoverable entities, to generate inference knowledge that can be used to determine useful patterns and predict future outcomes. The work in this paper presents a Semantic-Fuzzy mining approach that makes use of labels within event log about real-time process to provide a method which allows for mining and improved process analysis of the resulting process models through semantic – *annotation, representation* and *reasoning*. Qualitatively, the study shows by using a case study of *Learning Process* – how data from various process domains can be extracted, semantically prepared, and transformed into mining executable formats to support the discovery, monitoring and enhancement of real-time domain processes through further semantic analysis of the discovered models. Also, the paper quantitatively assess the level of accuracy of the classification results to predict behaviours of unobserved instances within the process knowledge-base by determing which traces are fitting or not fitting the discovered model by using a *training set* and *test log* for the cross-validation experiment. Accordingly, the work looks at the sophistication of the proposed semantic-based approach and the discovered models, validation of the classification results and their influence compared to other existing benchmark techniques and algorithms for process mining. The experimental results and data validation ends with the supposition that a system which is formally encoded with semantic labelling (annotation), semantic representation (ontology) and semantic reasoning (reasoner) has the capability to lift process mining analysis and outcomes from the syntactic level to a much more conceptual level, resulting in a mining approach that is able to induce new knowledge based on previously unobserved behaviours and a more intuitive and easy way to envisage the relationships between the process instances found within the available event data logs and the discovered process models.

Keywords: Process mining, process modelling, semantics, annotation, ontology, fuzzy models, event logs

1. Introduction

Many organizations have invested in projects to model their business processes. However, most of the derived process models are often incompatible, non-operational, or represents a form of reality that is pointed towards comprehensibility rather than covering all of the complexities of the actual business process. Over the decades, researches has shown that a better way of getting a closer look at organisations business process is to look into the event data logs readily available in its process information systems (Dou et al. [1], Van der Aalst [2], Carmona et al. [3], Okoye et al. [4], de Medeiros et al. [5]). Indeed, an accurate analysis of the event logs can give vital and valuable knowledge regarding the quality of the supported business processes and the existing information knowledge-base. Currently, a common challenge with many organisations processes has been on how to create effective tools and techniques capable of provid-

*Corresponding author: Kingsley Okoye, School of Architecture Computing and Engineering, University of East London, London, UK. E-mail: K.Okoye@uel.ac.uk.

ing platforms for exploring the additional, and most often, the monotonous tasks of managing the entire business process and ensuring quality of information derived from the available datasets presents in its process knowledge base as well as how to make the learned insights explicable in reality. One of the common discussions has been on how to create systems capable of providing effective platforms for information extraction by stemming understandable patterns or model behaviours as well as making the discovered patterns and models explicable. The increasingly volumes of available data in many organisations and the society in a wider scale means there is growing need for systems that can handle such big data, but can also get valuable information out of it for the company's business advantage and/or organizational use.

Following such developments, the *process mining* notion that was first proposed by Van der Aalst [2] has become a valuable technique used to discover meaningful information from event data about any domain process. According to Van der Aalst [2] and Carmona et al. [3], the field of process mining combines techniques from computational intelligence and data mining to process modelling and analysis, as well as several other disciplines to analyze large datasets from a process perspective or point of view. In essence, process mining techniques trails to link data analysis with process management. Nonetheless, a common problem with process mining has been the technical focus of the event logs. Most of the existing techniques depend on tags in event logs information about the captured processes to discover models. The level of abstraction of the models corresponds with the level of abstraction of the log, and therefore, to a certain extent are limited because they lack the abstraction level required from real world perspectives. This means that many process mining algorithms lack the ability to identify and make use of semantics across the different process domains. Majority of the mining techniques in literature are purely syntactic in nature, and to this effect are somewhat vague when confronted with unstructured data. Besides, those techniques do not technically gain from the real knowledge (semantics) that describe the tags in event log of the domain processes (de Medeiros and Van der Aalst [6]).

Following the identified challenges with the process mining techniques and analysis. The work in this paper in turn supports and extends our previous works in Okoye et al. [7,8] by presenting a Semantic-Fuzzy mining approach targeting the semantic challenges in all stages of process mining. This entails from the preliminary steps of gathering and transforming the raw event data to process models discovery, to semantically preparing and representation of the extracted models for further analysis at a much more conceptual level capable of describing the various process elements and improve quality of the system performance as well as accuracy of the classification results. In practice, this paper uses a case study of a learning process and data about a real-time business process to do the following:

- Extract data from process domains to show how we semantically synchronize the event log formats for various process domain data;
- Semantically prepare the data through an ontology driven search for explorative analysis of the process activities and executions;
- Transform the data into mining executable formats to support the discovery of valuable process models through our technique for annotating unlabelled learning activity sequences using ontology schema/vocabularies;
- Monitor and enhance the domain processes through further semantic analysis of the discovered models;
- Provide techniques for accurate classification of unseen process instances (traces) within the process models/knowledge-base, and useful strategies towards development of process mining algorithms that are more intelligent, predictive and robotically adaptive;
- Importance of semantics process mining to augment information value of data about domain processes: case study of learning process.

In summary, this study focus is on ascertaining by a series of validation experiments: how the outcome of the process mining techniques and individual trace classifications can be improved through further semantic analysis and representations of the deployed models.

Specifically, the work present the 3 key aspects that stems as a result of implementing the approach proposed in this paper and its main contributions as follows:

- Firstly, we use the fundamental concepts of semantic-based process mining to provide formal structures on how to perform and present process mining results in a more intuitive and easy way, in order to abstract key information that are used to envisage the relationships between process instances found within the event data logs and the discovered process models. The drive for such a

semantic-based approach is by pointing to references in an ontology and application of semantic reasoning, it becomes easy to refer to a particular trace or events within the discovered model. In principle, we provide a method towards finding useful structures for the different process elements or entities, and an easy way to determine the relationships they share within the process knowledge-base;

- Secondly, we provide a process mining technique that is able to induce new knowledge based on previously unobserved behaviours: which can be utilized by the process owners, process analysts or IT experts to perform useful information retrieval and query answering in a more efficient, yet effective way compared to other standard logical procedures due to the level of accuracy of the trace classifications to predict behaviours of unobserved instances within the process knowledge-base. Principally, the work in this paper employs a semantic-based process mining approach that shows a very high level of accuracy and as such do not make critical mistakes due to formal integration of semantic knowledge to the system. Indeed, the proposed approach can be exploited for predicting or suggesting missing information about process elements especially when completing large ontology-based systems as a result of the increase in predictive accuracy of the classifications and error-free analysis of the process at a more conceptual level;
- Thirdly, the work in this paper propose a Semantic-based Fuzzy mining approach to realise the study contributions. We propose a design framework and methods that highly influence and support the development of process mining algorithms that exhibits a high level of semantic reasoning and capabilities.

In turn, the work in this paper looks at what extent and how effective semantic reasoning can be used to lift process mining results and analysis from the syntactic level to a more conceptual level by semantically representing and analysing the resulting process models. The semantic analysis makes use of the metadata (semantics) described in the event log about the domain process, and links them to concepts in an ontology to extract and perform a more conceptual analysis of the data sets by means of the semantic reasoning. *Semantic Reasoning* is supported due to the formal definition of ontological concepts and expression of relationships that exist between the event logs. Thus, the method uses the semantics of the sets of activities within the process to generate rules and events relating to task, to automatically discover hidden traces (i.e., unobserved behaviours) and enhance the process models as well as the resulting ontologies through semantic annotation of the elements found within the process base. We introduce the approach as means towards discovering and enrichment of the sets of recurrent behaviours or patterns that can be found within any given process domain following the works we have done in [7,8] to determine attributes the process elements share amongst themselves, or that distinguishes a particular set of entities (process instance) from another. The technique is developed in order to address the problem of determining the presence of different patterns (traces) within the domain processes and derived models. The unabridged notion of the proposed semantic fuzzy mining approach and experimental results is aimed to prove 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 more conceptual analysis of the real time processes capable of providing real world insights and answers that can be more easily grasp by process owners, process analyst, system developers, software vendors etc. Accordingly, we qualitatively validate this notion using a case study of the learning process, and in turn, assess quantitatively the reliability and accuracy of the classification results of the approach using real time data from the IEEE Task Force on Process Mining [3]. The drive for such our approach is that by pointing to *references* (object property assertions and annotations) in an *ontology* and application of *semantic reasoning*, it becomes easy to classify and/or refer to individual cases or events within the available datasets and discovered process models.

The rest of the paper is structured as follows: in Section 2, the work discuss background information and the appropriate related works. Section 3 explains the design framework for our proposed approach including the various components, methodology and motivation towards using the semantic-based approach to perform process mining. In Section 4, the study show how we represent and analyse the individual process models and traces realized as a result of the classification task carried out in this paper. In addition, we show how we use the case study of learning process to illustrate our approach. Also, we describe the implementation of the approach to show the usefulness of the proposed semantic fuzzy miner algorithm. Section 5 describes

the experimentations we carried out and how we expound the application of the approach from fuzzy to semantic fuzzy mining. In Section 6, we evaluate and analyse in a qualitative and yet quantitative manner the outcomes of our experiments against other benchmark algorithms, to weigh up the proposed semantic fuzzy mining approach and its outcomes. Finally, we discuss and interpret the impact of the proposed approach and conclusions, and point out directions for future works in Section 7.

2. Background information and state of the art

Most of the existing techniques for analysing large knowledge bases or better still Big Data focus on constructing algorithms to help those knowledge bases or unprecedented growing data automatically or semi-automatically extend. According to Miani and Hruschka Junior [9] vast number of such systems built for managing the large knowledge bases continuously grow, and most often, they do not contain all facts for each process instance or elements thereby resulting in missing value datasets. Consequently, a well-designed information processing, retrieval or mining system should present results and the discovered patterns in a formal and structured format qua being interpreted as domain knowledge and to further enhance the existing process knowledge base [1].

According to Hicheur-Cairns et al. [10] one of the challenges with such process discovery and information retrieval and analysis techniques when applied to any domain – is that they rely exclusively on the syntax of labels in the databases, and are very sensitive to data heterogeneity, label name variation and frequent changes. As a result, majority of the process models are discovered without some kind of hierarchy or structuring. To address such problem, the authors show how by linking labels in event logs to the underlying semantics that describes the discovered models, one can bring processes discovery to the conceptual level in order to provide a more accurate mining and compact analysis of the processes at different levels of abstraction. Moreover, by extracting process models annotated with semantic information, the authors [10] propose a semi-automatic procedure used to associate semantics to training labels. They used the Ontology Abstract Filter plug-in in ProM [11] as input to a semantically annotated event log to produce as output an event log where the names of tasks, i.e. trainings labels, are replaced by the names of a set of chosen concepts. The produced log is then exported as Semantically Annotated Mining eXtensible Markup Language (SA-MXML) [5] file format, and subsequently perform a control-flow mining using the Heuristic Miner algorithm proposed by Weijters et al. [12] to extract the process models based on the concepts that has been derived.

Indeed, some of the existing techniques for semantic process mining and analysis focuses on information about resources hidden within a process knowledge-base, and how they are related (de Medeiros et al. [5], de Medeiros and Van der Aalst [6], Okoye et al. [7], Jareevongpiboon and Janecek [13]). In the work in [7] we describe how the semantic-based analysis allows the meaning of the domain entities and 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 the event tags or labels about the process. Even though, there are not too many algorithms that supports such semantic analysis and there are few existing applications that demonstrates the capabilities of the semantic-based technique [5–7,13]. Also, in [7], we show how semantic annotations and reasoning can be used to provide more quality analysis and enhancements of process models and event logs through concept matching (i.e. ontology classifications). We perform the semantic modelling and integration of the resulting process mappings with annotated terms and then describe the domain knowledge for the activity workflows and concepts defined in an ontology by using process description languages such as the Ontology Web Rule Language (OWL) [14] and Semantic Web Rule Language (SWRL) [15]. Reasoning on ontological knowledge plays an important role in the semantic representation of the processes. Besides, semantic reasoning allows the extraction and conversion of explicit information into some implicit information. For example, the intersection or union of classes, description of relationships and concepts or role assertions.

Classification, according to Han and Kamber [16] is one of the most universally data mining technique that aims at finding models or functions that describes or distinguishes data classes or concepts. One of the benefits of applying the technique is to help annotate the classification labels with sets of relations defined in an ontology especially for use in semantic enhancement of the captured datasets. Apparently, semantics encoded in classification tasks has the potential not only to influence the labelled data but also to handle large number of unlabelled data (Allahyari et al. [17],

Balcan et al. [18]). The authors in [18] incorporated ontology as consistency constraints into multiple related classification tasks by classifying multiple categories of unlabelled data in parallel to determine labels that violates the ontology. Also, d'Amato et al. [19] argue that classification is a fundamental task for a lot of intelligent systems or applications, and that classifying through logic reasoning may be both too demanding and frail because of inherent incompleteness and complexity within the knowledge bases. However, the authors observe that these methods adopt the availability of an initial drawing of ontology that can be automatically enhanced by adding or refining concepts, and have been shown to effectively solve process modelling problems (Okoye et al. [20]) using process description logics particularly those based on classification, clustering and ranking of individuals. Explicitly, the works in [19–21] show that the problems of modelling domain processes can be solved by transforming ontology population problem to a classification problem where for each entity within the ontology, and the concepts (classes) to which the entities belongs to have to be determined, hence, classified. Accordingly, Elhebir and Abraham [22] notes that pattern discovery algorithms makes use of statistical and machine-learning techniques to build models that predicts behaviour of captured datasets, and concedes that one of the most pattern discovery techniques used to extract knowledge from pre-processed data is Classification. The authors [22] observe that most of the existing classification algorithms attains good performance for specific problems but are not robust enough for all kinds of discovery problems and further propose that combination of multiple classifiers, i.e. hybrid intelligent systems (HIS), could be considered as a general solution for the pattern discovery because they obtain better results compared to a single classifier as long as the components are independent or have diverse outputs.

In principle, Baati et al. [23] propose two kinds of possibilistic classifiers for numerical data: one that extends the classical and flexible Bayesian classifiers by applying a probability-possibility transformation to Gaussian distributions, and the second, that directly express data in possibilistic formats using the idea of proximity between data values. According to the authors in Baati et al. [23,24] the Possibility theory, introduced by Zadeh [25] and further advanced by Dubois et al. [26] is a fusion theory based on fuzzy sets theory and are devoted to represent and combine imperfect information in a qualitative and/or yet quantitative way. Thus, information imperfections treated by possibility theory may represent the uncertainty due to variability of observations, the uncertainty due to poor information, the information ambiguity, or the information imprecision, etc. (Khaleghi et al. [27]). Even more, Baati et al. [24] notes that in many cases, the minimum-based possibilistic combination is likely to lead to a final decision that may have very close possibility estimate to other alternatives, and in such situation, the quality of decision may be seriously altered since the final classification tasks is likely to be inaccurate. However, to resolve this problem, the authors [24] states that the Generalized Minimum-based (G-Min) algorithm proposed in Baati et al. [28] can be employed to avoid those ambiguity between the final decision and the rest of classes, and thus, to find a decision with a possibility estimate widely away from other alternatives. According to the authors [24] the G-Min algorithm requires the matrix Π of possibilistic estimates and is based on two main steps: the first, aims to build a set of possible decisions, whereas, the second aims to filter those set in order to find a final class with a high score of reliability [28]. To this end, it is important that at the semantic level, the basic function in possibility theory is a possibility distribution (denoted as π) which assigns to each possible class c_j from C a value in either 1 (i.e. true) or 0 (i.e. false). The possibility value assigned to a class c_j stands for plausibility, i.e. the belief degree that this class is the right one. By convention, $\pi(c_j) = 1$ means that c_j is totally possible and if, $\pi(c_j) = 0$, c_j is considered as impossible. Besides, in this paper, we present a semantic-fuzzy mining technique that targets through conceptualization to turn process models and its analysis into a classification task (with a *training set* and a *test set* [3]) where the discovered models from the training set needs to decide whether traces found as a result of applying a classifier over the given test sets are fitting (true) or not (false). Indeed, the utilized approach aims at making use of semantic annotations to link elements in the event data logs with concepts that they represent in an ontology. The purpose of the semantic annotation process is to seek the equivalence between the *concepts of the fuzzy models* derived by applying the fuzzy miner algorithm on the readily available datasets and the *concepts of the defined enriched domain ontology*.

