Semantic Process Mining Towards Discovery and Enhancement of Process Models and Event Logs Analysis: Application on Learning Process Domain

A Thesis submitted in partial fulfilment of the requirements of the University of East London for the degree of

Doctor of Philosophy

Kingsley OKOYE

August, 2017

Abstract

This thesis introduces a Semantic-Fuzzy mining approach that makes use of labels (i.e. concepts) within event logs to propose a method which allows for mining and improved analysis of the resulting process models through semantic - *annotation, representation* and *reasoning*. Consequently, the thesis presents a framework referred to as Semantic Process Mining and Model Analysis Framework (SPMaAF) that proves useful towards construction of semantic-based process mining technique that exhibits a high level of intelligence and conceptual reasoning capabilities particularly with its application in real world settings.

Indeed, the ability to mine useful or worthwhile knowledge from readily available data in current information systems is a challenge, due to the exponential increase in volume of data that is generated. In consequence, this has spanned the need for a richer and advanced description of real-time processes that allows for flexible exploration of the large volumes of data targeted at improving the system performance and analysis. Such process-related analysis, often allied to *process mining*, means there exist not only the need for techniques that are capable of extracting valuable information from the events logs , but also methods that can be used to analyse and provide conceptual knowledge about the processes in reality.

For that reason, this thesis qualitatively show by using a case study of *Learning Process* – how 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. In addition, the study quantitatively assess the level of accuracy of the classification results to predict behaviours of unobserved patterns within the process knowledge-base.

Accordingly, the study looks at the level of impact and usefulness of the proposed SPMaAF framework, Algorithms formalizations, validity of the Classification results, and their influence compared to other existing benchmark algorithms and techniques used for process mining. The research outcomes 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 outcomes from the syntactic level to a much more conceptual level. The method results in a process 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 various process instances and the discovered models. To this end, the research 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 useful patterns and/or behaviour.

Acknowledgement

A lot of people contributed to the authenticity and successful completion of this thesis. Therefore, it's my pleasure to acknowledge their hard work, supports and contribution.

First of all, I will like to thank my Director of Studies: Dr Syed Islam and Supervisor: Dr Usman Naeem - for their relentless encouragement and precise support and advice throughout the course of this research. You both assisted me a lot in clarifying the research issues, and even more, make it become a successfully completed project. Indeed, your professionalism influenced me academically from which I have also gained the benefit of how to discipline and manage my workload especially for my future careers and engagements.

I will also like to express my special thanks to Dr Abdel-Rahman H. Tawil and Dr Elyes Lamine for their immeasurable input to the success of this research work. I am very grateful to them for their insightful guidance and valuable advice when this research all started.

Thank you to everyone at the University of East London for providing me with such an enjoyable and productive environment to carry out my research work. Most especially the Graduate School and all the staffs at the School of Architecture Computing and Engineering that provided me with the necessary programs, materials, and facilities that allowed me to conduct the doctoral research effectively for the past five years.

My tremendous and most important gratitude goes to my beloved family, friends and relatives who has stood behind me all through the duration of my PhD study. I have received from them overwhelming encouragement, financial support, and solace even at the times of difficulty. Thank you to everyone in my family for your continued believe in my education, and I will forever be indebted to each and every one of you. I will especially like to say a very big thank you to Jurate and Skaiva for giving me life entirely different from after long days and even nights of doing my university work. I profoundly appreciate your loving and unconditional support and patience during the whole time, and must acknowledge that you have all been my inspiration and anchor in life's deepest ocean. Of course, my thanks would not be complete without expressing my deep gratitude to all my friends who were there, and supported me through my PhD research. You have all been wonderful and great pillar to me.

