We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Ontology: Core Process Mining and Querying Enabling Tool

Kingsley Okoye, Syed Islam and Usman Naeem

Additional information is available at the end of the chapter

Abstract

Ontology permits the addition of semantics to process models derived from mining the various data stored in many information systems. The ontological schema enables for automated querying and inference of useful knowledge from the different domain processes. Indeed, such conceptualization methods particularly *ontologies for process management* which is currently allied to *semantic process mining* trails to combine process models with ontologies, and are increasingly gaining attention in recent years. In view of that, this chapter introduces an ontology-based mining approach that makes use of concepts within the extracted event logs about domain processes to propose a method which allows for effective querying and improved analysis of the resulting models through semantic labelling (annotation), semantic representation (ontology) and semantic reasoning (reasoner). The proposed method is a semantic-based process mining approach that is able to induce new knowledge based on previously unobserved behaviours, and a more intuitive and easy way to represent and query the datasets and the discovered models compared to other standard logical procedures. To this end, the study claims that it is possible to apply effective reasoning methods to make inferences over a process knowledge-base (e.g. the learning process) that leads to automated discovery of learning patterns and/or behaviour.

Keywords: ontologies, semantic annotation, semantic reasoning, process querying, process mining, event logs, process models

1. Introduction

Ontologies has been proven to be one of the essential tools used for semantic-based process mining. The schema is a useful technique towards improving information values of process

models and their 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 could be used to determine useful patterns as well as predict future outcomes.

Indeed, the ability to mine useful or worthwhile knowledge from readily extracted data in current information systems is a challenge, due to the exponential increase in volume of data that is continuously generated. Moreover, many of such organizations data collection systems and procedures for the process analysis is proving to be more and more complex. In consequence, this has spanned the need for a richer or advanced description of real-time processes that allows for flexible exploration of the large volumes of data targeted at improving the systems performance and of course the main business operations. Such process-related analysis means there is also need for techniques that are capable of extracting valuable information from the event logs and the resulting models about the real time processes in view.

More or less, most organization have invested in projects to model their various operational process. However, most of the derived process models are often unfitting, non-operational, or represents a form of reality that are pointed towards comprehensibility rather than covering the entire actual business process complexities. Perhaps, according to the works in Refs. [1–4] an accurate exploration or analysis of the extracted events log is capable of providing vital and valuable information with regards to the quality of support being offered for the so-called organizations and their information knowledge-base or system at large. For example, revealing the underlying relationships the process elements or resources share amongst themselves within the information knowledge-base.

Recently, the Process Mining [3] or yet still Process Querying [5] notion has become a valuable technique used to discover such kind of meaningful information from the event data logs and the derived process models. However, the study carried out in [6] observes that a shared challenge with most of the existing process mining techniques is that they depend on tags/labels in event logs information about the processes they represent, and therefore, to a certain extent are limited because they lack the abstraction level required from real world perspectives. This means that the techniques do not technically gain from the real knowledge (semantics) that describe the tags or labels in events log of the domain processes [6]. Practically, majority of the process mining techniques in literature are purely syntactic in nature, and to this effect are somewhat vague when confronted with unstructured data.

For that reason, this work explores the technological potentials and prospects in using ontology as a core process mining and querying enabling tool by pursuing to address such challenges posed by the lack of semantic information through provision of a method for formal structuring of the readily available datasets. In other words, the work in this chapter addresses the above challenges i.e. (i) lack of process mining or querying tools that supports semantic information retrieval, extraction and analysis, and (ii) mining of event logs and models at a much more conceptual levels as opposed to the syntactic nature or methods for process mining. The purpose is mainly as a way of providing formal structures for the datasets used for process mining and enhancement of the analysis and integration of the resulting process models. Such an ontology-based approach is significant because, indeed, it involves semantic

descriptions and/or reformulation of the meanings of the labels within the event logs and process models, as well as their comparisons for the purpose of improving the usefulness and performance of the entire domain processes in question particularly during the information retrieval, processing, and extraction process. In short, the propose approach in this chapter supports the augmentation of the informative values of the resulting models by semantically annotating the process elements with concepts they represent in real time, and linking them to an ontology in order to allow for analysis of the extracted data logs and models at a much more conceptual level.

In turn, the conceptual method of analysis provides an easy way to analyse the datasets (i.e. the event logs and models), and even more allows the meaning of the process elements to be enhanced through the use of property descriptions languages or syntax—such as the Ontology Web-Rule Language (OWL) [7] Semantic Web Rule Language (SWRL) [8], Description Logic (DL) queries [9], and classification of discoverable entities or taxonomy [4] in order to make available inference knowledge that could be utilized to determine useful patterns by means of the semantic reasoning aptitudes. On the other hand, the semantic modelling (*ontological representations*) and analysis techniques provide us with the opportunity to develop intelligent algorithms and tools which are capable of enhancing the resulting process models through explicit specification of the concepts (often referred to as *conceptualisation*) [5, 10, 11] in order to identify appropriate domain semantics and relationships amongst the process elements.

Finally, the work applies the proposed method on the case study of learning process domain to demonstrate the usefulness of the semantic-based approach. The study takes into consideration the different stages of process mining and its application—from the initial phase of collecting and transformation of the readily available event data to discovered process models, and then to semantically preparing the extracted models for further analysis and process querying at a more abstraction level. In essence, the chapter shows by using the case study of *Learning Process*—how the data from the various process domains can be extracted, semantically prepared, and transformed into mining executable formats to support the discovery, monitoring and enhancement of real-time processes through further semantic analysis of the discovered models. Indeed, the proposals and outcomes of the study shows that a system which is formally encoded with semantic labelling (annotation), semantic representation (ontology) and semantic reasoning (reasoner) has the capability to enhance process mining analysis and results from the syntactic level to a much more conceptual level.

Over the following section, the study looks at the ontological concepts and its main functions, and the describe how the work has utilised the schema to develop the proposed semantic-based process mining approach.

2. Ontologies

As a collection of *concepts* and *predicates*, ontology has the ability to perform logic reasoning and bridge the underlying challenges (semantic gaps) beneath event logs and models discovered especially through conventional process mining techniques with rich semantics.

To make the semantic knowledge available, ontologies are incorporated with the process models in order to pre-determine the model structure. Besides, the method also serves as a way of representing or bridging the distances between the labels within the process models and concepts in the defined ontologies.

Indeed, an ontological schema aims to transforms a process map into a *bipartite graph* (also referred to as Ontograph) to denote both the process models and its elements in a uniformed structure. So, whenever an inference (semantic reasoning) is made, a generalized associations (classification) of the process elements is created, and in consequence, infers the class hierarchies as well as performs a consistency check for those predicates. Besides, the sets of constraints (i.e. Object or Datatype property restrictions) driven by the ontology have the capacity to recognize inconsistent data and outputs particularly during the pre-processing stage, the algorithm executions, filtering or interpretation stage, and the results generation.

Several application and definition of the ontology term has been proposed in literature which most of the time concerns the varied domains of interest. According to Ref. [12] the term ontology is borrowed from the philosophy field which is concerned with being or existence study. The author mentions that in the context of computer and information science, ontology symbolizes as an artefact that is designed to model any domain knowledge of interest.

Even more, Ref. [13] refers to the ontological term as a formal explicit specification of a conceptualisation, and till date has been the most widely cited definition of ontology in the computer field. The definition means that ontology is able to explicitly define (i.e. specifies) concepts and relationships that are pertinent for modelling any domain of interest. Moreover, such specification can be represented in the form of *Classes, Relations, Constraints* and *Rules* to provide more meanings to use of the different expressions or relations. So therefore, ontology performs the following three functions, namely: *Formal—Explicitness—Conceptualisation*—to provide hierarchical structures and representation of information or knowledge.