Zadeh [25,29] introduced the Fuzzy logics as an extension of the Boolean logic. The fuzzy logic allows a proposal to be in another state as true or false (Dammak et al. [30]). The logic is based on the mathematical theory of fuzzy sets [29] where each fuzzy

set is defined by its *linguistic variable* or better still *membership function*. According to Rozinat [31] Fuzzy mining algorithms are practically used to discover process models in a less precise manner and to visualize complex processes. In principle, flexible and less-structured models. According to the author [31] the fuzzy miner algorithms are applied with the goal to show understandable models for very unstructured and flexible processes. Thus, fuzzy mining is a one of the process discovery techniques that aims to address the issue of mining unstructured processes by using a mixture of abstraction and clustering techniques. Moreover, models discovered as a result of applying the fuzzy miner – are able to abstract from details and aggregate behaviours that are not of interest into cluster nodes. Fuzzy models attempts to automatically hide visual noise, and group together the model elements that are likely to have low information value for the user. With tools that supports the fuzzy miner algorithm, e.g. ProM [11] and Disco [32], user controls the level of model details by setting a threshold value on a slider. Noticeably, the results of such models or mappings are not often suitable for enacting a process on a workflow system, but instead, they provide a means to explore complex processes in an interactive manner and on a variable levels of abstraction. The author in [31] notes that the results of the fuzzy miner algorithms are *relaxed* in nature especially when compared with the semantics of other process modelling languages such as the Petri nets or BPMN. Tactlessly, even with the relaxed execution nature and the adaptive simplification mechanism exhibited by the fuzzy miner, the resulting models are mainly useful only as a descriptive means for complex and unstructured processes which eventually would produce the so-called *spaghetti* models [2] if they would be precisely represented. Hence, fuzzy models are ambiguous and tends to lack the real descriptions (semantics) behind the event logs about the domain processes.

On the other hand, Van der Aalst [2] notably states that fuzzy mining approaches or techniques provides an extensible set of parameters to determine which activities and arcs needs to be incorporated. The author mention that the fuzzy approach can construct hierarchical models, i.e., less frequent activities may be moved to sub processes and the representation of a roadmap is exploited to create process models that can be understood easily while providing implicit information on the frequency and importance of activities and/or paths. In addition, fuzzy mining algorithms views process models as if they are geographic maps, e.g. road maps or hiking maps [2], and such interpretation characteristically means that fuzzy models are only useful when the process analyst is interested on how the activities has been performed or the paths they follow during the process executions, but does not actually describe the semantics about relationships the process elements share within the process in question which shows the limitation of the hierarchical decomposition. Nonetheless, fuzzy mining approaches are useful especially in settings where the process owners, process analysts or IT experts are interested in process discovery algorithms that are capable of providing simplified process models. Besides, the proposed approach in this paper reveals how the ambiguous problem of fuzzy models and the lack of real descriptions (semantics) behind the event log labels can be resolved by bringing analysis of the resulting process models to a much more conceptual level by means of the semantic-fuzzy mining approach.

In summary, the method introduced in this paper as opposed to other benchmark algorithms, uses the semantics of the sets of activities within a domain process – case study of the learning process and models to generate rules and events relating to task, to automatically discover and ascertain the various process instances. The use case scenario together with the effort to address those semantic challenges with process mining techniques and analysis forms part of the contribution of this work. Interestingly, this kind of knowledge could be used by the process owners in understanding their everyday processes and more importantly grasp information on how to improve on them by having a real world insight about their processes in reality. Another benefit provided by our approach is the ability to describe the semantics behind the labels in an event log of the learning process considered useful for discovery of new knowledge about the domain processes. The main opportunity is that the process knowledge-base is enhanced as a result of its analysis being based on concepts rather than event tags or labels, after all, when these real conceptual knowledge are inferred, and the semantic rules are executed, the knowledge base is updated with the newly discovered knowledge. Thus providing the process owners and analysts with new ways of extracting and analysing the captured event data logs.

3. Semantic process mining: Framework, design and methods for process models discovery and conceptualization

One of the main purpose of process mining technique is to discover and interpret process models from

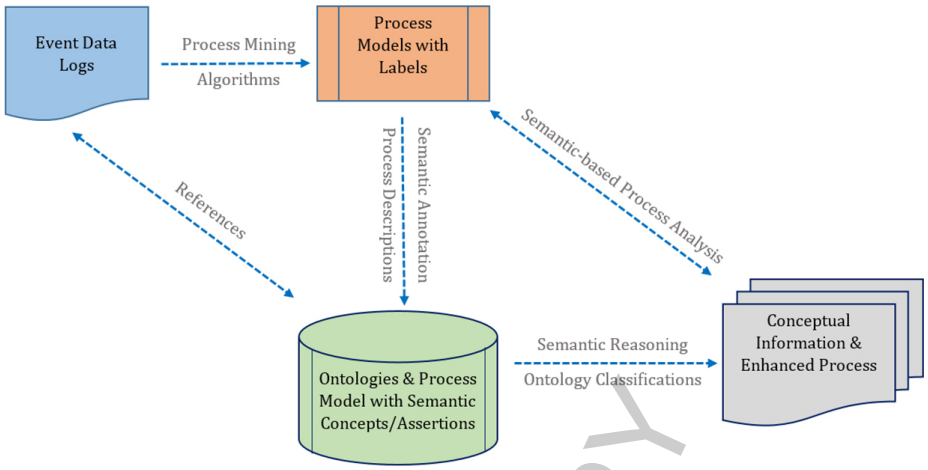

Fig. 1. Framework of the semantic-based process mining approach (2-dimensional rhombus mining technique).

event logs. Whilst on the other hand, the semantic-based process analysis supports the provision of domain knowledge (semantics) that can help improve or further enhance the information values of the discovered models. Indeed, one of the biggest challenge with process mining is mainly to find the right information and to understand what it means [3–6,33]. According to Rozinat [33] figuring out the semantics of existing information systems, or IT logs in many organisations can be anything between really easy and incredibly complicated. Most often, the outcomes largely depends on how distant the logs are from the actual business or organisational settings and the process logic. For instance, a performed learning process and steps may be recorded directly with their activity name, or a process analyst may need a mapping between some kind of hidden action code and the actual performed activity to be able to analyse the process. Instinctively, hints from current researches within the area of semantic process mining [5,6,21] and business process intelligence [3,33] suggests that it is best to work together with process analysts who can help extract the right information or data and explain the meaning of the different fields. Eventually, in terms of process mining, it helps not to try to apprehend everything at once but instead to focus first on the three critical elements [33]:

- How to differentiate process instances;
- Where to find the activity logs, and;
- The start and/or completion time or timestamps for the activities.

Perhaps, when these essential elements have been identified and addressed, subsequently, one can look further for additional metadata (process descriptions) that can help enhance the process analysis from a domain perspective. In view of that, the semantic-based approach and framework described in this paper focuses on these vital elements to look at what extent the effective raising of the learning process analysis from the syntactic to semantic level enable real time viewpoints on the process domain, and helps address the problem of analysing the available datasets based on concepts. The focus is on answering real time questions about relationships the process instances share amongst themselves within the process knowledgebase.

Furthermore, the quality augmentation of process models is as a result of employing mining approach which encodes the system with the three rudimentary building block – *semantic labelling* (annotation), *semantic representation* (ontology) and *semantic reasoning* (reasoner). Henceforth, it is important that we interpret how these components fit and rely on each other in carrying out the discovery of worthwhile process models, and consequently, promotes semantic enrichment of the resulting models. Over the next sub sections, we explain the various components of the proposed Semantic-Fuzzy mining approach including the different stages of its implementation, and then subsequently, look at the use case scenario of the *learning process* in order to show how this component's fits and is capable of analysing process models and event logs at a more conceptual level.

3.1. *Design framework of the semantic-fuzzy mining approach*

The design of the semantic-based process mining approach is primarily constructed on the following building blocks as shown in Fig. 1.

In Fig. 1 we describe the framework for the proposed semantic-based process mining and analysis (which we also referred to as the *2-Dimensional Rhombus approach*) which integrates the following:

- Extraction of process models from event data logs: the derived models are represented as a set of annotated terms which 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 *Reasoner* (inference engine): is designed to perform automatic classification of task and consistency checking to validate the resulting model as well as clean out inconsistent results, and consequently, presents the inferred (underlying) associations;
- 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 conceptual referencing supports reasoning over the ontologies in order to derive new information (knowledge) about the process elements and the relationships they share amongst themselves within the knowledge base.

Therefore, to summarize the design framework, we show that the application of semantic-based process mining and analysis approaches must focus on feeding the mining algorithms with two key core elements:

1. Event Logs and process models which elements have references to concepts in ontologies, and
2. Reasoners that can be invoked to reason over the ontologies used in the event logs/models.

Indeed, the implication of such semantic framework and its application have gained a significant interest within the field of process mining in recent years. On the one hand, the framework trails to make use of the semantics captured in event data logs (i.e. metadata about a process) to create new techniques for process mining and/or enhance existing ones to better support humans in obtaining a novel and more detailed accurate results. On the other hand, the semantic-based analysis helps to provide the process mining results at a more abstraction level so that they can more easily be grasped by the process owners, process analysts, or IT experts. Besides, event logs from various process domains usually carry domain specific information (semantics), but quite often, the traditional process mining algorithms lack the ability to identify and make use of such semantics across the different domains. In principle, the work in this paper shows using the example case study of the learning process and evaluation of the semantic fuzzy mining approach that by annotating and encoding the process models with rich semantics and the integration of semantic reasoning that it is possible to specify useful domain semantics which are capable of *bridging the semantic gap* conveyed by the traditional process mining techniques [1,5]. Thus, with the semantic-based process mining approach introduced in this paper, useful information (semantics) about how activities depend on each other in a process environment is made possible, and essential for extracting models capable of creating new knowledge. The technique has emerged due to the limitations identified with the existing process mining algorithms, and therefore, pursues to cater for such problems through its ability to describe the semantics behind the tags or labels in an event log considered useful for discovery of new knowledge and better still worthwhile process models. Currently, there are not too many algorithms that supports such semantic-based analysis, besides, semantic process mining is a new area in the field of process mining and there are few existing applications that demonstrates the capabilities of the technique.

4. Process modelling, event logs representation, concepts assertions & analysis

In this section, the study shows how semantic concepts and annotation can be used to provide more enhancements to process models and event logs analysis through concept matching (ontology classification) and semantic reasoning. For our approach, we perform the semantic modelling and integration of the resulting process mappings with annotated terms. The semantic model represents the domain knowledge for the activity workflows and concepts defined in an ontology by using process description languages such as the Ontology Web Rule Language (OWL) [14] and Semantic Web Rule Language (SWRL) [15]. The semantic depiction (representation) of the process models in an ontological form is a very important step in the proposed semantic-based process mining approach, aimed at unlocking the information value of the event logs and the derived process models by way of finding useful and previously unknown links between the process elements and the deployed models. Moreover, the use of the *reasoner* to infer individual process instances relies exclusively on the ability to represent such information in a formal way (ontology) to create plat-

form for a more conceptual analysis of the individual process instances. According to Gruber [34] ontologies, i.e. *Ont* ∈ *Onts,* are *formal explicit specification of (shared) conceptualization* that can be applied in any context as we exploited in this paper to model the research case study of a learning process. The annotated logs and models are very fitting for further steps of semantic lifting and analysis of the process models, because at this stage, the input data are presented in a formal and structured format that can connect to referenced concepts within the defined ontologies. The following Algorithm describes how we generate ontology from the process models and event logs.

Algorithm 1: Developing Ontology from process models and event logs

For all defined models ***M*** and event log ***EV***
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
Output: Semantic annotated graphs/labels & an ontology-driven search for process models and explorative analysis
Procedure: create semantic model with defined process descriptions and assertions
Begin
  For all process models ***M*** and event log ***EV***
    Extract Classes ***C*** ← from ***M*** and ***EV***
    while no more process element is left **do**
    Analyze Classes ***C*** to obtain formal structures
      If ***C*** ← Null **then**
        obtain the occurring Process instances (***I***) from ***M*** and ***EV***
      Else If ***C*** ← 1 **then**
        create the Relations (***R***) between subjects and objects // i.e. between classes ***C*** and individuals (***I***)
      If relations ***R*** exist **then**
        For each class ***C*** ← semantically analyse the extracted relationships (***R***) to state facts i.e. Axioms (***A***)
        create the semantic schema by adding the extracted relationships and individuals to the ontology
Return: taxonomy
End If statements
End while
End For

Ultimately, from the described Algorithm 1, we recognize that ontology is a quadruple $Ont = (C, R, I, A)$ which consists of different classes C and relations R between the classes [34]. A relation R connects a class either with another class or with a fixed literal and can define subsumption hierarchies between the classes and/or other relationships. Additionally, classes are instantiated with a set of individuals I, and can also contain a set of axioms A which state facts (e.g. what is true and fitting, i.e. true positives or what is true and not fitting, i.e. true negatives within the model) especially for use in semantic-based analysis of the process elements and models.

Therefore, to achieve this importance step in our approach it was necessary to:

- Create the various process domain ontologies, workflow ontologies, and the Individuals classes that will be inferred;
- Provide Process Descriptions for all Object and Data Types that allows for Semantic Reasoning and Queries (i.e. *CLASS_ASSERTIONS; OBJECT_PROPERTY_ASSERTIONS; DATA_PROPERTY_ASSERTIONS*);
- Create SWRL rules to map the existing class ontologies with concepts that are defined in the ontologies;
- Check for Consistency for all Defined Classes within the Model using Description Logic Queries.

Accordingly, the defined concepts and process descriptions as explained in the steps above are in line with the entire speculation of the work in this paper to show that a system which is formally encoded with semantic labelling (annotation), semantic representation (ontology) and semantic reasoning (reasoner) has the capability to lift process mining results and its analysis from the syntactic level to a much more conceptual level. This means that semantic annotation is an essential component in realizing such tools that supports semantic-based process mining by automatically conveying the formal semantics of the derived process models and extracted logs (Lautenbacher et al. [35,36]). Essentially, *semantic annotation* is described formally as a function that returns a set of concepts from the ontology for each node or edge in the graph:

SemAn : $N \cup E \rightarrow$ *COnts*, where: *SemAn* describes all kinds of semantic annotations which can be input, output, meta-model annotation etc. and: *COnts* are the set of concepts from the ontology.

According to Lautenbacher et al. [35] it is important to note that semantic annotation can either be

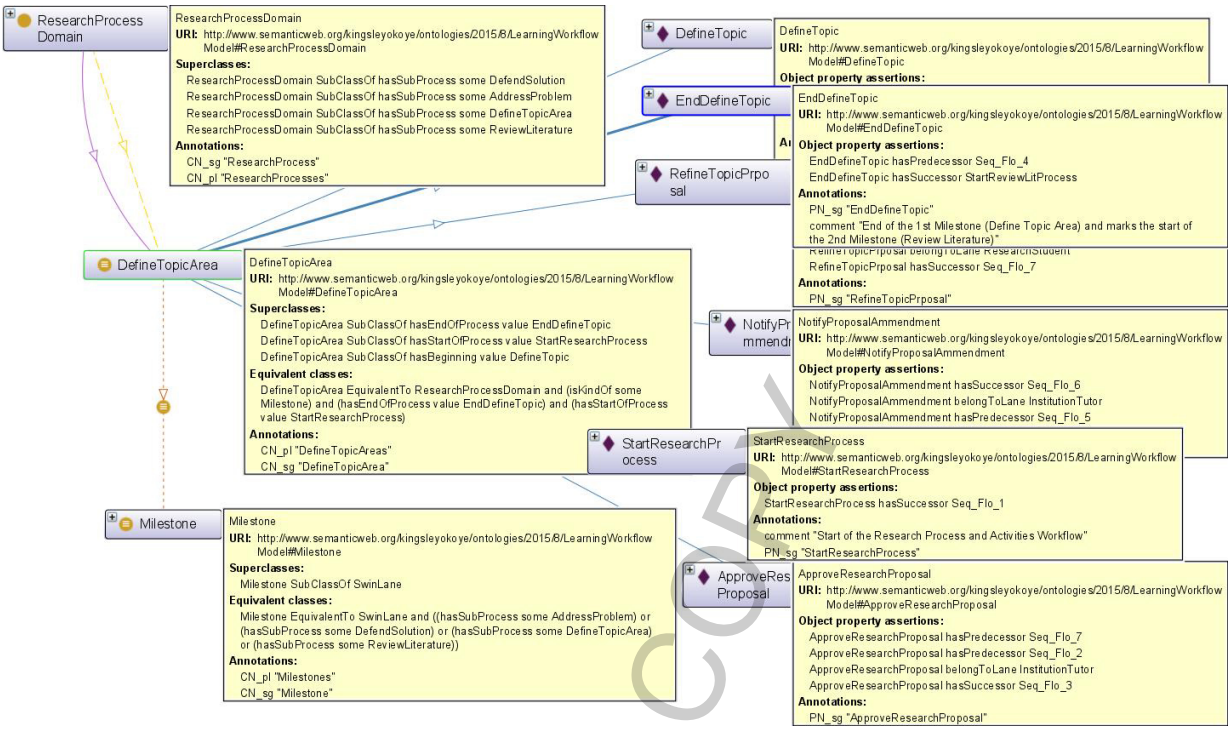

Fig. 2. Example of a semantic annotated graph with process descriptions and assertions for the different graph nodes.

done manually, semi-automatic or computed automatically considering word similarities (Born et al. [37]) to generalize the individual entities within the domain process in view. It is also important to note that the semantic-based planning often requires that all process executions include some form of *semantic annotation*. Thus,

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*.

Hence, a *semantic annotated graph* as shown in Fig. 2 can be defined as follows:

Gsem = (*Nsem, Esem, Onts*) with *Nsem* = {(*n, SemAn*(n)) | $n \in N$} and *Esem* = {(*nsem, n_sem*) | *nsem* = (*n, SemAn*(n)) $\wedge$ *n_sem* = (*n_, SemAn*(*n_*)) $\wedge$ (*n, n_*) $\in E$} [35].

Lastly, the third essential component in realizing the semantic-based approach as described in the proposed framework in this study is the capability of performing *semantic reasoning* to classify and even more check for consistency for all the defined classes and relationships that exist within the model. This means that based on the process description/assertions within the domain ontology, the *reasoner* is able use the underlying informations to check if it is possible for any instances (individuals) to become a member of a class, and to produce the necessary results as requested based on the query or information retrieval process. Indeed, the use of the reasoner to compute the relations between the concepts in the ontologies can be utilized to collectively combine tasks and/or compute process models in a hierarchical form (taxonomy) including several levels of abstraction. This means that the process models are either semantically annotated as earlier described in this paper, or already in a form which allows a computer (i.e. the reasoner) to infer new facts by making use of the underlying ontologies.

The following Algorithm 2 describes how this work makes use of the reasoner to classify and infer the necessary association to produce the outputs.

Indeed, as shown in the Algorithm 2, *semantic reasoning* (or better still *ontology classifications*) helps to infer and associate meanings to labels within the defined ontologies by referring to the concepts assertions (i.e. Objects and Datatype properties) and sets of rules/expressions that are defined within the ontologies to answer and produce meaningful knowledge, and even in many cases, new information about the process elements and the relationships they share amongst

themselves within the knowledge base. To this end, this work describes in the following sub sections – the use case implementation of the semantic-based process analysis, design framework and algorithms including the sematic-based planning and the algorithm formalizations.

Algorithm 2: Reasoning over Ontologies and Classification of Process Parameters and Outputs

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

4.1. Case study of learning process and problem scenario

The use case scenario in this paper is based on running example of a *Research Process* – to prove how the proposed semantic-based approach can be used to answer real time questions about a learning process, as well as, use in validation of our experiments. In our case study example we show that the first step to conducting a research is to decide on what to investigate, i.e. the research topic, and then go about finding answers to the research questions. At the end of the process, the researcher is expected to be awarded a certificate. These process involves the workflow of the journey from choosing the research topic to being awarded a certificate, and comprises sequence of practical steps or set of activities through which must be performed in order to find answers to the research questions. The workflow for these steps are not static, it changes as a researcher travel along the research process. At each phase or milestone of the process, the researcher is required to complete a variety of learning activities which will help in achieving the research goal. Even more, from event log and mining perspective, the derived process models may not disclose to us how the individual process instances that makes up the model interact or differ from each other (i.e. the semantic abstraction levels), which attributes they share amongst themselves within the knowledge base, or the activities they perform together or differently, despite all of the useful information from mining the process. For example, questions like – who are the individuals that have successfully completed the research process? may not be established. For this reason, we show in this paper that by adding semantic knowledge to the deployed models, it becomes possible for one to determine and address the identified problems. To explicate such tactics, we presume that for a research process to be classified as successful, it is necessary that the researcher must complete a given set(s) of milestones in order to be awarded the degree. Moreover, in any case whereby the researcher has not completed the set(s) of milestones which is necessary to ensure the research outcome, such learner can be classified as incomplete. In so doing we can ascertain which individuals has successfully completed the research process or not. Over the next sub section, this study describes how we make use of the case study (Research Process domain) to illustrate the capability of our approach by analyzing the learning activities in the event logs based on the defined concepts, thus, presenting the mining results at a much more conceptual level.

4.1.1. Semantic representation of the research learning process

Here, we implement the semantic-based approach to find out patterns/behaviour that describes or distinguishes certain entities within the learning knowledge base by recognizing what attributes/paths the learners (i.e. process instances) follow or have in common, or what attributes distinguishes the successful learners from the incomplete ones. The purpose is not only to answer the specified questions by using the semantic-based approach, but to show how by referring to at-

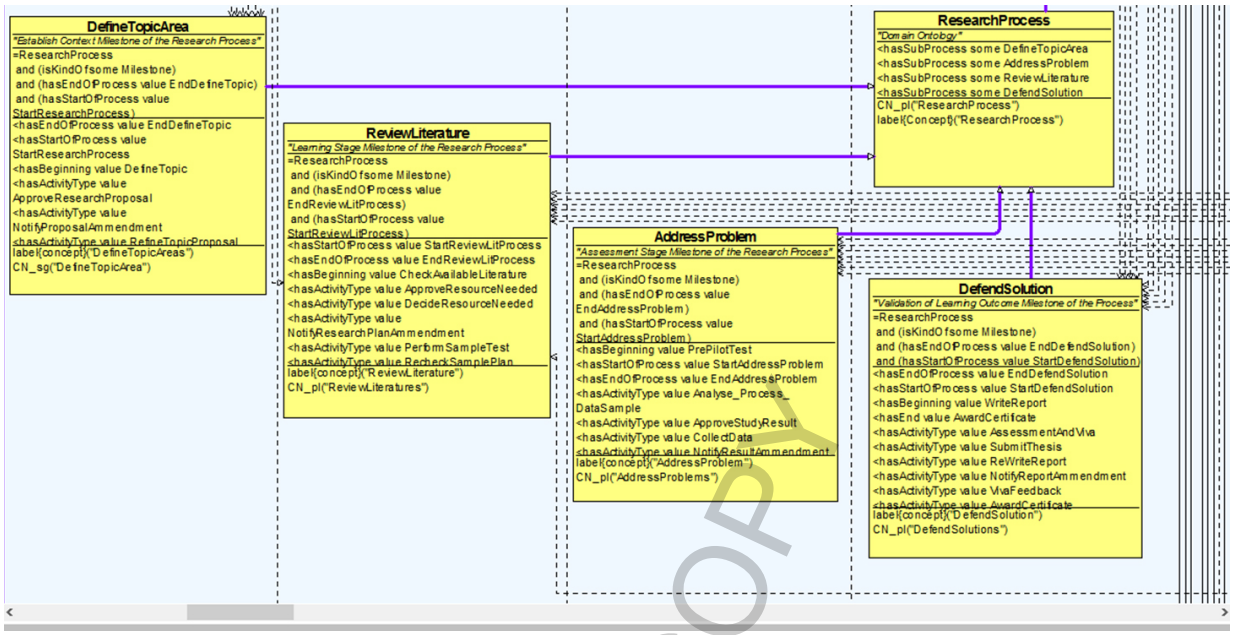

Fig. 3. Research process domain with description of the learning activity concepts and relationships.

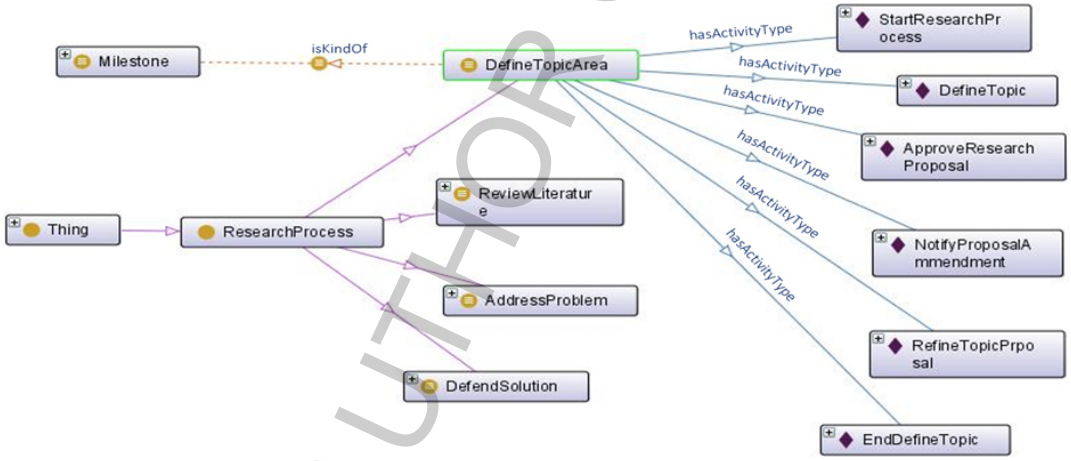

Fig. 4. OntoGraph and the ActivityConcept mapping for the *DefineTopicArea* milestone.

tributes (concepts) and the application of semantic reasoning, it becomes easy to refer to a particular case (i.e. certain group of learners) which in our example we focus on the use case of *Successful* and *Uncomplete* learners. Accordingly, we show that the flow of the research process from the definition of research topic to being awarded a certificate; consist of different learning steps which a researcher has to or partly perform in order to complete the research process [4,7]. We provide four milestones; *Establish Context* → *Learning Stage* → *Assessment Stage* → *Validation of Learning Outcome* in order to determine and explain the steps taken during the research process, thus, from Defining the Topic Area – *to* – Review Literature – *and* – Addressing the Problem – *then* – Defending the Solution. These milestones consist of sequence of activities, and the order in which the individual learning activities are carried out has the capability of determining the research outcome. Furthermore, as described in Fig. 3 we show the *Learning Activity* concepts that are defined in the learning ontology model, and how they are mapped to the various milestones of the Research Process in order to ensure sequence of transitions during the entire learning process. In Fig. 4 we show an example

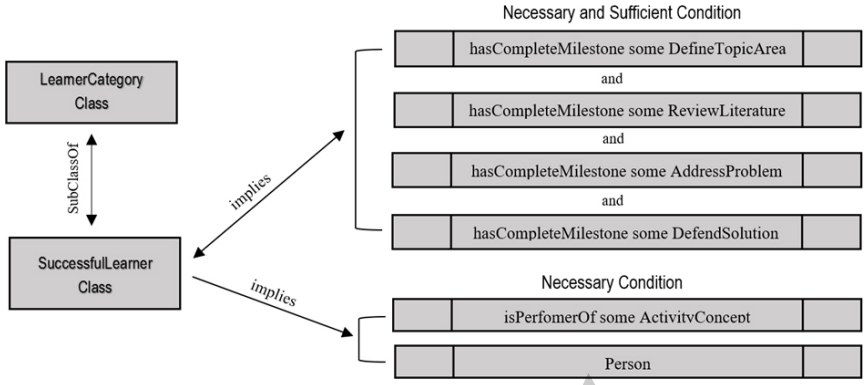

Fig. 5. Attributes/object property assertions for the *SuccessfulLearner* class.

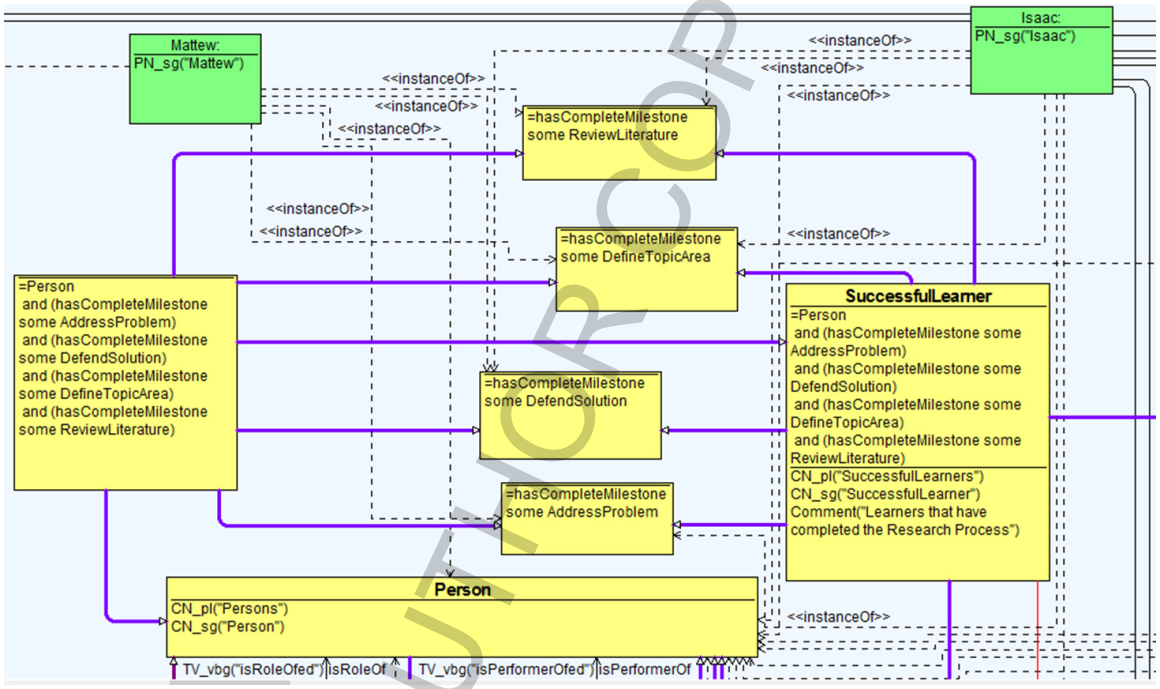

Fig. 6. Concept assertions and the various structural relationships for the *SuccessfulLearner* class.

of the DefineTopicArea activity concepts and the relations between the process instances (entities) that are defined in the resulting model.

The drive for the semantic planning and mapping of the activity concepts is that the approach allows the meaning of the learning objects and properties to be enhanced through the use of *property characteristics* and *classification of discoverable entities.* For instance, to address the real time learning questions we have identified in sub Section 4.1 in relation to the successful and uncomplete learners, we refer to the deployed model, and to this effect, describe 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* – Review Literature – *and* – Addressing the Problem – *then* – Defending the Solution). Moreover, as shown in the example Fig. 5 – the *necessary condition* is: if something is a Successful Learner, it is *necessary* for it to be a participant of the Learning ActivityConcept class and *necessary* for it to have a kind of sufficiently defined condition and relationship with the ResearchProcessClass: DefineTopicArea, ReviewLiterature, AddressProblem and DefendSolution.