In principle, ontology helps in description of the various concepts as well as the associations that holds amongst those concepts within a process domain. Hence, ontologies range from taxonomies, classifications, database schemas to fully axiomatized theories which state facts. Moreover, ontologies are nowadays an essential tool to a lot of systems or algorithms that are used for information retrieval and extraction, information management and integration of systems, scientific-knowledge portals, including e-commerce and web services.

Equally, ontology has been broadly used in many other sub-fields of computer science and AI, particularly in areas that concerns Information Retrieval (IR) [14] and Information Extraction (IE) [15], Ontology-Based Information Extraction (OBIE) [16], database management systems [17], information management and intelligent systems integration [18], knowledge representation [19], and in context of this study, Semantic-based Process Mining [2, 4, 6].

Clearly, the representation of knowledge using ontologies helps in organising datasets of complex structures (e.g. the fuzzy models). Moreover, the work in this chapter claims that by using the ontology as a conceptual consistency constraint, a fuzzy model with unlabelled data can be tuned into one (semantic model) that have the best consistency based on the prior knowledge or information. In addition, the formal representations and the resulting metadata

(process descriptions) allows for automatic reasoning of the whole ontology with the aim of retrieving meaningful and useful knowledge that are inferred. Apparently, such reasoning disposition ensures that the process elements specifications within the ontologies are logically interpreted in a suitable manner that enables the automatic reasoning over the explicit knowledge about the domain processes in view [13].

Therefore, the main benefits of ontologies can be summarised in two forms:

i. encoding knowledge about specific process domains, and

ii. advanced analysis and reasoning of the processes at a more conceptual levels.

Likewise, one of the main benefits of ontologies particularly the OWL is that the schema is capable of declaring the different *classes* and *object/data properties* in any given process domain. In turn, it classifies those classes or properties into a taxonomy (i.e. *subClass* and *subProperty* hierarchy) by assigning the *domains* and *ranges* in the same way as the RDF schema [7]. Moreover, the resulting logical models allows the use of a *reasoner* to check if or not all of the definitions or expressions within the ontologies are equally consistent and recognises which concepts fits under which class, as well as, what the meaning of the individual specific properties are [19]. To end with, state of the art tools used for constructing ontologies (e.g., Protégé, SWOOP, and TopBraid Composer) makes use of those reasoners to make available the inference knowledge (i.e. the underlying inferred classes) to the developers or users predominantly in understanding the logically impacts or implications of their developed ontologies and design frameworks [18, 20].

3. Semantic reasoning

The main benefit of OWL ontologies is the capability to automatically compute the class hierarches (i.e. taxonomy) and the underlying relationships that exist amongst the different process elements (entities) by making use of a *reasoner*. Truly, *Reasoners* [2, 9] are essentially used to infer and check if a specific class is a subClass, or superClass of another, or not at all within the ontology, and as such automatically computes the inferred class hierarchy [4, 12].

Indeed, an additional function offered by the reasoner especially as used in this study is *consistency checking* of the process elements and parameters. This means that based on the *process description* or *attributes* within the ontology, the reasoner is able to use the underlying information to *check* if it is possible for any instances (individuals) to become a member of a class. Hence, a class is classified as being inconsistent if it cannot perhaps have any instance.

Moreover, a reasoner is every now and then also referred to as *classifier*. According to Ref. [3] a classifier is a function that maps the attributes of an event onto a label used in the resulting process model. Therefore, in context of ontology-based systems, a classifier (i.e. the reasoner) maps the taxonomy of the defined domain process by matching the various classes with their resulting process instances and/or attributes. In short, the process of computing the inferred class hierarchies in an ontology is typically known as *classifying the ontology*. Henceforth, the reasoner is regarded as the *classifier* or the *inference engine* used in querying and manipulation of the whole ontology.

Thus, the main function of the reasoner is summarized as follows:

- *Classifier*—used in computing the class hierarchies i.e. taxonomy
- *Consistency Checking*—for the inferred process elements, relations and parameters.

4. Ontology-based method and design framework

This study claims that the quality augmentation of process models is as a result of employing *semantic process mining* or better still ontology-based approaches and querying methods which encodes the envisaged system with the three rudimentary building blocks—semantic labelling (annotation), semantic representation (ontology), and semantic reasoning (reasoner) as described in the following section.

4.1. *Semantic process mining* framework: the 2-D rhombus approach

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

In **Figure 1**, the work introduces the framework for the proposed sematic approach (also referred to as 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 that links and relates to defined terms in an ontology, and in so doing, encodes the process logs and the deployed models in the formal structure of ontology (semantic modelling).
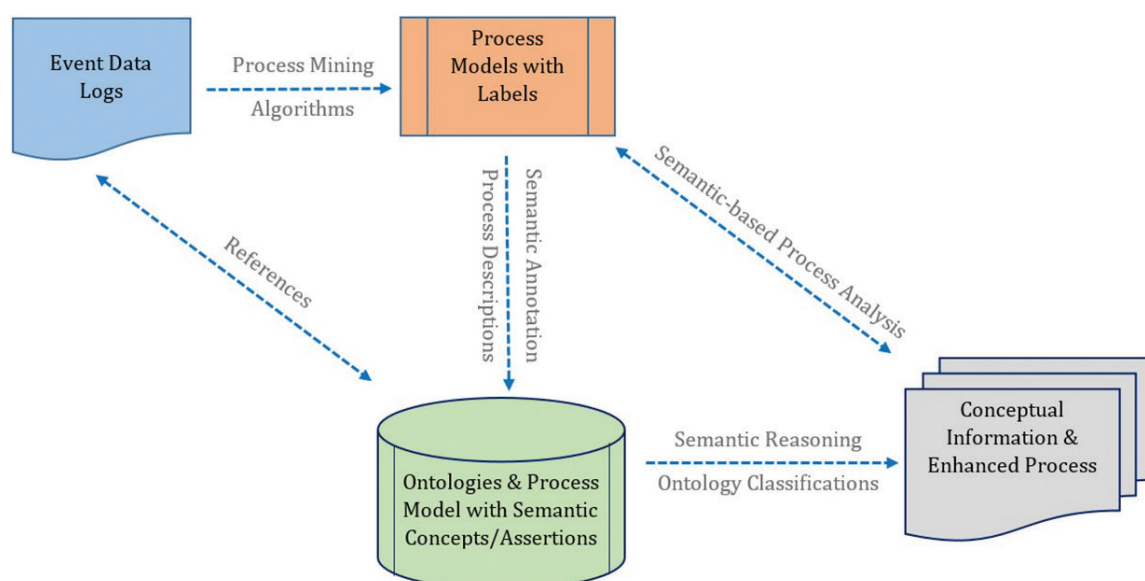

Figure 1. Proposed Framework for the semantic-based (ontology) process mining and querying method.

- the Reasoner (inference engine): which is designed to perform automatic classification of task and consistency checking to validate the resulting model as well as clean out inconsistent results, and in turn, presents the inferred (underlying) associations.

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

- the conceptual referencing supports semantic reasoning over the ontologies in order to derive new information (or knowledge) about the process elements and the relationships they share amongst themselves within the knowledge base.