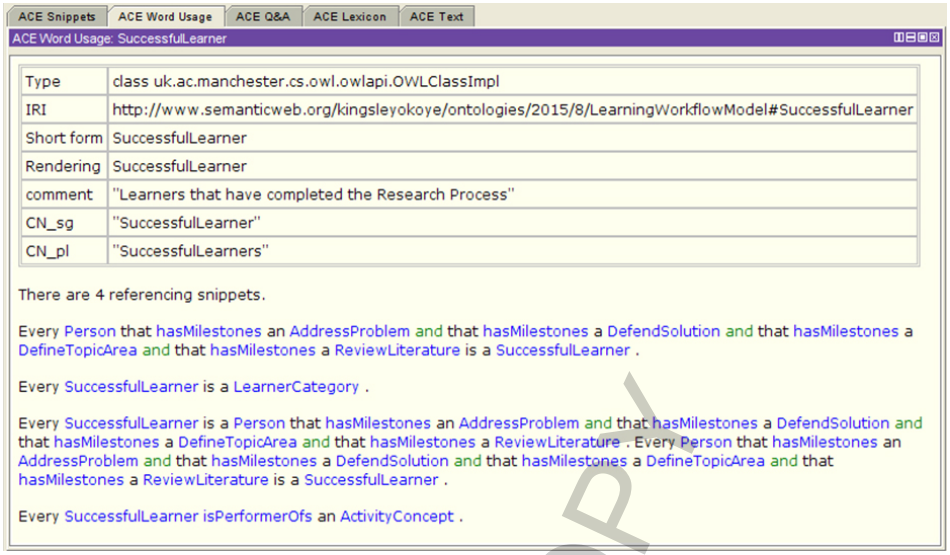

Fig. 7. Example of referencing class expressions for the *SuccessfulLearner* class.

Ideally, we notice that the *Object Property Restrictions* are used to infer anonymous classes that contains all of the individuals that satisfies the restriction. In essence, all of the individuals that have the relationship required to be a member of the successful learner Class. The consequence is the *necessary and sufficient condition:* which makes it possible to implement and check for consistency in the model, meaning that it is necessary to fulfil the condition of the universal or existential restriction – for any individual to become a member of the class, as we have used to answer the real life learning question. Indeed, process restriction properties (structured organisation) and semantic labelling (assertions) serves as a good practice for representation of the learning process information by providing a formal way of representing the individual process instances within the learning knowledge base as illustrated in Figs 6 and 7. For example, the following are description of the implemented ontology concepts and axiom for the *successful learner* class within the learning model following the definitions in Figs 6 and 7 including the OWL XML file syntax as follows:

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

```
<EquivalentClasses>
  <Annotation>
    <AnnotationProperty IRI="http://atte
    mpto.ifi.uzh.ch/acetext#acetext"/>
    <Literal datatypeIRI="&xsd;string">
    Every SuccessfulLearner is a Person
    that hasMilestones an AddressProblem
    and that hasMilestones a
    DefendSolution and that
    hasMilestones a DefineTopicArea and
    that hasMilestones a
    ReviewLiterature. Every Person that
    hasMilestones an AddressProblem and
    that hasMilestones a DefendSolution
    and that hasMilestones a
    DefineTopicArea and that
    hasMilestones a ReviewLiterature is
    a SuccessfulLearner.</Literal>
  </Annotation>
  <Annotation>
    <AnnotationProperty IRI="http://purl
    .org/dc/elements/1.1/date"/>
    <Literal datatypeIRI="&xsd;string">
    2016-04-19 13:40:36</Literal>
  </Annotation>
  <Class IRI="#SuccessfulLearner"/>
  <ObjectIntersectionOf>
    <Class IRI="#Person"/>
    <ObjectSomeValuesFrom>
      <ObjectProperty IRI=
      "#hasCompleteMilestone"/>
      <Class IRI="#AddressProblem"/>
    </ObjectSomeValuesFrom>
    <ObjectSomeValuesFrom>
```

```
      <ObjectProperty IRI=
      "#hasCompleteMilestone"/>
      <Class IRI="#DefendSolution"/>
    </ObjectSomeValuesFrom>
    <ObjectSomeValuesFrom>
      <ObjectProperty IRI=
      "#hasCompleteMilestone"/>
      <Class IRI="#DefineTopicArea"/>
    </ObjectSomeValuesFrom>
    <ObjectSomeValuesFrom>
      <ObjectProperty IRI=
      "#hasCompleteMilestone"/>
      <Class IRI="#ReviewLiterature"/>
    </ObjectSomeValuesFrom>
  </ObjectIntersectionOf>
</EquivalentClasses>
```

4.2. Formalization of the semantic learning process mining algorithm

The following section describes the semantic learning process mining algorithm formalization and ordering for our proposed approach. We show how by constructing semantic process models and description of the process elements based on the learning activity concepts, it becomes possible for us to determine the individual learning patterns/behaviours within the learning process knowledge base.

The semantic learning process algorithm (SLPM) formalization in [7] explains the basis for our approach. To expound the strategies for constructing the learning activity concepts and classification of learning classes (sub sets), we propose in this paper the following algorithm:

Algorithm 3: Generating process instances, classes, and learning sub sets for defined ActivityConcept ***AC***.

For all definite classes and process descriptions
Input: ***AC***, learners prior activity list ***ACL_List***
Output: ***AC***'s learning activity sequence set ***LS***
Procedure: Generate Learning Activity Classes & Subsequence Sets
Begin
LS = Null
AC_*ProcessInstance_List* = Null
AC_*LearningActivity* = 0
LS ← ***LS*** + ***AC***
For each ***Ci*** ∈ ***LS***
  Ci_*Precondition_List* ← Get_Precondition (*OWL_xml*_***Ci***)
  For each ***Cj*** ∈ ***Ci***_*Precondition_List*
    Cj_CorrespondingSubclassSet_List = Null
    Cj_ProcessInstance_List = Null
    If ***Cj*** ∉ ***ACL_List*** **AND** ***Cj*** ∉ ***LS*** **then**
      LS ← ***LS*** + ***Cj***
      Cj_ CorrespondingSubclassSet_*List* ← ***Cj***_ CorrespondingSubclassSet_*List* + ***Ci***
      Cj_*ProcessInstance_List* ← ***Cj***_*ProcessInstance_List* + ***Ci*** + ***Ci***_*ProcessInstance_List*
      Cj_*LearningActivity* = ***Ci***_*LearningActivity* + 1
    Else If ***Cj*** ∉ ***ACL_List*** **AND** ***Cj*** ∉ ***LS*** **AND** ***Cj*** ∉ ***Ci***_*ProcessInstance_List* **then**
      Cj_ CorrespondingSubclassSet_*List* ← ***Cj***_ CorrespondingSubclassSet_*List* + ***Ci***
      Cj_*ProcessInstance_List* ← ***Cj***_*ProcessInstance_List* + ***Ci*** + ***Ci***_*ProcessInstance_List*
      If ***Cj***_*LearningActivity* < ***Ci***_*LearningActivity* + 1 **then**
        For each ***Ck*** ∈ ***LS***_*Subsequently*_***Cj***
        Ck_*LearningActivity* = ***All*** *(Ck_CorrespondingSubclassSet_ LearningActivity)* + 1
Return ***LS***
End If
End For

Accordingly, it is important to note that from the use case scenario and example of the Learning process, we refer that the research process comprises of the workflow (i.e. sequence of steps) or set of activities through which the learners has to perform in order to find answers to the research questions. Hence, a single set of learning activity will not be practicable for a learner to meet this goal because the learning activities and concepts themselves may have prerequisites that the learner has to complete before moving to the next stage or milestones of the process. In view of that, there is need to provide pre-defined activity concepts to be able to identify or monitor the entire process, and in any case for particular set of individuals or process instances. The learning activity concepts and class generation Algorithm 3 outlines the executions taking place during the generation of instance lists for defined activity concepts within the learning knowledgebase. Hence, for each concept ***Ci*** in the current learning process, first extract the precondition (prerequisite) list from its OWL file description *OWL_xml*_***Ci***. Then for each concept ***Cj*** in the class list, *if* it does not belong to an activity list and the corresponding subclass sets, add it into the learning activity sets and revise ***Cj***'s correspondingSubclassSet list, process instance list, and number of steps to the targeted learning concepts as

described in line 17 to 19. *If* ***Cj*** already exists in the learning class list, but does not belong to the activity list and the individual (process instance) list of ***Ci***, *End* the process, but also update its corresponding subclass list, process instance list, and number of steps to the target learning concepts as described in line 20 to 25.

Therefore, in principle *if* use the following standard notations, $\boldsymbol{R}$ to refer to the research process, and $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{d}$, for the activity concepts [7]: *Then*

$\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d} \in \boldsymbol{R}$ is a function with domain $\boldsymbol{R}$ and process logs $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}$

Domain $\boldsymbol{R}$ is a SuperClass of the SubClasses $\boldsymbol{a}$, $\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}$.

The Subclass (also referred to as Subset) is a set where each of the individual Learning Activity occurs and sometimes may occur multiple times. For example, $[\boldsymbol{a1}, \boldsymbol{a2}, \boldsymbol{a3}, \boldsymbol{a4}, \boldsymbol{a2}, \boldsymbol{a5}]$ may be the sequence set of learning activity for Person, $\boldsymbol{P} \ldots \boldsymbol{n}$ over $\boldsymbol{a}$ (the DefineTopicArea Milestone), hence,

$$\boldsymbol{P} \ldots (\boldsymbol{a}) = |n \subseteq \mathcal{L}a|.$$

So therefore, *If*
$\boldsymbol{a1}$ = Define Topic
$\boldsymbol{a2}$ = Approval Activity
$\boldsymbol{a3}$ = Topic decline
$\boldsymbol{a4}$ = Refine Topic
$\boldsymbol{a5}$ = End Topic Proposal
Then, the sequence set of activities for $\boldsymbol{P} \ldots \boldsymbol{n}$ $(\boldsymbol{a})$ = {*Define Topic, Approval Activity, Topic Decline, Refine Topic, Approval Activity, End Topic Proposal*}.

On the other hand, our focus is on computing the sets of individual process instances that has completed (*successful learners*) or not completed (*incomplete leaners*) the research process. We note that to complete a research process, one must complete a given set(s) of milestones and must perform the set (or perhaps a subset) of the activities that comprise it. Given the fact for transition purposes, a process instance does not move on to the next milestone without completing a distinctive sequence set of learning activities that makes up the milestone or preceding learning concepts. So, for this reason, the sum or difference in process logs for a named person, $\boldsymbol{P}$, is defined in a straightforward way:

$$\boldsymbol{P} \ldots \boldsymbol{n} = |n \subseteq \mathcal{L}a| \pm |n \subseteq \mathcal{L}b| \pm |n \subseteq \mathcal{L}c| \pm |n \subseteq \mathcal{L}d|.$$

Thus, $\boldsymbol{P} \ldots \boldsymbol{n}$ is a finite set $|n \subseteq \mathcal{L} \in R|$.

For example, we describe in Figs 6 and 7 that "Every Person that hasCompleteMilestone a DefineTopicArea and that hasCompleteMilestone a ReviewLiterature and that hasCompleteMilestone an AddressProblem and that hasCompleteMilestone a DefendSolution is a SuccessfulLearner".

Thus, the Class Successful Learners, ***PSL***, is the sum of the set of activities log, $\mathcal{L}$, that a learner has completed for the learning activity milestones $\boldsymbol{a}$, and $\boldsymbol{b}$, and $\boldsymbol{c}$, and $\boldsymbol{d}$. Hence

If ***PSL*** is the Class that consist of the set $|SL \subseteq \mathcal{L}a| + |SL \subseteq \mathcal{L}b| + |SL \subseteq \mathcal{L}c| + |SL \subseteq \mathcal{L}d|$
Then ***PSL*** is the set $|SL \subseteq \mathcal{L} \in R|$.

In the same way, we also defined in reference [7] that "Every Person that hasOnlyCompleteMilestone a DefineTopicArea or that hasOnlyCompleteMilestone a ReviewLiterature or that hasOnlyCompleteMilestone an AddressProblem is an UncompleteLearner".

Accordingly, the Uncomplete Learners, ***PUL***, is the class of leaners where some set(s) of activities for the milestone $\boldsymbol{a}$, or $\boldsymbol{b}$, or $\boldsymbol{c}$, or $\boldsymbol{d}$ is missing over a finite set $|n \subseteq \mathcal{L} \in R|$. Hence,

If ***PUL*** is a Class that consist of the set $|UL \subseteq \mathcal{L} \in R - a|$ *or* $|UL \subseteq \mathcal{L} \in R - b|$ *or* $|UL \subseteq \mathcal{L} \in R - c|$ *or* $|UL \subseteq \mathcal{L} \in R - d|$,
Then ***PUL*** is the set $|UL \subseteq \mathcal{L} \in R - 1|$.

5. Fuzzy to semantic fuzzy mining: Experimentations and process analysis

To describe how we utilize and expand the amalgamation of two process mining techniques namely: Fuzzy Miner and Business Process Modelling Notation (BPMN) approach which we previously employed in [8] in order to weigh up the performance of the semantic-based fuzzy miner: to perform a more accurate classification of the individual traces within the process knowledge-base, and the capability to discover worthwhile process models given a dataset with *training set* and a *test set* provided in reference [3] where the discovered model from the training set needs to decide whether traces found as a result of applying a classifier over the given test set are fitting or not.

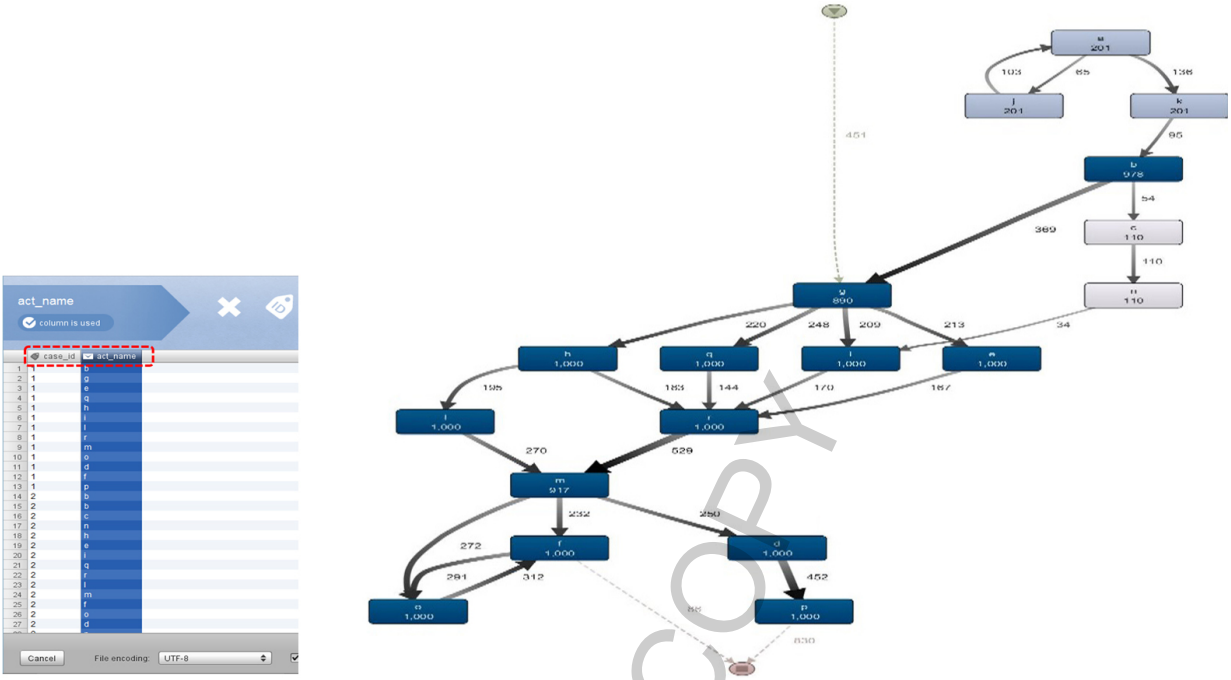

Fig. 8. Example of process model for *training_log_1* discovered using the fuzzy miner in Disco [32].

Firstly, for this step, we discover 10 process models from the training sets using the Fuzzy miner [38] and then makes use of the Business Process Modelling Notations (BPMN) [2] to analyse and provide the replaying semantics of the process models. Figure 8 shows an example of the discovered fuzzy models for the training set using the Disco tool [32]. The resulting process map allows us to quickly, and interactively explore the process into multiple directions and more importantly reveals the workflow-net [2] for the individual cases that makes up the process.