Therefore, to summarize the design framework, the work shows that the application of semantic-based or better still ontology-based process mining and querying methods must focus on feeding the algorithms with two key core elements:

1. Event Logs and process models which their labels have references to concepts in an ontology, and.

2. Reasoners which are invoked to reason over the resulting ontologies for the event logs and models.

Indeed, the use of such framework and its application have gained a significant interest within the field *semantic process mining* in recent years. On the one hand, the proposed framework trails to make use of the semantics captured in event data logs (i.e. metadata) to create new techniques for process mining or yet still support the enhancement of existing ones in order to assist humans in gaining a novel and much more accurate results. On the other hand, the semantic-based analysis helps to provide the process mining and querying results at a much more level of abstraction so they can be understood easily 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 techniques and algorithms lack the ability to identify and make use of such semantics across the different domains. Nonetheless, the work in this chapter shows through the proposed approach in Section 4.2 and the semantically motivated algorithms in Section 4.3—that by annotating and encoding process models with rich semantics and the integration of semantic reasoning, that it is possible to specify useful domain semantics capable of bridging the semantic gap conveyed by the traditional process mining techniques. Thus, with the semantic-based approach, useful information (i.e. semantics) about how activities depend on each other in a process domain is made possible, and essential for extracting models capable of creating new and valuable knowledge.

To this end, the next section of this chapter presents the main components and architecture of the proposed approach in details, as well as, explain how the study have used the method to support the implementation of the proposed approach and algorithms.

4.2. Main components of the proposed semantic-based approach

This section looks at the general architecture of the semantic-based approach and how the main building blocks (i.e. annotated logs/models, ontology, and semantic reasoning)

has been integrated in the development of the system. Clearly, the work summarizes in **Figures 2** and **3** the various components of the proposed system and its implementation as follows:
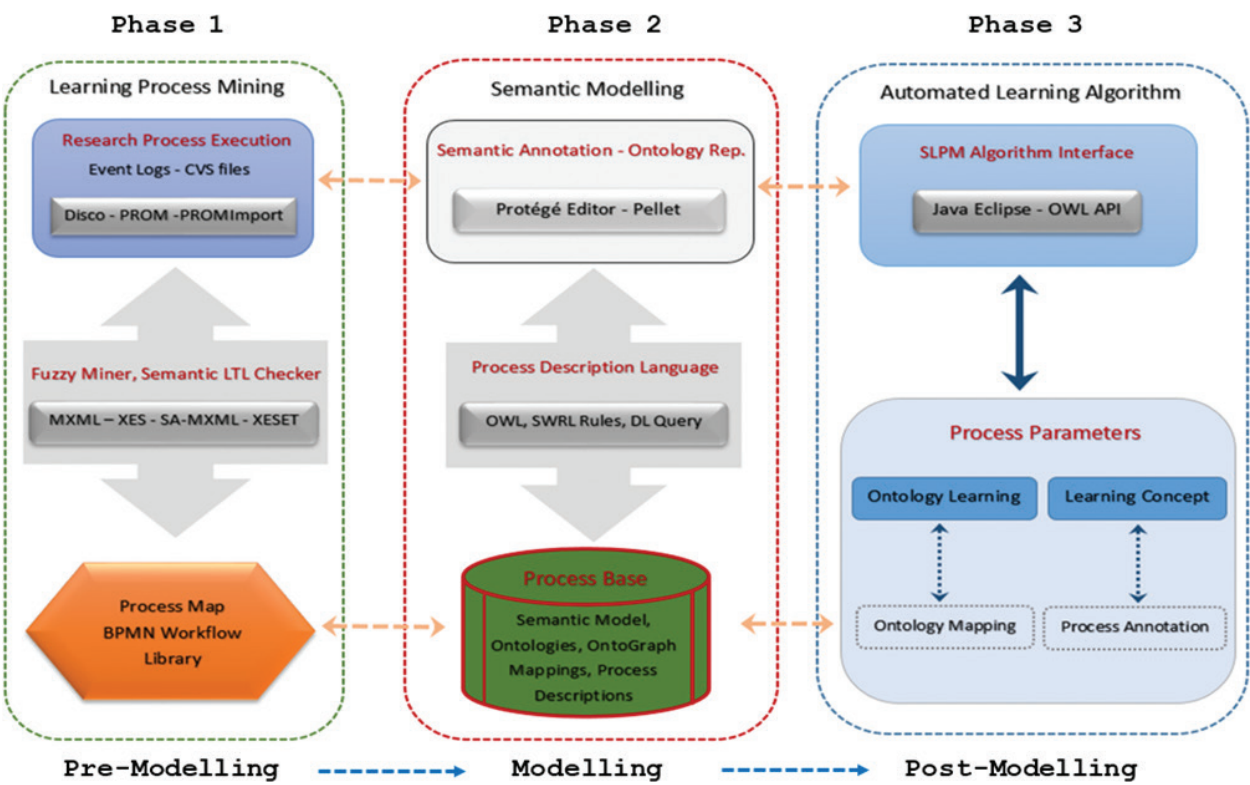

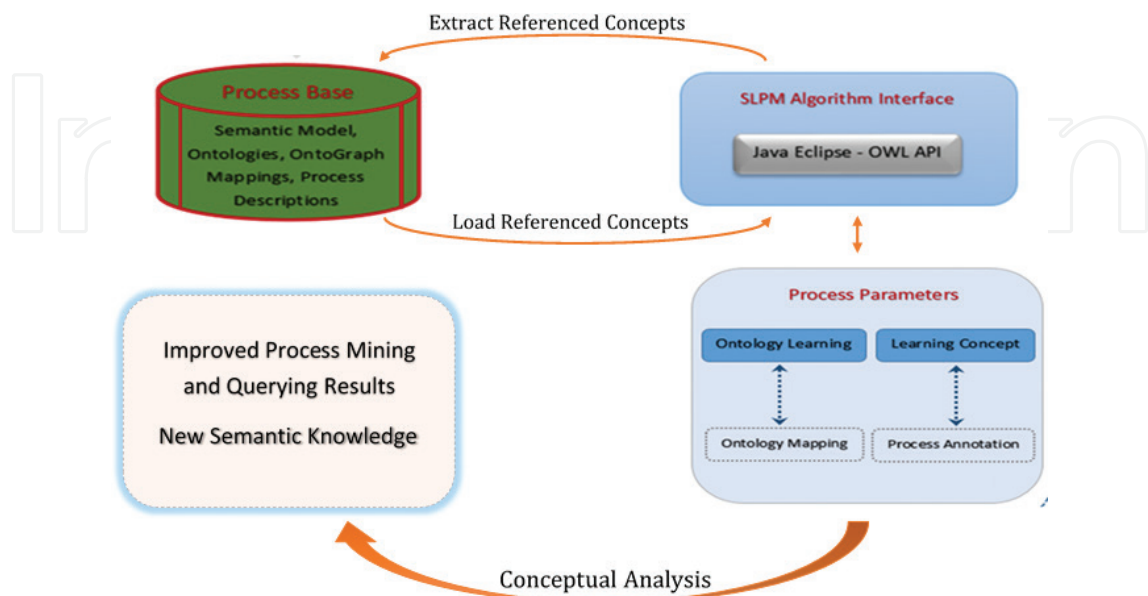**Figure 2.** Architecture of the proposed semantic-based process mining and querying approach.

Figure 3. Practical aspects of implementing the proposed system and its main functions.

Figures 2 and **3** represents an overview of the various components of the semantic-based approach proposed by this study including the different stages of its development and implementation, as follows:

In Phase 1: the study applies the *process mining* techniques in order to make available the process mappings for the learning process, and check its conformance with the event logs based on the Fuzzy Miner as described in Ref. [4]. The main reason is that the resulting process map allows us to quickly, and interactively explore the processes into multiple directions and to show the individual *activities workflow*, and then provide platform for semantic annotation of the different process elements within the knowledge base.

In Phase 2: the work performs *semantic modelling* of the resulting process mappings in terms of the annotated terms. Thus, the semantic model represents the domain knowledge about the various activities and sequence workflows including the concepts defined in an *Ontology* by using process description languages such as the OWL [3] and SWRL [7]. In addition, the approach also makes use of the *Reasoner* i.e., Pellet—to infer the different process instances and the ontological representation (taxonomy) of the learning process model in reality [6].

In Phase 3: the study implements the semantic-based application used for extraction and automated mining or querying of the learning concepts. The work uses the Eclipse Java Runtime Environment to create the methods and interface for loading the *Process Parameters* (i.e. the ontology concepts). Essentially, the work makes use of the OWL Application Programming Interface (OWL API) to extract and load the inferred concepts within the ontology. The purpose is to match the questions one would like to answer about the relationships or attributes the process instances share amongst themselves by linking to the inferred *concepts* within the defined *ontology*.

4.3. Proposed semantic-based algorithms and its formalization

The semantic depiction (*representation*) of processes in an ontological form is a very important step in the proposed approach in this study. The method is aimed at unlocking the information value of the event logs and the derived 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 the individual process instances relies exclusively on the ability to represent such information in a formal way (ontology) to create platform for a much more conceptual analysis of the process instances.

The following Algorithm 1 describes how this work generates the ontology from the process models and event logs:

Algorithm 1: Developing ontology from process models and event logs

1: For all defined models M and event log EV

2: **Input:** C—different classes for all process domain

R—relations between classes

I—sets of instantiated process individuals

A—sets of axioms which state facts

3: **Output:** Semantic annotated graphs/labels & an ontology-driven search for process models and explorative analysis

4: **Procedure**: create semantic model with defined process descriptions and assertions

5: **Begin**

6:   **For all** process models *M* and event log *EV*

7:      **Extract** Classes *C* ← from *M* and *EV*

8:      **while** no more process element is left **do**

9:      **Analyze** Classes *C* to obtain formal structures

10:        **If** *C* ← Null **then**

11:           obtain the occurring Process instances (*I*) from *M* and *EV*

12:        **Else If** *C* ← 1 **then**

13:           create the Relations (*R*) between subjects and objects // i.e. between classes *C* and individuals (*I*)

14:        **If** relations *R* exist **then**

15:           **For** each class *C* ← semantically analyse the extracted relationships (*R*) to state facts i.e. Axioms (*A*)

16:           create the semantic schema by adding the extracted relationships and individuals to the ontology

17: **Return**: taxonomy

18: **End** If statements

19: **End** while

20: **End** for

According to Ref. [13] ontologies, i.e. *Ont* ∈ *Onts*, are formal explicit specification of shared conceptualization that can be applied in any context, for example, as exploited in this study to model the case study of the learning process. Indeed, the semantic annotated logs and models are very fitting for further steps of semantically enhancing and accurate 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 ontologies.

Ultimately, from the described Algorithm 1, we recognize that ontology is a quadruple, i.e.

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

which consists of different classes *C* and relations *R* between the classes [13, 21]. Perhaps, a relation *R* trails to connect a set of classes with either another class, or with a fixed literal and is capable of also describing the sub assumption hierarchy (i.e. taxonomy) that exists between

the various classes and their relationships. In addition, the classes are instantiated with a set(s) of individual, I, and can likewise contain a set(s) of axiom, A, which states fact (e.g. what is true and fitting within the model, or what is false and not fitting in the model).

Therefore, to achieve this importance step in this work, 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 the Objects 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.

Obviously, the defined concepts and process descriptions as explained in the steps above means that the *semantic annotation* is also another essential component in realizing such an ontology-based approach that supports automated process mining and querying by automatically conveying the formal semantics of the derived process models and extracted logs [21]. In other words, the annotated process models or logs are necessary for the semantic-based analysis, process querying and further steps of enhancing the model.

Essentially, *semantic annotation*(*SemAn*) is defined formally as a function that returns a set of concepts from the ontology for each node or edge in the graph [21]. Thus,

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

where: *SemAn* describes all kinds of annotations which can be input, output, meta-model annotation etc. It is also important to note that semantic annotations could be carried out either manually or automatically computed bearing in mind the similarity of words [22] to generalize the individual entities within the domain process in view. Therefore, a *semantic annotated graph* (see **Figure 4**) is defined as follows:

$$Gsem \; = \; (Nsem, Esem, Onts) \; with \; Nsem \; = \; \{(n, SemAn(n)) \mid n \; \in \; N\} \; and \; Esem$$
$$= \; \{(nsem, n\_sem) \mid nsem \; = \; (n, SemAn(n)) \wedge n\_sem$$
$$= \; (n\_, SemAn(n\_)) \wedge (n, n\_) \; \in \; E\}$$

In fact, semantically planning of any ontology-based system requires that all process actions within the defined ontology must perhaps include some form of semantic annotation. Thus;

According to the definitions in Ref. [21] if we Let A be the set of all process actions. A process action $a \in A$ is characterized by a set of input parameters $Ina \in P$, which is required for the execution of a and a set of output parameters $Outa \subseteq P$, which is provided by a after execution. All elements $a \in A$ are stored as a triple (*namea, Ina, Outa*) in a process library *libA*.
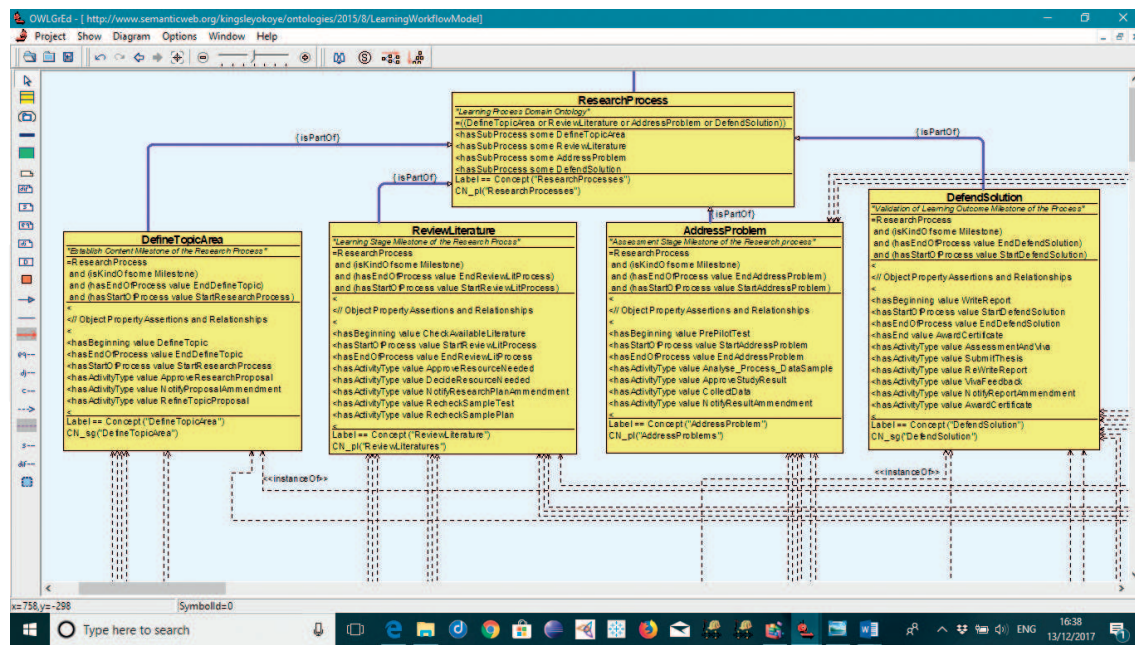
Figure 4. Research process domain with description of the learning activity concepts and relationships.