Furthermore, we perform a classification task for the *test set* [3], to generate the various cases (sub-processes) that makes up each of the process executions. We also explicate how we generated the 20 individual traces for each of the test log and the sequence of the activity executions for each individual trace. The *data set* [3] that has been provided by the IEEE CIS Task Force on Process Mining for the Process Discovery contains the typical information needed to perform process mining and implementation of the Fuzzy-BPMN miner as well as the proposed semantic-fuzzy mining approach. The data represents events logs generated from a business process model to show different behavioural characteristics. We assume that each of the event log contains data related to a single process which refers to a single process instance (*Case*) and can be related to some type of *Activity*. According to Van der Aalst [39] a "Case ID" and "Activity" is the minimum requirement for any process mining task. Equally, the given event logs [3] contains two attributes *case_id* and *act_name* as shown in Fig. 8 which precisely specify the requirements that allows for implementing the process discovery technique following the Definition 4.1 in [39].

We assume the following standard:

- $\#case\_id(e)$ is the Case associated to any event e.
- $\#act\_name(e)$ is the Activity associated to event e.

The standard definitions were necessary because for our approach the activities play an important role for the discovered model and thus corresponds to the individual cases within the discovered fuzzy model. As there are multiple events referring to the same *Activity*, we support the filtering of the 200 individual traces that makes up the test event logs [3] with a *classifier* [39]. A *classifier* is a function that maps the attributes of an event onto a label used in the resulting process model (Definition 4.2 in [39]).

Obviously, if we use the notation e to refer to the event name used in the process model, then the classifier for any event in the given log will be, $e \in E$, where e is the name of the event. Since the events are simply identified by their activity name (*act_name*), we then

assume

$$\underline{e} = \#act\_name(e)$$

We apply the classification conversion of the event logs provided, i.e. simple event log (Definition 4.4 in [39]) to obtain the test Log traces.

Applying the described simple event log definition: Let A be a set of *act_name*. A simple or single trace σ is a sequence of activities, i.e., $\sigma \in$ A*. A simple event Log L is a multiset of traces over some set A.

Thus, $L \in \mathbb{B}$ (A*)

For the *training set* [3] there are 1000 cases (trace) that defines the log. However, our focus is to identify the set of 200 traces that characterize the *test log* for use in validating the model following the objective and positioning of the process discovery [3]. Thus,

- Given a trace (t) representing real process behaviour, the process model (m) classifies it as allowed, or
- Given a trace (t) representing a behaviour not related to the process, the process model (m) classifies it as disallowed.

Apparently, there are total number of 200 traces from the *test log* to be classified. Therefore, if we Let $L \subseteq$ C be the event logs for the test log, and assuming that the classifier $e \in E$, is applied to the set of sequence data, then from (Definition 4.5 in [39])

$$\langle e1, e2, \ldots, en\rangle = \langle e1, e2, \ldots, en\rangle$$

where: $\underline{L} = [(\hat{c})|c \in L]$ is the simple event log corresponding to the *test log*.

All the Cases in the test Log are converted into sequences of the activities (*act_name*) using the classifier. Hence

A Case $c \in L$, is an identifier from the case C.

$\hat{c} = \#trace(c) = \langle e1, e2, \ldots, en\rangle \in \varepsilon*$ is the sequence of events executed for c

$(\hat{c}) = \langle e1, e2, \ldots, en\rangle$ maps these events onto the activity names (act_name) using the classifier.

From the defined classification formula, $e = \#act\_name(e)$, we obtain from the data containing the set of 200 traces for the test event log, i.e. (*test_log_april_1*) to (*test_log_april_10*) with 20 traces for each log as shown below:

$\underline{L}$(test_log_april_1) =

$[\langle b, g, e, q, h, i, l, r, m, o, d, f, p\rangle,$

$\langle b, b, c, n, h, e, i, q, r, l, m, f, o, d, p\rangle,$

$\langle g, h, i, q, q, m, r, o, e, d, p\rangle,$

$\langle j, a, k, b, b, g, e, h, q, l, r, i, m, d, f, o, p\rangle,$

$\langle b, g, h, i, q, i, r, m, o, d, p, f\rangle,$

$\langle e, e, e, q, h, r, d, o, r, p\rangle,$

$\langle g, h, e, i, i, q, l, m, o, f, p, d\rangle,$

$\langle b, a, j, k, g, e, q, h, l, i, r, m, o, f, d, p\rangle,$

$\langle g, i, e, r, l, i, m, d, o, p, d, p\rangle,$

$\langle b, b, g, e, l, l, h, q, r, r, r, d, o, o, p, f\rangle,$

$\langle b, g, e, h, i, q, l, r, m, d, p, o, f\rangle,$

$\langle b, q, g, h, i, h, l, m, m, r, p, f\rangle,$

$\langle h, g, h, e, r, l, q, i, f, f, p\rangle,$

$\langle b, j, a, k, g, q, e, i, h, l, r, f, d, o, p\rangle,$

$\langle c, n, q, e, i, h, r, d, m, o, p, f, p\rangle,$

$\langle b, g, h, i, e, q, r, l, m, d, o, p, f\rangle,$

$\langle g, i, h, e, r, q, m, l, o, d, f, p\rangle,$

$\langle k, b, n, n, c, h, h, e, q, l, q, r, r, i, m, f, f, i, p\rangle,$

$\langle b, b, b, g, q, i, h, e, r, l, m, f, o, d, p\rangle,$

$\langle b, b, g, q, e, h, i, r, m, l, d, o, p, f\rangle]$

The Log $\underline{L}$ (*test_log_april_1*) is an example of the set of 20 traces which we obtained for the *test_log_april_1*. Further examples of all the other classified traces for the complete test logs can be found in [8].

In view of the trace classifications, with the Fuzzy-BPMN miner approach we determine the *fitness* (replaying semantics) of the individual traces for the test event log classifications results cross-validated against the discovered process models from the *training logs*. To achieve the set objective, it was necessary to construct BPMN model with notational elements capable of describing the nesting of individual activities (also referred to as *task*) by using the event-based *AND, XOR,* and *OR* split and join gateways. According to Van der Aalst [2,39] an event within a BPMN model is comparable to a place in a Petri net, and just like Petri net, are token based semantics which can be used to replay a particular trace within the discovered process model. Since our target is to classify as correctly as possible the traces which are allowed and the traces which are not allowed in the original process model, we utilise the BPMN event-based gateways to replay the classified traces alongside the derived model from the training event log, and in so doing, identify which traces that are fitting or not fitting within the model. To do this, we used the *Convert Petri net to BPMN* plu-

Table 1
Trace fitness and classification table for the test event logs (*test_log_april_1* to *test_log_april_10*) using the Fuzzy-BPMN miner

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
Trace_1	TP *	TN *	TP *	FP	TN *	FP	TP *	TP *	TP *	TP *
Trace_2	TN *	TN *	TP *	TP *	TP *	TP *	TP *	TN *	TP *	TP *
Trace_3	TP *	TP *	TP *	TN *	TN *	FP	FP	TP *	TP *	TN *
Trace_4	TP *	TP *	FP	TP *	TN *	TP *	TN *	TP *	TP *	FP
Trace_5	TN *	FP	FP	TP *	TN *	TP *	TN *	TP *	TP *	TN *
Trace_6	TP *	FP	FP	TP *	TN *	TP *	TP *	TN *	TN *	TP *
Trace_7	TN *	TP *	TP *	TN *	TN *	TP *	TN *	TP *	TN *	TN *
Trace_8	TN *	TP *	TP *	FN	TN *	FP	TP *	TP *	TP *	TP *
Trace_9	TP *	TN *	TP *	TN *	TP *	FP	TP *	TP *	TN *	TP *
Trace_10	TP *	FP	TP *	TN *	TN *	FP	TP *	TP *	TP *	TP *
Trace_11	TN *	TP *	TP *	FN	TP *	TN *	TN *	FP	TN *	TP *
Trace_12	TP *	FP	FP	TP *	TP *	TP *	TP *	FP	TP *	TN *
Trace_13	TP *	TP *	FP	TN *	TP *	FP	TN *	TN *	TN *	TP *
Trace_14	TN *	TP *	TN *	TN *	TN *	FP	TN *	TP *	TN *	TP *
Trace_15	TP *	TN *	TN *	TN *	TP *	TP *	TN *	TN *	TN *	TN *
Trace_16	TN *	TN *	FP	TP *	TP *	FP	TN *	FP	TP *	TN *
Trace_17	TP *	TP *	TP *	TP *	TP *	TP *	TP *	TN *	TN *	TP *
Trace_18	TN *	TP *	FP	TN *	TP *	TP *	TP *	TN *	TN *	TN *
Trace_19	TN *	TP *	TP *	TP *	TN *	TP *	TP *	TP *	TN *	TN *
Trace_20	TN *	TN *	FP	TN *	TP *	FP	TN *	TN *	TP *	TN *
True Positive (TP):	10	10	10	8	10	10	10	10	10	10
False Positive (FP):	0	4	8	1	0	9	1	3	0	1
True Negative (TN):	10	6	2	9	10	1	9	7	10	9
False Negative (FN):	0	0	0	2	0	0	0	0	0	0
NO. of traces correctly classified	20	16	12	17	20	11	19	17	20	19

The cells colours indicates the classification attempt for each of the traces discovered from the test event logs. Also, the cells with gold sign * indicates the traces that were correctly classified by the Fuzzy-BPMN Miner with total of 171 traces out of 200.

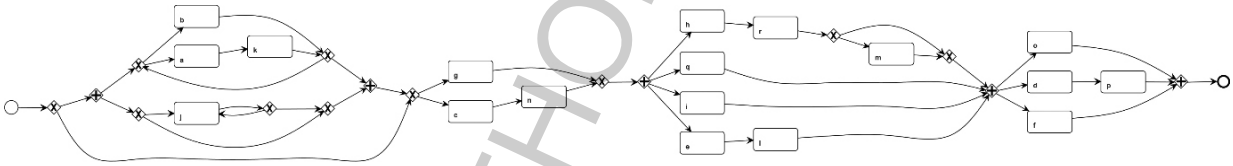

Fig. 9. Example of discovered BPMN model for the *training_log_1* with the event-based split and join gateway.

gin in ProM [11] to discover the BPMN models for the training logs. Figure 9 shows an example of the resulting BPMN diagram discovered for the *training_log_1*.

Consequently, in Table 1 we present the classification results and analysis of the Fuzzy-BPMN miner approach for the *test_log_april_1* to *test_log_april_10* cross-validated against the corresponding training set: where each cell indicates if the discovered model classifies the corresponding trace as fitting (allowed) or not fitting (disallowed). The columns represents the process models for the 10 training logs, while the rows represents the individual traces for the test log. For example, cell at (row *Trace_3*; column *Training model_5)* contains the classification attempt for the 3rd trace discovered from the *test_log_april_5* cross-validated against the *training_log_5*.

As shown in Table 1, the following metrics were used to measure the fitness of the individual traces from the datasets, where:

- *TP* is the number of *true positives* (i.e. instances that are correctly classified as positive);
- *FN* is the number of *false negatives* (i.e. instances that are predicted to be negative but should have been classified as positive);
- *FP* is the number of *false positives* (i.e. instances that are predicted to be positive but should have been classified as negative);
- *TN* is the number of *true negatives* (i.e. instances that are correctly classified as negative).

Accordingly, the cells with gold sign (*) indicates the traces that were correctly classified by the Fuzzy-BPMN miner after scoring of the classification process. The IEEE CIS Task Force on Process Mining contest committee [3] published on its website: a) 10

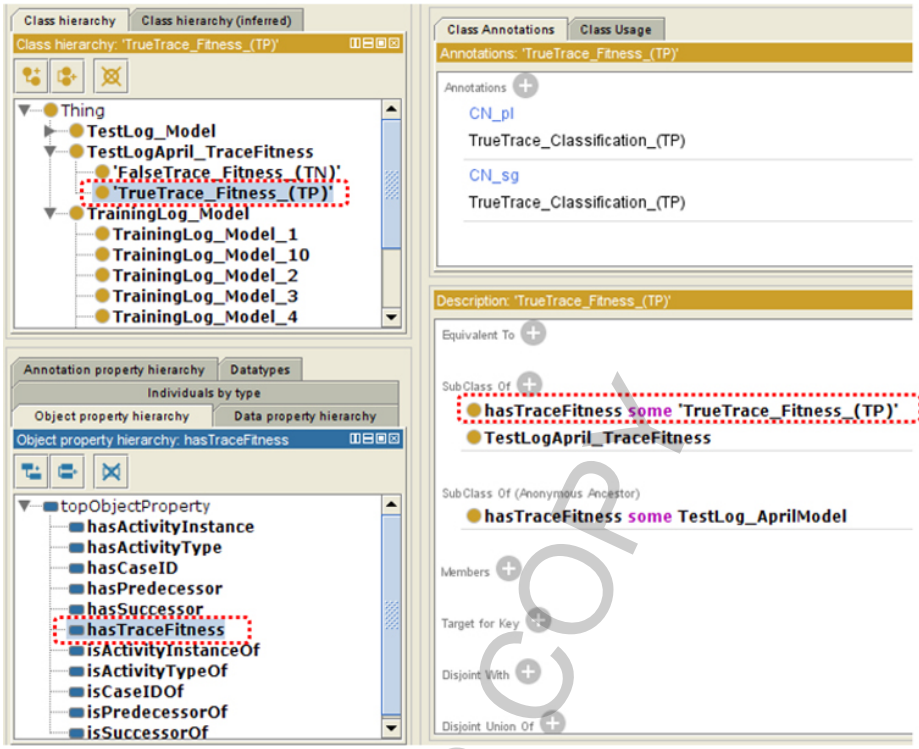

Fig. 10. Object property assertion (annotation) for the True trace classification.

evaluation logs, each of which contains 20 traces that were used to score the submissions, and b) 10 reference process models in BPMN format and notation that have been generated from the original data logs that were undisclosed. Indeed, the final result after scoring by the committee (panel of judges) shows that the Fuzzy-BPMN miner approach has correctly classified 171 out of 200 (85.5%) traces in the original process model.

5.1. *Enhancing the outcome of the fuzzy-BPMN miner through the semantic-based process mining approach*

The sematic-based analysis allows the meaning of the process elements to be enhanced through the use of property characteristics and classification of discoverable entities, to generate inference knowledge that are used to determine useful patterns (traces) and predict future outcomes. Indeed, this form of conceptualisation allows the analysis of the process instances at a more conceptual level. Perhaps, as mentioned earlier in Section 4, *COnts* is a set of concepts of (possibly different) ontologies of the set *Onts* (*COnts* $\subseteq$ *Onts)*. Definitively, the ontology *Ont* $\in$ *Onts* is a *formal explicit specification of a (shared) conceptualization* [34] which are exploited to represent the resulting models. As we noted earlier in the *Algorithm 1*, ontology is a quadruple $Ont = (C, R, I, A)$ which consists of different classes C and relations R between the classes [34]. Moreover, classes can be instantiated with a set of individuals I, and can also contain a set of axioms A which state facts. For example, what is true and fitting? (true positives) or what is true and not fitting? (true negatives) etc. within the process base. In view of that, as shown in Fig. 10 and as mapped in Fig. 11, we have used the "*hasTraceFitness*" object property to reference the sets of class from the test logs that has a "*TrueTrace_Classification_(TP)*" or "*FalseTrace_Classification_(TN)*".