To this end, the last essential component in realizing the ontology-based approach 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 (i.e. assertions) within the domain ontology, the *reasoner* is able to use the underlying information to check if it is possible for any process instances (individuals) to become a member of a class, and to provide the necessary results or associations as requested based on the executed queries or information retrieval process.

Accordingly, the following Algorithm 2 describes how this study makes use of the reasoner to classify and infer the necessary associations to produce the outputs:

Algorithm 2: Reasoning over Ontologies and Classification of Entities and Outputs

1: For all defined Ontology models *OntM*

2: **Input:** classifier e.g. Pellet Reasoner

3: **Output:** classified classes, process instances and attributes

4: **Procedure**: automatically generate process instance, their individual classes and Learning concepts

5: **Begin**

6:    **For all** defined object properties (*OP*) and datatype properties (*DP*) assertions in the model (*OntM*)

7:       **Run** reasoner

8:       **while** no more process and property description is left **do**

9:       **Input** the semantic search queries *SQ* or set parameter *P* to retrieve data from *OntM*

10   **Execute** queries

11:         **If** *SQ* or *P* ← Null **then**

12:             re-input query or set the parameter concepts

13:         **Else If** *SQ* or *P* ← 1 **then**

14:             infer the necessary associations and provide resulting outputs

15: **Return**: classified Concepts

16: **End** If statements

17: **End** while

18: **End** for

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 in order to answer and produce meaningful knowledge, and even in most cases, new information about the process elements and the relationships they share amongst themselves within the knowledge base.

5. Use case scenario and implementation

The use case scenario in this chapter is based on running example of a *Research Learning Process*. The work makes use of the events log about the research process to prove how the proposed approach is applied to represent and answer real time questions about a learning process. In the case study example as presented in our previous study in [6], the work shows that the first step to conducting a research is to decide on what to investigate, i.e. 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. Basically, these process involves the workflow of the journey from choosing the research topic to being awarded a certificate, and comprises of sequence of practical steps or set of activities through which must be performed in order to find answers to the research questions [6].

Indeed, as shown in [6] the workflow for those 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 the process mining perspective, the derived process models may not disclose to us some of the valuable information at the semantic or abstraction levels, despite all of the mappings from mining the process. For example, the process maps may not disclose how the individual process instances that makes up the model interact or differ from each other, which attributes they share amongst themselves

within the knowledge base, or the activities they perform together or differently. In turn, questions like—who are the individuals that have successfully completed the research process? may not be established. For such reason, the study in [6] has shown that by adding semantic knowledge to the deployed models, it becomes possible 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 milestone which is necessary to ensure the research outcome, such learner can be classified as incomplete. In such formal way, it becomes possible to logically ascertain which individuals has successfully completed the research process or not.

Therefore, the following section explains how the work uses the case study of the Research Process domain to demonstrate the capability of the ontology-based approach and algorithms by analyzing the learning activity logs based on concepts. Henceforth, presenting the process mining and querying results at a much more conceptual level.

5.1. Semantic representation and modelling of research learning process

In this section, the work implements the semantic-based approach to find out patterns/behaviour that describes or distinguishes certain entities within the learning knowledge base from another. Thus, 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 attributes (concepts) and the application of semantic reasoning, it becomes easy to refer to a particular case (i.e. certain group of learners). Principally, the study focus is therefore on the use case scenario of the *Successful* and *Uncomplete* learners.

Apparently, the work in [6] describes 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. In view of that, the work provides the four milestones; Establish Context → Learning Stage → Assessment Stage → Validation of Learning Outcome (as illustrated in **Figure 4**) 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 [6].

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 [6]. Henceforth, as described in **Figure 4** the work shows the *Learning Activity* concepts that are defined in the learning model ontology, and how they are mapped to the various milestones of the Research Process to ensure sequence of transitions during the entire learning process.

Indeed, the drive for such semantic mapping of the activity concepts is that the method allows the meaning of the learning objects and properties to be enhanced through the use of property descriptions (semantic annotations) and classification of discoverable entities (reasoning).

For instance, to address the real time learning questions the work have identified in Section 5 in relation to the successful and uncomplete learners. We refer to the deployed model, and to that 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) [6].

Moreover, as shown in **Figure 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 [6].

Accordingly, to ascertain the class of the "uncomplete learners", it was also necessary to refer the object properties in order to determine what attributes distinguishes such learners from the Successful ones.

Therefore, the work describes that an Uncomplete Learner is a subclass of, amongst other NamedLearnerCategory, a Person that performs some Learning ActivityConcept who has a universal object property restriction/relationship with only some of the milestones of the ResearchProcess Class but not all of the classes [6].

As shown in **Figure 6**—the necessary condition is: if something is an Uncomplete 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 only some of the Class, i.e. DefineTopicArea, ReviewLiterature, AddressProblem but not all of the four classes [6].

Ideally, we observe in **Figures 5** and **6** 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 participant or member of a
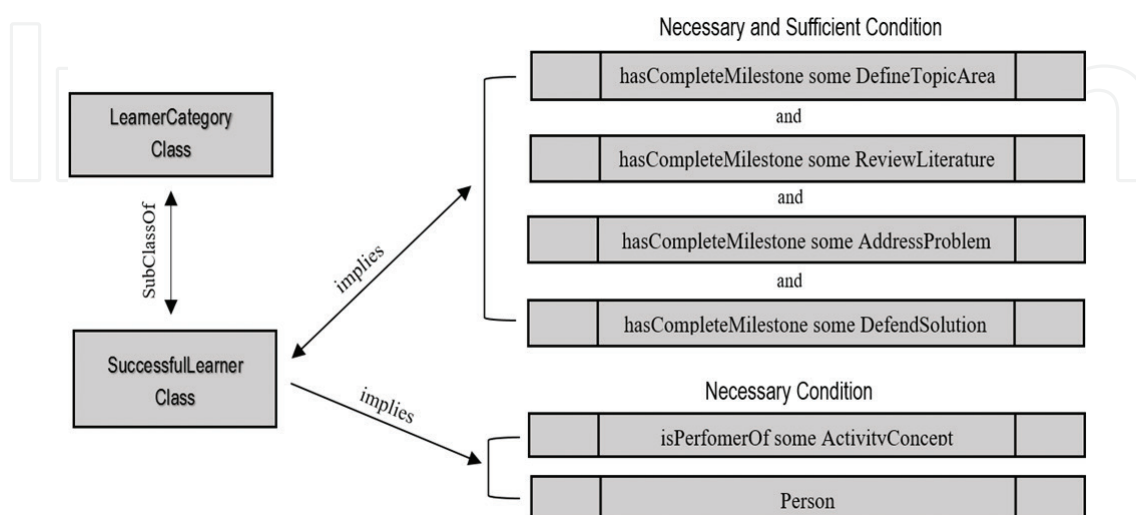

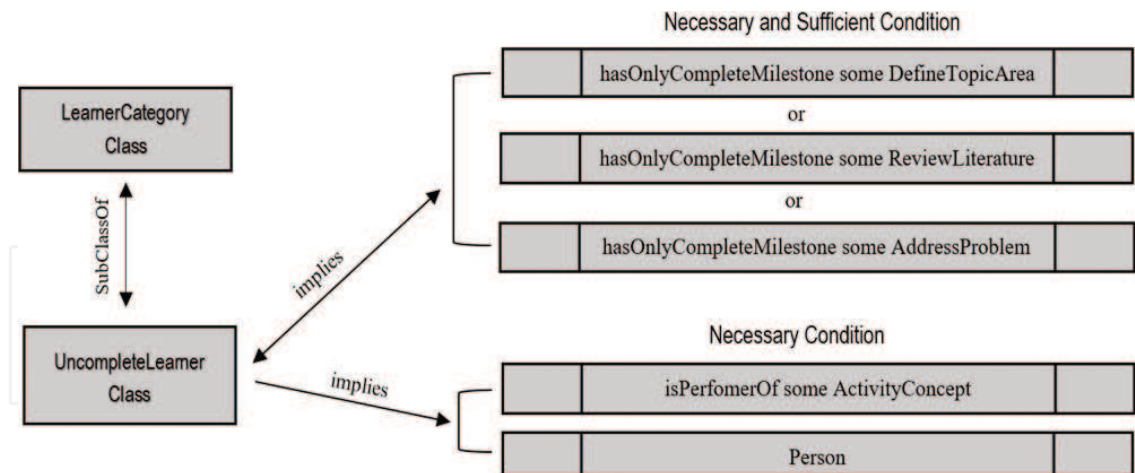**Figure 5.** Attributes/object property assertions for the SuccessfulLearner Class.

Figure 6. Attributes/object property assertions for the UncompleteLearner Class.

specific class e.g. the successful or uncomplete learner class. As noted in Ref. [6], 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 identified in Section 5.

Indeed, property restrictions (structured organisation) and semantic labelling serves as a good practice for representation of the learning process information by providing a formal way of determining the individual process instances within the learning knowledge base.

For example, the following are description of the implemented ontology concepts and axioms for the "successful learner" class within the learning model following the definitions in **Figure 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://attempto.ifi.uzh.ch/acetext#acetext"/>
```
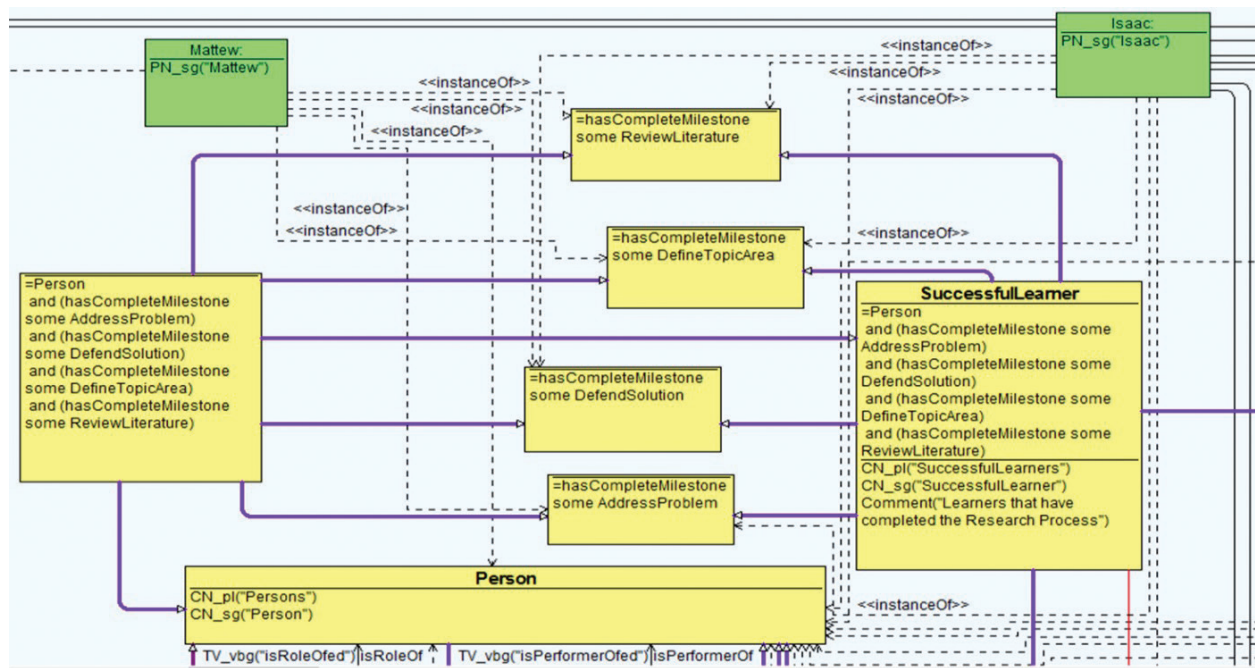
Figure 7. Concept assertions and the different formal relationships for the SuccessfulLearner Class.

```
<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 has-
Milestones a DefineTopicArea and that hasMilestones a ReviewLiterature
is a SuccessfulLearner.</Literal>
```

```
</Annotation>
```

```
</EquivalentClasses>
```

On the other hand, the work also provides example description of the implemented ontology concepts and axioms for the "uncomplete learner class" within the learning model following the definitions in **Figure 8** including the OWL XML file syntax as follows:

```
1: ontology ResearchProcess
```

```
2: concept UncompleteLearner
```

```
3: hasOnlycompleteMilestone ofType {DefineTopicArea, Or
ReviewLiterature, Or Address Problem, Not DefendSolution}
```

```
4: isPerformerOf some LearningActivity
```

```
5: is ofType Person
```

```
6: hasInstance members {Paul, Danny, Mark, Gregory, John}
```

```
7: axiom DefinitionOfUncompleteLearner
```

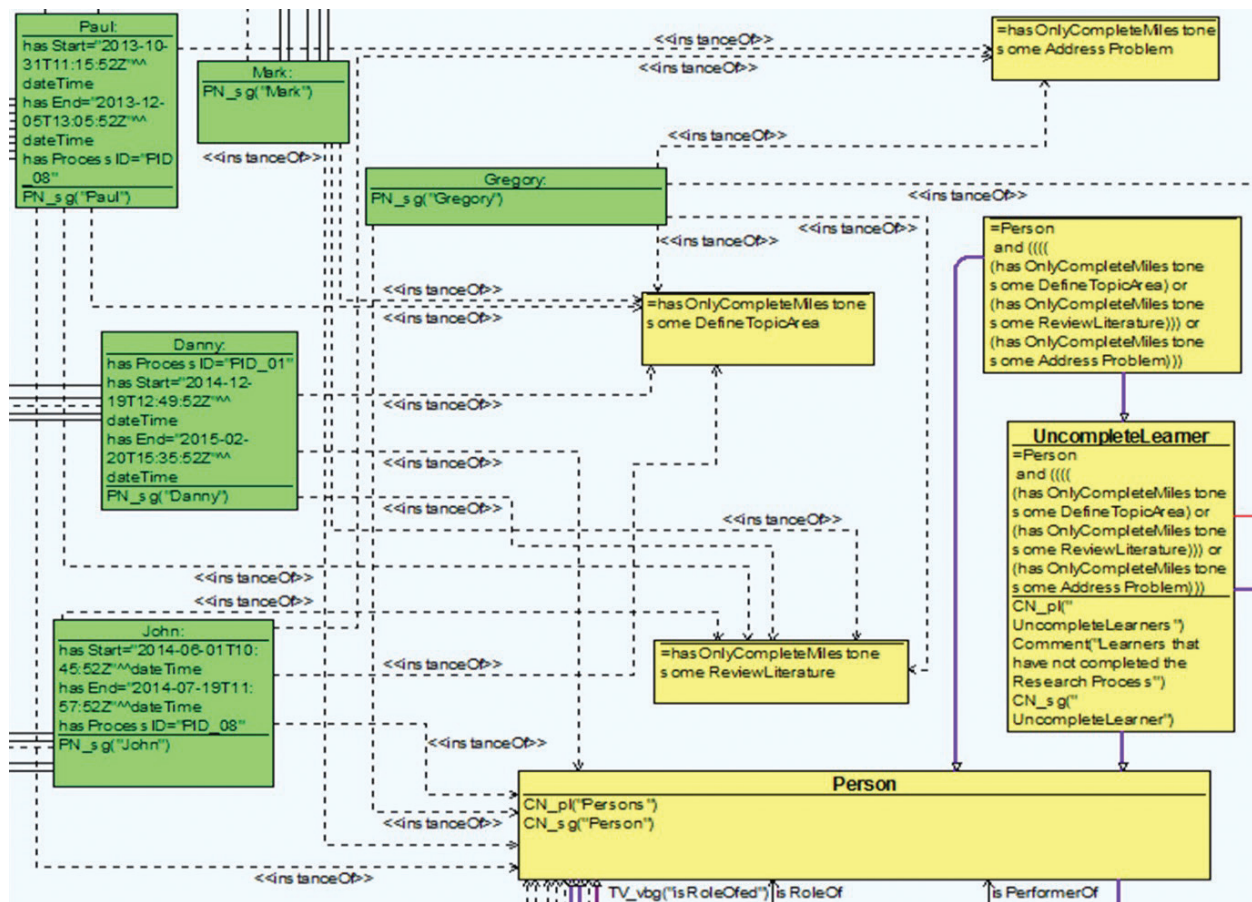**Figure 8.** Concept assertions and the different formal relationships for the UncompleteLearner Class.