More so, Let A be the set of all process executions or 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 the process library *libA*. For instance, we execute the Description Logic (DL) [40] queries below as a set of input parameters to output the set of traces for "TestLog_Apri_1" within the

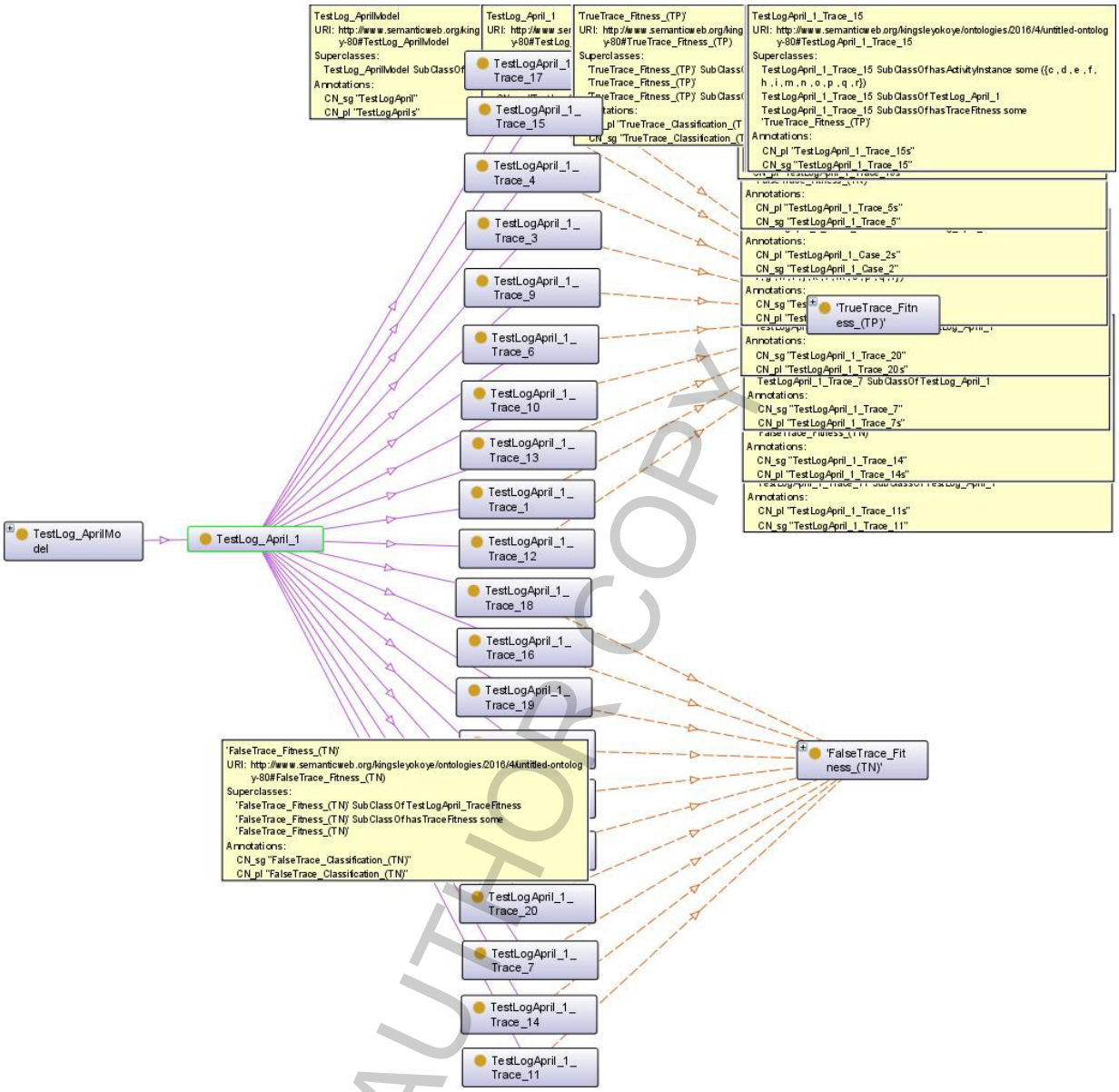

Fig. 11. Example of OntoGraph for the *TestLog_April_1* class with description of some of the semantic annotations.

model that has 'TrueTrace_Fitness_(TP)' and 'FalseTrace_Fitness_(TN)' in turn.

"TestLog_April_1 and hasTraceFitness some 'TrueTrace_Fitness_(TP)'"
"TestLog_April_1 and hasTraceFitness some 'FalseTrace_Fitness_(TN)'"

The results of computing the input and output parameters are as shown in Figs 12 and 13 respectively.

Accordingly, for the application phase of the approach in this paper, we implement a semantic-based fuzzy mining application – the Semantic Fuzzy Miner (SFM). The application is developed for use in extraction and automated mining of the process parameters and the concepts defined within the ontology. The work makes use of the Eclipse Java runtime environment to create the methods and interface for loading the sets of parameters. And then applies the Ontology Web Language Application Programming Interface (OWL API) [41] to extract and load the inferred concepts ascertained within the ontology (i.e. the semantic model). The purpose for designing the appli-

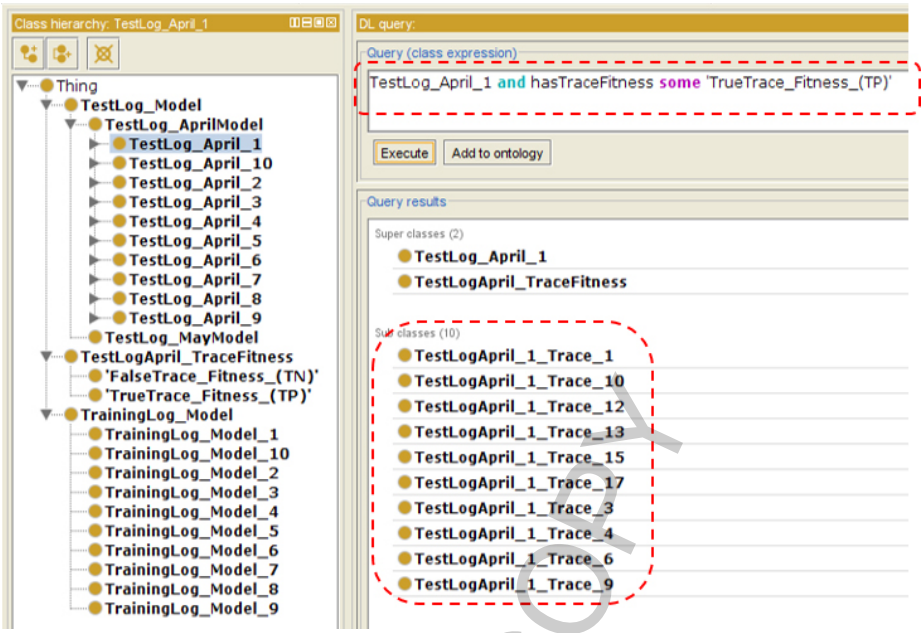

Fig. 12. Example of the *TrueTrace_Fitness_(TP)* classification for the *TestLog_April_1* with the correctly classified traces.

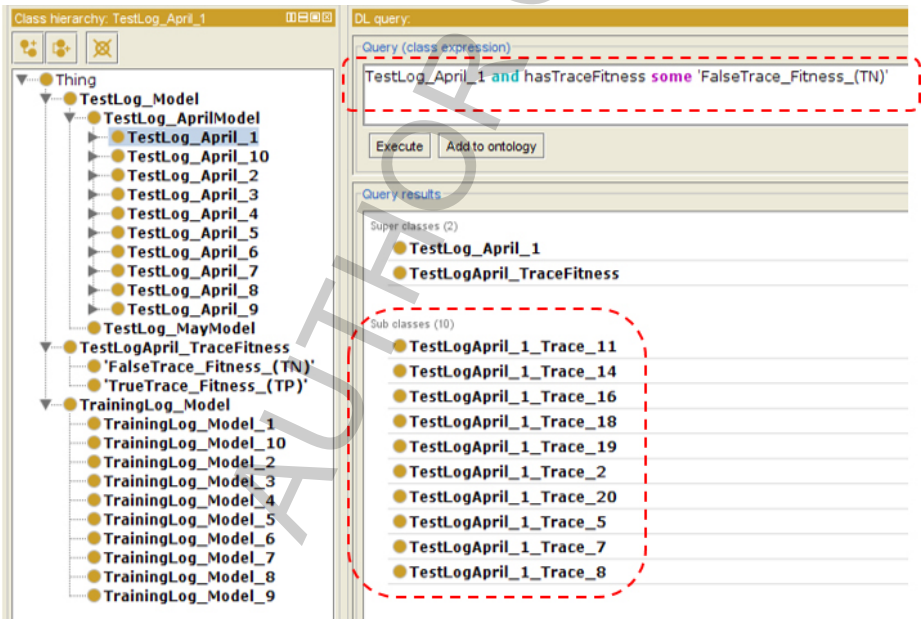

Fig. 13. Example of the *FalseTrace_Fitness_(TN)* classification for the *TestLog_April_1* with the correctly classified traces.

cation is to match the questions one would like to answer about attributes and relationships the process elements share amongst themselves by linking to the referenced concepts (classes) within the ontology. Figure 14 shows the application interface the work has developed for querying and retrieving the sets of data within the defined model.

6. Experimentation outcomes & results analysis

The semantic fuzzy mining approach and its application references a number of different OWL ontologies (e.g. the training model ontology, test set ontology, traceFitness Classification ontology etc.) which were generated for the experiment. For each ontol-

Table 2
Trace fitness and classifications for the test event logs (*test_log_april_1* to *test_log_april_10*) using the Semantic-Fuzzy Mining approach

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
Trace_1	TP *	TN *	TP *	TN *	TN *	TN *	TP *	TP *	TP *	TP *
Trace_2	TN *	TN *	TP *	TP *	TP *	TP *	TP *	TN *	TP *	TP *
Trace_3	TP *	TP *	TP *	TN *	TN *	TN *	TN *	TP *	TP *	TN *
Trace_4	TP *	TP *	TN *	TP *	TN *	TP *	TN *	TP *	TP *	TN *
Trace_5	TN *	TN *	TN *	TP *	TN *	TP *	TN *	TP *	TP *	TN *
Trace_6	TP *	TN *	TN *	TP *	TN *	TP *	TP *	TN *	TN *	TP *
Trace_7	TN *	TP *	TP *	TN *	TN *	TP *	TN *	TP *	TN *	TN *
Trace_8	TN *	TP *	TP *	TP *	TN *	TN *	TP *	TP *	TP *	TP *
Trace_9	TP *	TN *	TP *	TN *	TP *	TN *	TP *	TP *	TN *	TP *
Trace_10	TP *	TN *	TP *	TN *	TN *	TN *	TP *	TP *	TP *	TP *
Trace_11	TN *	TP *	TP *	TP *	TP *	TN *	TN *	TN *	TN *	TP *
Trace_12	TP *	TN *	TN *	TP *	TP *	TP *	TP *	TN *	TP *	TN *
Trace_13	TP *	TP *	TN *	TN *	TP *	TN *	TN *	TN *	TN *	TP *
Trace_14	TN *	TP *	TN *	TN *	TN *	TN *	TN *	TP *	TN *	TP *
Trace_15	TP *	TN *	TN *	TN *	TP *	TP *	TN *	TN *	TN *	TN *
Trace_16	TN *	TN *	TN *	TP *	TP *	TN *	TN *	TN *	TP *	TN *
Trace_17	TP *	TP *	TP *	TP *	TP *	TP *	TP *	TN *	TN *	TP *
Trace_18	TN *	TP *	TN *	TN *	TP *	TP *	TP *	TN *	TN *	TN *
Trace_19	TN *	TP *	TP *	TP *	TN *	TP *	TP *	TP *	TN *	TN *
Trace_20	TN *	TN *	TN *	TN *	TP *	TN *	TN *	TN *	TP *	TN *
True Positive (TP):	10	10	10	10	10	10	10	10	10	10
False Positive (FP):	0	0	0	0	0	0	0	0	0	0
True Negative (TN):	10	10	10	10	10	10	10	10	10	10
False Negative (FN):	0	0	0	0	0	0	0	0	0	0
Number of traces correctly classified	20	20	20	20	20	20	20	20	20	20

The cells colours indicates if the specified trace has been classified as true positives (TP) or true negatives (TN). All the cells with gold sign * indicates traces that were correctly classified by the Semantic-Fuzzy Miner with total of 200 traces out of 200.

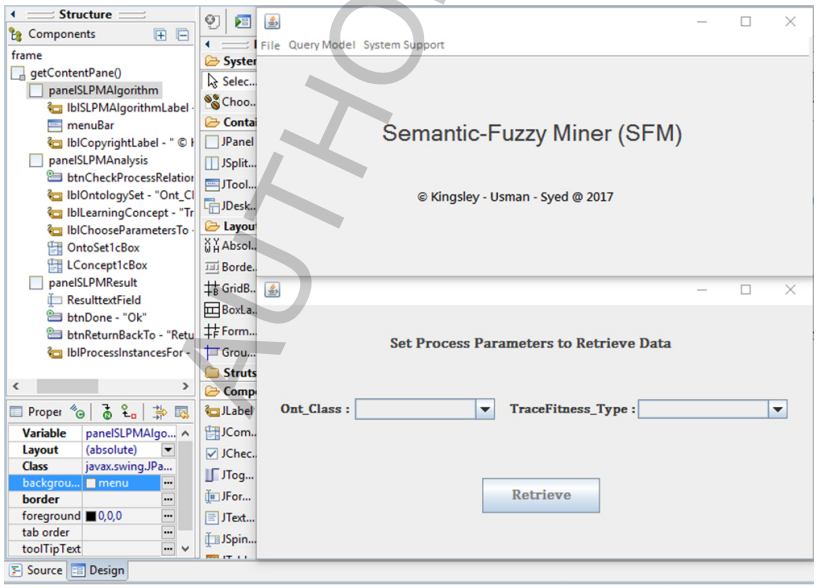

Fig. 14. Application interface for the Semantic-Fuzzy Miner (SFM).

ogy, all concepts in their turn were considered by the reasoner and were checked for consistency using the process parameters defined within the resulting semantic model. Based on the behavioural characteristics of the provided datasets [3], a cross validation design 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 response, and the classifier tested on the resulting individuals (traces) by assessing its performance with respect to correctly classified traces produced by the reasoner. For each result of the classifier for the test set, the replayable (true positives) and non-replayable (true negatives) traces were learned. The outcome of the experiments with regards to the discovered models and the classification of the corresponding individual traces occurring in each test set are as reported in Table 2.

From the Table 2, it is important to note that for every run set of parameters, the commission error, i.e. *false positives (FP)* and *false negatives (FN)* was null, hence equal to 0. This means that the classifier did not make critical mistakes. For example, settings where a trace is deemed to be an instance of a class while it really is an instance of another class. Also, at the same time, it is important to note that the trace accuracy rates was very high, i.e. for the *true positives (TP)* and *true negatives (TN)*, and were consistently observed for all the test sets. Significantly, such method of quality and accurate classification process for the individual traces within the process base can be utilized as a way of performing useful information retrieval and query answering in a more efficient, yet effective way compared to other standard logical procedures. Practically, it is shown that the classification performance is not only comparable to the outcome of just a reasoner, but also a classifier that is able to induce new knowledge based on previously unobserved behaviours. Indeed, an increase in the predictive accuracy was achieved by means of the semantic-based annotations and conceptual analysis, and as such, the technique can be exploited for predicting or suggesting missing information (metadata) about process elements especially when completing large ontology-based systems. Besides, the new knowledge and semantic assertions can be used by the process owners, process analysts or IT experts to address and answer real time questions about their processes in view.

6.1. Qualitative evaluation and impact of the semantic fuzzy mining approach and outcomes

Evidence from the study design and experimentation shows that the semantic-based approach sparks methods that highly influence and support:

(i) The application of process mining techniques to any domain process (e.g. case study of learning process), and

(ii) Provision of real time semantic knowledge and understanding about processes which are useful towards the development of process mining algorithms that are more intelligent with high level of effective conceptual reasoning capabilities.

In our experimentations, we observe that ontologies help in harmonizing the various process elements that are found within the process models and data sets, and also, that semantic annotations and reasoning helps to add useful conceptual knowledge to the mining results. We address the typical real time learning questions as identified in Subsection 4.1 to show in details how the semantic-based approach is implemented and relevant in the context of process mining and analysis. The main components realised as a result of implementing the semantic-based learning process mining approach is summarised as follows:

- *Event Logs* – to show how process mining can be applied to improve the informative values of learning process data.
- *Process Model* – describe how improved process models can be derived from the large volume of event data logs found within the domain processes e.g. learning process.
- Annotation – describe how semantic descriptions (annotation) of the deployed model can help enrich the result of the process mining and outcomes through discovering of new knowledge about the domain process and its elements.
- Ontology – use of ontologies with effective semantic reasoning to lift process mining analysis from the syntactic level to a more conceptual level.
- Semantic Learning Process Mining Algorithm – that reveals how references to ontologies and effective raising of process analysis from the syntactic to semantic level enables real time viewpoints on the learning process domain and models, which helps to address the problem of analysing the learning process data sets based on concepts and to answer questions about relationships the learning elements (process instances) share amongst themselves within the learning knowledge-base.