```
<EquivalentClasses>

<Annotation>

<AnnotationProperty IRI="http://attempto.ifi.uzh.ch/acetext#acetext"/>
<Literal datatypeIRI="&xsd;string">Every UncompleteLearner is a Person
that onlyHaveMilestones an AddressProblem or that onlyHaveMilestones
a DefineTopicArea or that onlyHaveMilestones a ReviewLiterature. Every
Person that onlyHaveMilestones an AddressProblem or that onlyHaveMile-
stones a DefineTopicArea or that onlyHaveMilestones a ReviewLiterature
is an UncompleteLearner.</Literal>

</Annotation>

</EquivalentClasses>
```

5.2. Description logic queries and process reasoning

The Description Logic (DL) query [9] is a process description language or syntax that could be used to check for consistency for all defined entities within the ontology model. It makes use of the *Reasoner* as previously explained in Section 3 to perform automatic classification of the relationships (i.e. property assertions) that are described within the ontology.

Likewise, this work makes use of the syntax to compute and ascertain the inferred classes and individuals within the learning domain ontology [23]. The queries are implemented in order to check that all parameters (entities) within the defined classes are true and at least falls within the universal restriction of validity by definition, and that there are no inconsistency of data or repeatable contradicting discovery.

Consequently, the study as shown in Ref. [23] provides the following example queries to explain how it employs the DL queries to perform automatic classification and/or retrieval of the process instances (entities) within the ontology. Thus:

DQ1. Is DefineTopic an Activity of the first Milestone (DefineTopicArea)?

DL Query: `ActivityConcept and is ActivityType Of some DefineTopicArea`

== the DL query checks if the activity of the first Milestone equal to Define Topic, thus compares the activity of the first Milestone DefineTopicArea with Activity Concept (DefineTopic)

DQ2. Is the Last Activity of the Research Process Award Certificate?

DL Query: `(i) ResearchProcess and hasEnd value AwardCertificate`

`(ii) ActivityConcept and isEndOf some ResearchProcess`

== the query computes and checks the last Milestone of the research process and compares if the last activity is equal to Award Certificate. Hence, compares the activity of the last Milestone DefendSolution with AwardCertificate

DQ3. Is CollectData an Activity of the Third Milestone Address Problem?

DL Query: `ActivityConcept and isActivityTypeOf some AddressProblem`

== computes and check the activities of the Third Milestone AddressProblem, thus compare if the result is equal to the Activity Concept CollectData

DQ4. Does Person P Activity A?

Example: Does Person (Richard) Activity Approve Research Proposal?

DL Query: `Person and hasActivityType value ApproveResearchProposal`

== the query computes and check persons related to the Approve Research Proposal and then compares if person (Richard) does the activity ApproveResearchProposal.

DQ5. Does person P activity of activity A and B?

Example: Which Persons does Activity RecheckSamplePlan and ReWriteReport?

DL Query: `Person and hasActivityType some {RecheckSamplePlan, ReWriteReport}`

== computes and check which persons in the model does activity RecheckSamplePlan and ReWriteReport.

DQ6. Does Person P activity A and then B and then C?

Example: Does person Paul activity of type CollectData and then Edit_Code_Data Sample and then Analyse_Process_Data Sample?

DL Query: `Person and hasActivityType some {CollectData, Edit_Code_Data Sample, Analyse_Process_Data Sample}`

== the query computes and check if person Paul does the activity {Collect Data, Edit_Code_Data Sample, Analyse_Process_Data Sample} [23].

6. Related works

Process querying is an emerging method for automated management of real-world and envisioned processes, models, repositories, and knowledge within the field of business process management and organisational data analysis [4, 5, 24]. According to [24] the process querying techniques concerns automatic methods for handling (e.g. filtering or manipulating) repositories of models of observed and unseen processes as well as their relationships, with intension of transforming the process-related information into decision making capabilities.

In practice, Ref. [5] notes that the process querying research spans a range of topics from theoretical studies of algorithms and the limits of computability of process querying techniques to practical issues of implementing the querying capabilities in software products [2–4, 17, 19, 25]. Also, Ref. [5] observes that such approaches which trails to combine process models and ontologies (particularly *ontologies for process management*) are increasingly gaining attention in recent years. According to the authors one reason for such growing interest, is that ontologies permits the adding of semantics to discovered or existing process models which in turn enables the automated inference of knowledge from the domain processes in question. Consequently, the derived knowledge (semantics) could then be used to manage any process (e.g. business processes) both at design and/or execution time.

In view of that, the authors in [5] propose a process querying framework used for enabling business intelligence through query-based process analytics. The framework structures the state of the art components built on generic functions that can be configured to create a range of querying techniques, and also points to gaps in existing research and use cases within the BPM and BI fields [3]. According to [3, 5] process querying methods need to address those gaps. For instance, organizations often fail to convert the high volume of data recorded in the information system into strategic and tactical intelligence. This is due to the lack of dedicated technologies that are designed to effectively manage the information about the instances (entities) encoded within the envisioned process models or data records, in order to better support strategic decision-making and provide the next generation of Business Intelligence. Interestingly, the proposed framework listed in [5] is an abstract system in which components can be selectively replaced to result in a new process querying method.

For the purpose of the work done in this chapter, our focus is particularly on the *Process Querying with Rich Annotations* [24] which studies the use of rich ontology annotations of process models for the purpose of process querying. Besides [11] notes that a trace abstraction technique for semantic-based process mining and model analysis should present methods or design frameworks which are able to convert actions found within the discovered models into higher level concepts based on the domain knowledge, thus, the term *conceptualization*.

7. Discussion and conclusion