Principally, we utilized the case study of the learning process to pilot the structure of event logs and process models to determine various semantic viewpoints on information (metadata) related to how a process have been executed in the past and to discover real process

Table 3
The Semantic-Fuzzy miner and its application properties evaluated against existing benchmark algorithm

	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 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)

flows within the process knowledge-base. Moreover, the semantic-based modelling and analysis provides us with the opportunity to develop algorithms which are capable of analysing the resulting process model through explicit specification of conceptualisation to identify appropriate domain semantics and relationships among the process elements and/or concepts as well how we make use of the reasoner to check for consistency of all the defined concepts within the model. Clearly, with the use case example of the learning process, our focus is based on the learners interaction within the learning execution environment, to identify useful characteristics that describes the presented behaviours/patterns within the deployed model, and then respond by making decisions based on the semantic process descriptions and reasoning capabilities in order to improve the entire process analysis and engagement. Besides, the integration of the different ontologies, conceptual model references, and reasoner makes it possible to define more universal analysis questions and automatically find the answer for those questions. Furthermore, because the analysis is performed at the conceptual level (e.g. as shown in Figs 11 and 12) it is closer to human understanding and the addition of new elements in the ontologies or changes to the attribute labels does not necessarily require updating the analysis questions. For instance, the process to determine the individuals (learners) that have successfully completed the research process, one could easily include more activity concepts or attributes without requiring updating the question. The question remains the same and applicable to the class of individuals that fulfils the universal or existential restrictions by way of the object property assertions and semantic descriptions. This brings much more flexibility to the entire process and analysis.

From all evidence, the semantic-based approach as described in this paper is a significant contribution to the state of the art, where many existing process mining techniques requires some form of reconstruction to bring process analysis to a conceptual level or in many cases lacks the ability to identify and make use of semantics across different process domains. Moreover, to the best of our knowledge, this form of conceptualisation has not previously been applied in the area of learning process domain. In summary, this paper proves and show that a system which is formally encoded with semantic labelling, ontology and reasoning capabilities as presented in our design framework and the proposed semantic-based fuzzy mining approach, has the potential to assist in process mining tasks by allowing the analysis of the different process elements at a much more conceptual level.

In Table 3 we have carefully analysed the influence of the proposed semantic fuzzy mining approach compared to other existing benchmark algorithm for semantic process mining. Noticeably, as described in our approach and the analysis in Table 3, the use of ontologies, semantic reasoning/assertions, and references to labels in event logs and process models makes it possible to define a more easy and yet effective way to analyse real-time questions about the process elements and the relationships they share between themselves, and to automatically find the answer for those questions – as previously shown in Figs 12–14. Indeed, the semantic-fuzzy miner differs as well as combine interesting properties with existing, if not the only, semantic process mining algorithm (the Semantic LTL Checker) [5] currently in literature as presented in Table 3.

Firstly, the semantic fuzzy mining approach based on these critical elements proves to be more accurate and robust than conventional mining techniques because the approach also take the semantic perspectives of event logs and process models into account. More-

over, as opposed to the existing semantic LTL checker which only considers and takes event Logs concepts as input to parameters of Linear Temporal Logic (LTL) formulae to analyse the process, the semantic fuzzy mining approach also takes the process models as input. Besides, because these models are automatically generated from the actual event logs of the process, the system tends not to unnecessarily lose or leave out important information or missing data.

Secondly, even though both approaches makes use of ontologies, a major difference between the existing semantic LTL checker algorithm and our proposed approach is the fact that ontologies are defined in Web Service Modelling Language (WSML) format with the semantic LTL checker, while in our approach ontologies are defined using OWL and SWRL format. Perhaps, whilst there are limitations with WSML ontologies with respect to the exchange of syntax over the web, OWL ontologies aims to bring the expressive and reasoning power of description logic to the semantic web. Thus, it's the state of the art *logical layer* upon which semantic architectures are currently built in literature [42]. In fact, OWL ontologies allows one to specify far more about the *properties* and *classes* which are defined within a process domain knowledge-base. In essence, they are designed to represent rich and complex knowledge about *things* (superclass), *groups of things* (subclasses) and *relations between things* (i.e. relationships between the classes and individuals). Therefore, the OWL ontology as utilized in this paper is designed for use by applications that need to process the content of information instead of just presenting information to humans, in other words, machine-understandable rather than just machine-readable.

Thirdly, from a *reasoning* point of view, the semantic LTL checker uses the WSMLReasoner to perform a more complex inferences that are beyond subsumption reasoning by only benefiting from the inclusion of semantic annotations, whilst on the other hand, the semantic fuzzy mining approach is integrated with Pellet reasoner which typically in addition to semantic annotations has been proven to incorporate optimizations for nominals, conjunctive query answering, and incremental reasoning capabilities that supports process descriptions and logic, i.e., class assertions and object/data property assertions, and are indeed shown to be very effective in reasoning particularly at a more conceptual level.

Lastly, the semantic LTL checker and the proposed Semantic Fuzzy miner both has 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 proposed 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.

Table 4
Performance measures formula for the classifiers

Classifier name	Formula
tp-rate	tp/p
fp-rate	fp/n
Error	$(fp + fn)/N$
Accuracy	$(tp + tn)/N$
Precision	tp/p'
Recall	tp/p
F1 score	$(2 \times \text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall})$

6.2. Quantitative analysis and evaluation of the semantic fuzzy mining approach

In this section, we present how the study quantitatively assess and validate the accuracy and performance of the classification outcomes for our approach. Fore mostly, it is important to note that to quantitatively measure the quality of process mining algorithms or techniques, it is essential that one must first focus on the accuracy of the classification results (i.e. the outcome of the classifier over the given data set) rather than focusing on the *seen* (observed) process instances. The quality of analysis of the classification result is useful to further predict good classification for *unseen* (unobserved) instances. Henceforth, given the data set consisting of N instances we know for each instance: what the actual class is and what the predicted class is (often expressed as *confusion matrix* [2]). The confusion matrix considers a given set of data with only two classes: Positive (+) and Negative (−) values [2] and are measured using some performance formula for the classifiers as shown in Table 4.

Where:

– *tp-rate* (true positive rate) = *tp/p* also known as *hit rate* measures the proportion of positive instances that are indeed classified as positive.

- *fp-rate* (false positive rate) = *fp*/*n* also known as *false alarm rate* measures the proportion of negative instances wrongly classified as positive.
- *Error* = (*fp* + *fn*)/*N* is defined as the proportion of instances misclassified.
- *Accuracy* = (*tp* + *tn*)/*N* measures the fraction of instances on the transverse of the confusion matrix, i.e., the proportion of instances correctly classified.
- *Precision* = tp/p' where *tp* is the number of traces that have been retrieved and also should have been retrieved, and p' the number of traces that have been retrieved based on some search query.
- *Recall* = *tp*/*p* where *tp* is as defined in *Precision* and *p* is the number of traces that should have been retrieved based on some search query.
- *F1 Score* = (2 × *precision* × *recall*)/(*precision* + *recall*) takes the harmonic mean of *precision* and *recall*, i.e. if either the *precision* or *recall* is really poor, then the *F1 Score* is close to or equals to 0. On the other hand, if the *precision* and *recall* are really good, then the *F1 Score* is close to or equals to 1.

Indeed, *If* $N = tp + fn + fp + tn$ is the total number of instances within the data set, *Then* based on the definitive expression, it is easy to determine the values of the class Positive (+) and Negative (−) classified by the classifier. For example, the number of instances that are actually positive, i.e., $p = tp + fn$ can perhaps be realized. On the other hand, the number of instances that are actually negative, $n = tn + fp$ can also be determined. Also, $p' = fp + tp$ is the number of instances that are classified as positive by the classifier, while n' = *fn* + *tn* is the number of instances that are classified as negative by the classifier. To this end, the formulas in Table 4 are construed. According to Van der Aalst [2] the number of *unseen* instances is potentially vast (if not infinite) and therefore an estimate needs to be computed on a test set which is commonly known as *cross-validation* i.e. where the data set is split into a *training set* and a *test set*.

Cross-validation [2] is one of the performance indicator approach that can be used to evaluate process mining algorithms. The event logs are split into a *training log* and a *test log* and the employed mining technique tends to learn process models from a major part of the event log (i.e. the training log) and the individual cases that forms the event log (i.e. the test log). Hence, the *training log* is used to learn a process model, whereas the *test log* is used to evaluate the discovered model based on unseen cases (or traces). With the cross-validation approach, the test log is replayed using the model that is discovered from the training log and can be repeated k times when *k-folds* are used (i.e. the evet log is split into k equal parts, e.g. $k = 10$, and then k test are done. For each test, one of the subsets serves as a test log whereas the other $k - 1$ subsets serves collectively as the training log. The main idea of cross-validation is to quantitatively compare the quality of the discovered model with respect to the test log containing *actual behaviour* (fitting traces) and the quality of the discovered model with respect to a test log containing *random behaviour* (artificially generated negative events). Superlatively, it is expected that the model scores much better on the log containing *actual behaviours* than on the log containing *random behaviour*. Therefore, the experimentations carried out in this paper measures to what extent the scoring of the discovered model when encoded with real semantics (formal domain knowledge) about the process elements helps lift the analysis of the process mining techniques from the syntactic level to a more conceptual level. Indeed, the main objective is to formally encode semantic knowledge to the discovered models to help identify and enhance the fitness of the individual traces as well as the quality of the model and its analysis through semantic assertions (process descriptions) and automated computing of the classes, namely: Positive (+) and Negative (−) values by the classifier.

Therefore, in order to assess performances of the semantic-based approach (i.e. the Semantic-Fuzzy Miner) being able to correctly classify and analyse the individual traces within the models:

- Given a trace (t) representing real process behaviour (i.e. *true positives* or allowed traces) or
- Trace (t) representing a behaviour not related to the process (*true negatives* or *disallowed* traces) in the given sets of data.

The work conducted experimentations on the results of the data as provided in [3]. The available datasets stand for the same ones we used in this paper and also in participating in the contest [8]. Characteristics of the datasets are explained in the objectives [3] of the contest, which is to discover process models from a *training event log* representing 10 different real time business process executions, and a set of *test event logs* provided for evaluation of the employed process mining approach. Each of the test event logs (*test_log_april_1 to test_log_april_10*) represents part of the original model with complete total of 20 traces for each of the individual test logs, and are characterized by having 10 traces that can be replayed (*allowed*) and 10 traces that

Table 5
Experimental results from the Semantic-Fuzzy miner and other benchmark process mining techniques

	Inductive miner	Decomposition	DrFurby	Fuzzy-BPMN	Semantic-fuzzy
Model_1	100	100	100	100	100
Model_2	100	100	100	80	100
Model_3	60	95	100	60	100
Model_4	100	100	100	85	100
Model_5	95	100	100	100	100
Model_6	85	95	100	55	100
Model_7	100	100	100	95	100
Model_8	75	70	95	85	100
Model_9	100	100	100	100	100
Model_10	100	100	100	95	100
Ave. Mean – PCC (%)	91.5	96	99.5	85.5	100
No. of traces correctly classified	183	192	199	171	200

cannot be replayed (*disallowed*) by the model. Therefore, a wide variety of problems is represented. In this paper, we have used the test event logs with complete total of 200 traces to validate our approach.

Accordingly, the final outcome of the experimentation and cross-validation were carried out on other existing benchmark algorithms which includes namely: Inductive Miner and Decomposition [43], DrFurby Classifier [44], Heuristic Alpha + Miner [45] Fuzzy-BPMN miner [8] etc., that uses the same event logs in [3] to discover process models and provides replaying semantics for the individual traces within the test log. We used the standard Percent of Correct Classification (PCC) [24] to assess the performance of the classifiers.

Henceforth, the standard Percent of Correct Classification [24] for the *test log* is defined as follows:

$$Log\_PCC = (number\ of\ correctly\ classified\ traces)/(total\ number\ of\ traces) \times 100$$

For example, for the *training_model_7* as previously shown in Table 1, the standard Percent of Correct Classification (PCC) for the April test log for the initial result from the process discovery contest (i.e. Fuzzy-BPMN miner) [8] approach is determined as follows:

$$\begin{aligned} \text{Training\_Model\_7}(PCC) &= (19)/(20) \times 100 \\ &= 0.95 \times 100 \\ &= 95\% \end{aligned}$$

On the other hand, the standard Percent of Correct Classification (PCC) for the *training_model_7* as shown in Table 2 for the Semantic-Fuzzy miner approach is as follows:

$$\begin{aligned} \text{Training\_Model\_7}(PCC) &= (20)/(20) \times 100 \\ &= 1 \times 100 \\ &= 100\% \end{aligned}$$

Using the logical formula, i.e., standard Percent of Correct Classification [24] we measure and analyse in Table 5 the sophistication of the other existing benchmark algorithms [43–45] as well as the initial result of the Fuzzy-BPMN miner [8], to weigh up the proposed Semantic-Fuzzy mining approach and experimental results. The outcome from our approach and the different benchmark techniques and classification results are as shown in Table 5.

From the experimental results in Table 5, and the plots in the charts – Figs 15–17, we observe that the Semantic-Fuzzy miner considerably outperform respectively the Inductive miner and Fuzzy-BPMN miner, even though, the two algorithms Decomposition and DrFurby stands for the state of the art classifiers amongst the existing process mining techniques when compared to analysis of the classifications results and outcome. Additionally, the semantic-based approach has shown an error free performance measure by using the classifier formulas, i.e. $Error = (fp + fn)/N$ where $fp = 0$ and $fn = 0$, thus, $Error = (0 + 0)/200 = 0$. Also, the approach has shown using the $Accuracy = (tp + tn)/N$ where $tp = 100$ and $tn = 100$, thus, $Accuracy = (100 + 100)/200 = 1$. Clearly, going by the *F1 Score* $= 1$, the *Precision* and *Recall* of the Semantic-Fuzzy Miner classifications are indeed efficient.

7. Discussion

Indeed, the use of ontologies ($Ont \in Onts$) and the relations (R) between the concepts ($COnts$) de-

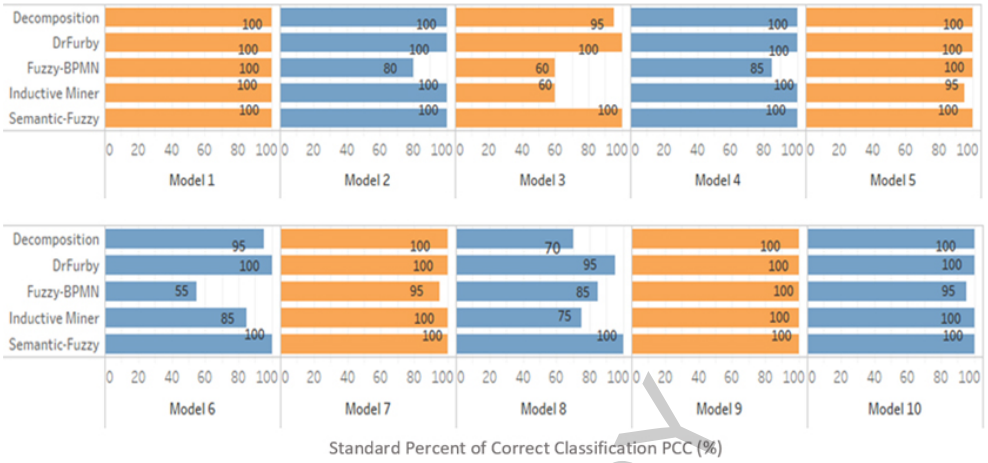

Fig. 15. Chart showing the sum of correctly classified traces by the various algorithms for each Model 1 to 10.

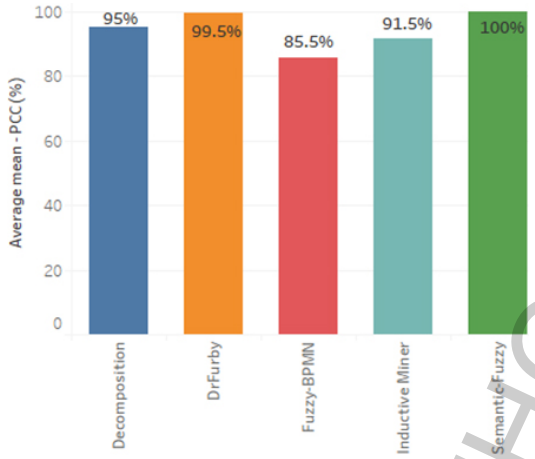

Fig. 16. Sum of average mean – PCC (%) for each of the algorithms.

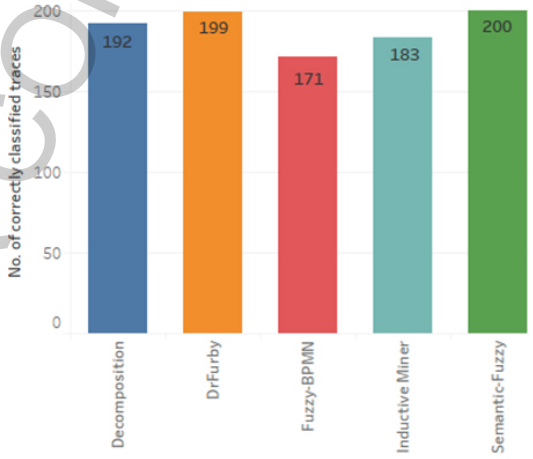

Fig. 17. Total number of traces correctly classified by each algorithm.

fined in the ontologies were beneficial to aggregate tasks and compute formally the structure of the process models including the several abstraction levels [4,34]. The main idea is that for any semantic-based process mining approach, these aspects of aggregating the task [19], computing the hierarchy of the process models (Lehmann and Hitzler [46]) or prediction of the model behaviours (Trstenjaka and Donkob [47] should not only be *machine-readable*, but also *machine-understandable,* which means that the process models are either semantically annotated or already in a form which allows the computer to infer new facts or perform the process querying (Polyvyanyy et al. [48]) by using the underlying ontology such as the one introduced in this paper using the case study of the research learning process and data from the IEEE Task Force on Process Mining. Essentially, the purpose of the semantic annotation process is to seek the equivalence between *the concepts of the model* (i.e. the fuzzy models derived by applying the fuzzy miner algorithm on the event data sets and the *concepts of the defined domain ontology*. Besides, the fuzzy logic [25,29] has since been introduced as an extension of the Boolean logic which allows a proposal to be in another state as true or false [30] by enabling the modelling of uncertainty and imprecision that often characterize the human representations of knowledge. Perhaps, we observe that by semantically integrating the fuzzy system with concepts within a defined ontology, they can make decisions like humans do (for instance, the learning question that allows us to determine which entities within the learning model that are classified as successful or incomplete learners) by offering solutions that bear characteristics of “intelligence” which is usually attributed to humans only. Moreover, this has been con-

sidered broadly as a specific feature of *Computational Intelligence* rather than literally an area of *Artificial intelligence* notion.

Currently, the fuzzy logic has become mature and is being used in different areas of application as we have applied in the process carried out in this paper to support the semantic-fuzzy mining approach. The study approach and experimental interest is particularly focused on using the fuzzy logic to represent imprecise and uncertain (complex) data (e.g. the data sets in [3]) for semantic labelling (annotation), representation (ontology), and reasoning (reasoner). Accordingly, fuzzy algorithms are applied with the goal to show understandable models for very unstructured and flexible processes [31]. Moreover, one of the main strengths of the fuzzy models is that they are conceived to be easily adaptable, in essence, *extendible*. We have provided the semantic-fuzzy miner as a tool which can be exploited to create models that can be understood easily while providing implicit information on the extensible set of parameters (concepts) used to determine and analyse process models at a more conceptual level as explained in Sections 4 and 5 in this paper. The semantic-fuzzy mining technique establishes a direct connection between the discovered process models and the actual low-level event log information about the process elements in reality to analyse the available data at a different level of abstraction, hence, conceptualisation.

In turn, as a collection of *concepts* and *predicates*, the system being ontology-based has the ability to perform logic reasoning and bridge the underlying relations beneath the event logs and the process models discovered using traditional process mining with rich semantics. In essence, whenever an *inference* (semantic reasoning) is made, a generalized associations of the process elements is created, and thus, provides consistency inference for those predicates by tuning the unlabelled data associated with the fuzzy models into one (i.e. semantic model) that have the best consistency by making use of the prior knowledge about the data.

Therefore, the main benefits of the semantic-based fuzzy mining approach described in this paper can be summarised in two forms:

(i) Encoding knowledge about specific process domains, and
(ii) Advanced analysis and reasoning of processes at a much more conceptual level.

Indeed, the semantic-based fuzzy mining approach as described in this paper can be regarded as a fusion theory that is based on the fuzzy logics and devoted to represent and analyse information in a qualitative and yet quantitative manner.

8. Conclusion

The main focus for designing the semantic-based process mining *Algorithms*, the *Semantic-Fuzzy Miner,* and the proposed Framework which we refer to as *2-dimensional Rhombus approach* is to extract, semantically prepare, and transform event log about domain processes into mining executable formats that allows for an improved process analysis of the captured event data logs through conceptualization method. We build a semantic model to represent the deployed process models as a result of applying the fuzzy mining algorithm on the sets of data used for the work in this paper. The primary aim is to provide platform that allows us to semantically represent the model and then carry out effective reasoning on the resulting models and ontologies in order to infer and identify individual traces that makes up the process as well as answer questions about relationships the process elements share amongst themselves within the knowledge-base. We have used the cases study of the learning process to illustrate this approach. Accordingly, the technique makes use of semantic annotations to link elements in the event logs with concepts that they represent in an ontology and through semantic reasoning allows us to expound and enhance the process analysis of the sets of data from the syntactic level to a more conceptual level that can easily be grasp by the process owners, process analysts or IT experts. By referring to ontologies, the approach provides us with the capability to determine the relationships the process instances share within the knowledge-base and then infer and discover unseen (unobserved) patterns automatically by means of semantic reasoning. The purpose for designing such an intelligent system is to perform semantic-based process analysis of the available data capable of providing real world answers that are closer to human understanding. In this paper, we also describe the various components of the proposed system in details and explain how the study have integrated the main building blocks (semantic annotation, ontology, and reasoner) to support the design and development of the proposed semantic-based process mining algorithm as well as its formalization. Finally, the paper looks at the level of impact and implications of the semantic-based approach, the discovered process models, validation of the classification results and its influence compared to other existing benchmark algorithms within the field of process mining.

References

[1] D. Dou, H. Wang and H. Liu, Semantic data mining: A survey

of ontology-based approaches, *9th IEEE Int Conference on Semantic Computing* (2015), 244–251.
[2] W.M.P. Van der Aalst, Process mining: Data science in action, Springer, 2016.
[3] 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.
[4] 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, *Proc of 2016 IEEE International Conference on Big Data (Big Data)* (2016), Washington, DC, 3529–3538.
[5] A.K.A de Medeiros, W.M.P van der Aalst and C. Pedrinaci, Semantic process mining tools: Core building blocks, *In ECIS* (June 2008), Galway, Ireland, 1953–1964.
[6] A.K.A de Medeiros and W.M.P. Van der Aalst, Process mining towards semantics, *Advances in Web Semantics*, Lecture Notes in Computer Science **4891** (2009), 35–80.
[7] 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: *Proceedings of the 8th Int Conference on Soft Computing and Pattern Recognition*, (SoCPaR 2016), A. Abraham, A.K. Cherukuri, A.M. Madureira and A.K. Muda, eds, Advances in Intelligent Systems and Computing book series (AISC, volume 614), Springer International Publishing AG, 2018, pp. 622–633.
[8] K. Okoye, A.R.H Tawil, U. Naeem and E. Lamine, Fuzzy-BPMN miner approach – Process Discovery Contest @ BPM 2016, Technical Report Submission, *IEEE CIS Task Force on Process Mining Discovery Contest* [1st Edition] held during the BPI workshop that is co-located with the BPM 2016 conference, Rio de Janeiro, Brazil (18th September 2016).
[9] R.G.L Miani and E.R. Hruschka Junior, Exploring association rules in a large growing knowledge base, *Int J of Comp Info Syst and Ind Mangt Apps*, ISSN 2150-7988, **7** (2015), 106–114.
[10] A. Hicheur-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, *4th Int Symp on Data-Driven Process Discovery & Analysis* (2014), Italy, 150–161.
[11] ProM Tool. Available at http://www.processmining.org/prom/start.
[12] A.J.M.M. Weijters, W.M.P. Van der Aalst and A.K.A. de Medeiros, Process mining with the heuristics miner-algorithm, Tech Report, EUT, Eindhoven, BETA Working Paper Series, WP 166, 2006.
[13] W. Jareevongpiboon and P. Janecek, Ontological approach to enhance results of business process mining and analysis, *Journal of Business Process Management* **19**(3) (2013), 459–476.
[14] OWL Web Ontology Language. http://www.w3.org/TR/owl-ref/.
[15] SWRL: A Semantic Web Rule Language http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/.
[16] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
[17] M. Allahyari, K.J. Kochut and M. Janik, Ontology-based text classification into dynamically defined topics, in: *IEEE International Conference on Semantic Computing (ICSC)* (2014), 273–278.
[18] N. Balcan, A. Blum and Y. Mansour, Exploiting ontology structures and unlabeled data for learning, in: *Proceedings of the 30th Int Conference on Machine Learning* (2013), 1112–1120.
[19] C. d'Amato, N. Fanizzi and F. Esposito, Query answering and ontology population: An inductive approach, in: *Proc of the 5th Euro Semantic Web Conference*, S. Bechhofer, M. Hauswirth, J. Hoffmann and M. Koubarakis, eds, ESWC2008. Vol. 5021 of LNCS, Springer, 2008, pp. 288–302.
[20] K. Okoye, A.R.H. Tawil, U. Naeem and E. Lamine, A semantic reasoning method towards ontological model for automated learning analysis, In: *Advances in Nature and Biologically Inspired Computing*, N. Pillay, A. Engelbrecht, A. Abraham, M. du Plessis, V. Snášel and A. Muda, eds, Advances in Intelligent Systems and Computing, 2016, pp. 49–60.
[21] K. Okoye, A.R.H. Tawil, U. Naeem and E. Lamine, Discovery and enhancement of learning model analysis through semantic process mining, *Int Journal of Computer Info Systems and Industrial Management Applications*, ISSN 2150-7988, **8** (2016), 93–114.
[22] M.H.A. Elhebir and A. Abraham, A novel ensemble approach to enhance the performance of web server logs classification, *Int Journal of Computer Information Systems and Industrial Management Applications*, ISSN 2150-7988, **7** (2015), 189–195.
[23] K. Baati, T.M. Hamdani, A.M. Alimi and A. Abraham, A modified naïve possibilistic classifier for numerical data, in: *Proceedings of the 16th International Conference on Intelligent Systems Design and Applications*, Springer (2016).
[24] K. Baati, T.M. Hamdani, A.M. Alimi and A. Abraham, Decision quality enhancement in minimum-based possibilistic classification for numerical data, in: *Proceedings of the 8th Int Conference on Soft Computing and Pattern Recognition* (SoCPaR 2016), A. Abraham, A.K. Cherukuri, A.M. Madureira and A.K. Muda, eds, Advances in Intelligent Systems and Computing Book Series (AISC, volume 614), Springer International Publishing AG, 2018, pp. 634–643.
[25] L. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems* (1978), 3–28.
[26] D. Dubois, H.M. Prade, H. Farreny, R. Martin-Clouaire and C. Testemale, Possibility theory: An approach to computerized processing of uncertainty, New York: Plenum press, 1988.
[27] B. Khaleghi, A. Khamis, F.O. Karray and S.N. Razavi, Multisensor data fusion: A review of the state-of-the-art, *Information Fusion* **14**(1) (2013), 8–44.
[28] K. Baati, T.M. Hamdani, A.M. Alimi and A. Abraham, A new possibilistic classifier for heart disease detection from heterogeneous medical data, *International Journal of Computer Science and Information Security* **14**(7) (2016), 443–450.
[29] L. Zadeh, Fuzzy sets, *Information and Control, Information Science* Vol. 4, W4, (1965), 338–353.
[30] S.M. Dammak, A. Jedidi and R. Bouaziz, Fuzzy semantic annotation of Web resources, *2014 World Symposium on Computer Applications & Research (WSCAR)*, Sousse (2014), 1–6.
[31] A. Rozinat, Process Mining: Conformance and Extension, PhD Thesis, Technische Universiteit Eindhoven, Eindhoven, the Netherlands, 2010.
[32] Disco Users Guide. Available at: https://fluxicon.com/disco/files/Disco-User-Guide.pdf.
[33] A. Rozinat, Top 5 Data Quality Problems for Process Mining, 2016. Available at: http://fluxicon.com/blog/2011/06/data-quality-process-mining/.
[34] T.R. Gruber, A translation approach to portable ontology specifications, *Journal of Knowledge Acquisition* **5**(2) (1993),

199–220.
[35] F. Lautenbacher, B. Bauer and S. Forg, Process mining for semantic business process modeling, *13th Enterprise Distributed Object Computing Conference Workshops*, Auckland (2009), 45–53.
[36] F. Lautenbacher, B. Bauer and C. Seitz, Semantic Business Process Modeling – Benefits and Capability, *AAAI Spring Symposium: AI Meets Business Rules and Process Management*, Stanford University, California, USA (26–28 March 2008).
[37] M. Born, F. Dörr and I. Weber, User-friendly semantic annotation in business process modeling, in: *WISE 2007 Workshops*, ser. LNCS, M. Weske, M.-S. Hacid and C. Godart, eds, no. 4832. Springer-Verlag, 2007, pp. 260–271.
[38] C.W. Günther and W. Van Der Aalst, Fuzzy mining: Adaptive process simplification based on multi-perspective metrics, in: *5th International Conference on Business Process Management (BPM'07)*, G. Alonso, P. Dadam and M. Rosemann, eds, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 328–343.
[39] W.M.P. Van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer (2011).
[40] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi and P.F. Patel-Schneider, Description logic handbook, Cambridge University Press, 2003.
[41] OWL API: Ontology Web Language Application Programming Interface http://owlapi.sourceforge.net/.
[42] F.A. Lisi, Building rules on top of ontologies for the semantic web with inductive logic programming, *Journal of Theory and Practice of Logic Programming* **8**(3) (2008), 271–300.
[43] R. Ghawi, Process Discovery using Inductive Miner and Decomposition, in CoRR abs/1610.07989 (2016) Technical Report Submission for the Process Discovery Contest @ BPM 2016 [1st Edition], Available at: https://arxiv.org/abs/1610.07989.
[44] E. Verbeek and F. Mannhardt, DrFurby Classifier: Process Discovery Contest @ BPM 2016, Technical Report Submission for the Process Discovery Contest @ BPM 2016 [1st Edition], Available at: http://www.win.tue.nl/∼hverbeek/wp-content/uploads/2016/05/drfurby12.pdf.
[45] M. Shteiner, L. Bodaker and A. Senderovich, Heuristic Alpha+ Miner (HAM): Process Discovery Contest 2016, Technical Report Submission for the Process Discovery Contest @ BPM 2016 [1st Edition], Available at: https://web.iem.technion.ac.il/images/ISE-TR-16-2.pdf.
[46] J. Lehmann and P. Hitzler, Concept learning in description logics using refinement operators, *Machine Learning* **78** (2010) 203–250.
[47] B. Trstenjaka and D. Donkob, Web prediction framework for college selection based on the hybrid case based reasoning model and expert's knowledge, *International Journal of Hybrid Intelligent Systems* **13**(3–4) (2016), 161–171.
[48] 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 Systems* **100** (2017), 41–56.