The study in this book chapter introduces a design framework, method and algorithms used for implementation and semantic integration of process models in order to improve their analysis and querying process. Typically, the work recognizes that much of the effort in developing sematic-based process mining or better still ontology-based systems and approaches, relies mainly on constructing an effective system that integrates the three main building blocks (i.e. annotated logs or models, ontology and semantic reasoning). Hence, whilst the *semantic annotation* process is focused on describing the meaning of the process models and its entities or attributes, the *ontology* is devoted to binding together the different concepts, classes and properties in a way that maximizes their influence and outcomes. The work notes that the best way to create such systems is to make use of tools that supports the different components particularly the ontology which every now and then are required to maintain consistency of the process elements and formal hierarchy. Without a doubt, the use of a *reasoner* to compute relations between the various entities (process instances) in the ontology is practically possible, especially when building huge ontologies with numerous entities in them. Perhaps, without an automated classification process (semantic reasoning) it may become very challenging to manage those massive ontologies particularly in a precise logic way. Moreover, not only does this kind of ontology-based approach supports the application of rules and languages such as the OWL, SWRL and DL queries and/or re-use of an ontology by another ontology, but it also minimalizes the level of human-errors which are every now and again present especially when managing the manifold existence of entities or concepts within the ontologies or process knowledge-base.

Even more, the work has shown how the proposed semantic-based approach is applied to answer real time questions about the process domains as well as the classification of the individual process elements that can be found within a process knowledge-base. The study illustrates this through the use case scenario of the learning process. Significantly, such method of quality classification for individual traces within the learning process base can be utilized by the process analysts or IT experts as a way of performing useful information retrieval and/or 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.

In summary, the use of ontologies and the relations between the concepts in the ontologies can be utilized to collectively combine tasks and compute process models in a hierarchical form (taxonomy) including several levels of abstraction The main idea is that for any

ontology-based system such as the semantic-based process mining approach, these aspects of aggregating the task or computing the hierarchy of the process models should not only be machine-readable, but also machine-understandable. This means that the process models are either semantically annotated, or already in a form which allows a computer (i.e. the reasoner) to infer new facts by making use of the underlying ontologies.

Author details

Kingsley Okoye*, Syed Islam and Usman Naeem

*Address all correspondence to: k.okoye@uel.ac.uk

School of Architecture Computing and Engineering, Docklands Campus, University of East London, London, UK

References

[1] Dou D, Wang H, Liu H. Semantic data mining: A survey of ontology-based approaches. In: 9th IEEE International Conference on Semantic Computing; California, USA; 2015. pp. 244-251

[2] de Medeiros AKA, Van der Aalst WMP, Pedrinaci C. *Semantic Process Mining* Tools: Core Building Blocks. In: ECIS; June 2008; Galway, Ireland; 2008. pp. 1953-1964

[3] Van der Aalst WMP. Process Mining: Data Science in Action. 2nd ed. Berlin: Springer-Verlag Berlin Heildelberg; 2016

[4] Okoye K, Tawil ARH, Naeem U, Islam S, Lamine E. Semantic-based Model Analysis towards Enhancing Information Values of Process Mining: Case Study of Learning Process Domain. In: Abraham A, Cherukuri AK, Madureira AM, Muda AK, editors. Advances in Intelligent Systems and Computing book series (AISC, volume 614). Springer International Publishing AG; 2017. pp. 622-633

[5] Polyvyanyy A, Ouyang C, Barros A, van der Aalst WMP. Process querying: Enabling business intelligence through query-based process analytics. Decision Support Systems; 2017;**100**(1):41-56

[6] Okoye K, Tawil ARH, Naeem U, Islam S, Lamine E. Using semantic-based approach to manage perspectives of process mining: Application on improving learning process domain data. In: Proceedings of 2016 IEEE International Conference on Big Data (BigData); Washington, DC; 2016. pp. 3529-3538

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

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

[9] Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF. Description Logic Handbook: Theory, Implementation, and Applications. 1st ed. New York, NY, USA: Cambridge University Press; 2003

[10] Balcan N, Blum A, Mansour Y. Exploiting ontology structures and unlabeled data for learning. In: Proceedings of the 30th International Conference on Machine Learning; Atlanta Georgia, USA; 2013. pp. 1112-1120

[11] Montani S, Striani M, Quaglini S, Cavallini A, Leonardi G. Knowledge-based trace abstraction for *semantic process mining*. In: ten Teije A, Popow C, Holmes J, Sacchi L, editors. Artificial Intelligence in Medicine. AIME 2017. Lecture Notes in Computer Science. Vol. 10259. Australia: Springer, Cham; 2017, pp. 267-271

[12] Hashim H. Ontological structure representation in reusing ODL learning resources. Asian Association of Open Universities Journal. 2016;**11**(1):2-12

[13] Gruber TR. Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human Computer Studies. 1995;**43**(5):907-928

[14] Manning CD, Raghavan P, Schütze H. Introduction to Information Retrieval. Cambridge, UK: Cambridge University Press; 2008

[15] Cunningham, H. Information Extraction, Automatic, University of Sheffield, UK [Internet]. 2005. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.66.8785& rep=rep1&type=pdf [Accessed: August, 2017]

[16] Calvanese D, Montali M, Syamsiyah A, van der Aalst WMP. Ontology-driven extraction of event logs from relational databases. In: Reichert M, Reijers H, editors. Lecture Notes in Business Information Processing. BPM Workshops 2015. Springer, Cham; 2016. pp. 140-153

[17] Alkharouf NW, Jamison DC, Matthews BF. Online analytical processing (OLAP): A fast and effective data mining tool for gene expression databases. Journal of Biomedicine and Biotechnology. 2005;**2005**(2):181-188

[18] De Giacomo G, Lembo D, Lenzerini M, Poggi A, Rosati R. Using ontologies for semantic data integration. In: Flesca S, Greco S, Masciari E, Saccà D, editors. A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years. Studies in Big Data. Springer, Cham; 2018. pp. 187-202

[19] Kumar AP, Abhishek K, Vipin Kumar N. Architecting and designing of semantic web based application using the JENA and PROTÉGÉ – A comprehensive study. International Journal of Computer Science & Information Technologies (IJCSIT). 2011;**2**(3):1279-1282

[20] Horrocks I. Ontologies and the semantic web. Communications of the ACM. 2008;**51**(12): 58-67

[21] Lautenbacher F, Bauer B, Forg S. Process mining for semantic business process modeling. In: 13th Enterprise Distributed Object Computing Conference Workshops; Auckland; 2009. pp. 45-53

[22] Born M, Dörr F, Weber I. User-Friendly semantic annotation in business process modeling. vol. 4832. In: Weske M, Hacid M, Godart C, editors. Web Information Systems Engineering – WISE 2007 Workshops. WISE 2007. LNCS Series, Berlin, Heidelberg: Springer; 2007. pp. 260-271

[23] Okoye K, Tawil ARH, Naeem U, Lamine E. Discovery and enhancement of learning model analysis through *semantic process mining*. International Journal of Computer Information Systems and Industrial Management Applications. 2016;**8**(2016):93-114

[24] Polyvyanyy A, et al. Process Querying [Internet]. 2016. Available from: http://process-querying.com/ [Accessed: July, 2017]

[25] Okoye K, Tawil ARH, Naeem U, Lamine E. A semantic reasoning method towards ontological model for automated learning analysis. In: Pillay N, Engelbrecht A, Abraham A, du Plessis M, Snášel V, Muda A, editors. Advances in Intelligent Systems and Computing. NaBIC 2015. Switzerland: Springer; 2016. pp. 49-60