Dedication

This thesis is most especially dedicated to God Almighty for His protection, wisdom and strength to complete this project. And to my late Parents, I am glad and proud to have fulfilled my promise and could not have been ever happier and pleased to make such goal come true.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AEHS	Adaptive Educational Hypermedia Systems
AI	Artificial Intelligence
AIs	Augmented Intelligence systems
BAM	Business Activity Monitoring
BDM	Balanced Distance Metric
BI	Business Intelligence
BPI	Business Process Intelligence
BPIC	Business Process Intelligence Challenge
BPM	Business Process Management
BPMN	Business Process Modelling Notation
CFSM	Casual and Fuzzy Student Model
C-net	Casual Nets
CPI	Continuous Process Improvement
CPM	Corporate Performance Management
CRM	Customer Relationship Management
DL	Description Logic
DM	Data Mining
EDM	Educational Data Mining
ELS	Educational Learning Systems
EPM	Educational Process Mining
ERP	Enterprise Resource Planning
FM	Fuzzy Miner
GATE	General Architecture for Text Engineering
HM	Heuristic Miner
IAELS	Intelligent and Adaptive Educational Learning Systems
ICT	Information and Communication Technology
IE	Information Extraction
IM	Inductive Miner
IR	Information Retrieval
KIF	Knowledge Interchange Format

KIM	Knowledge and Information Management system
LA	Learning Accuracy
LTL	Linear Temporal Logic
MXML	Mining eXtensible Markup Language
NLP	Natural Language Processing
OBDA	Ontology-Based Data Access
OBIE	Ontology-based Information Extraction
OKBC	Open Knowledge Base Connectivity Protocol
OLAP	Online Analytical Processing
OWL API	Ontology Web Language Application Programming Interface
OWL	Ontology Web Language
PAIS	Process-Aware Information Systems
PM	Process Mining
PQ	Process Querying
RDF	Resource Description Framework
RML	Rule Markup Language
SA-MXML	Semantic Annotated Mining eXtensible Markup Language
SFM	Semantic Fuzzy Miner
SLPM	Semantic Learning Process Mining
SM	Student Model
SWRL	Semantic Web Rule Language
TQM	Total Quality Management
WAPS	Workflow Activity Patterns
WFM	Workflow Management System
WSML	Web Service Modelling Language
XES	eXtensible Event Streams
XESEXT	eXtensible Event Stream Extension
XML	eXtensible Markup Language

Research Publications

Okoye, K., Naeem, U., & Islam, S., 2017. Semantic Fuzzy Mining: Enhancement of Process Models and Event Logs Analysis from Syntactic to Conceptual Level. Manuscript submitted to the *International Journal of Hybrid Intelligent Systems* (IJHIS), vol. 14 (1-2), pp. 67–98

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

Okoye, K., 2016. Learning Pattern Discovery: Impact of User-centric Design Approach Towards Enhancement of E-learning Systems. *Computing and Information Systems Journal*, 20(2), pp. 37-49.

Okoye, K., Tawil, A. R. H., Naeem, U., Islam, S. & Lamine, E., 2017. Semantic-based Model Analysis towards Enhancing Information Values of Process Mining: Case Study of Learning Process Domain. In: A. Abraham, A. K. Cherukuri, A. M. Madureira & A. K. Muda, eds. *Advances in Intelligent Systems and Computing book series* (AISC, volume 614). Proceedings of SoCPaR Conference 2016. Springer International Publishing AG. 2018. Chapter 61, p. 622-633.

Okoye, K., Tawil, A. R. H., Naeem, U. & Lamine, E., 2016. 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., eds. *Advances in Intelligent Systems and Computing.* Proceedings of NaBIC 2015 Conference, Springer International Publishing, Chapter 5, pp. 49-60.

Okoye, K., Tawil, A. R. H., Naeem, U. & Lamine, E., 2016. Semantic Process Mining Towards the Discovery and Enhancement of Learning Model Analysis. In: Petrova, V. M, eds. *Advances in Engineering Research*, vol. 13, Nova Science Publishers, USA. Chapter: 6, pp. 121-164

Okoye, K., Tawil, A. R. H., Naeem, U., Islam, S. & Lamine, E., 2016. *Using semantic-based approach to manage perspectives of process mining: Application on improving learning process domain data.* Proceedings of 2016 IEEE International Conference on Big Data (Big Data), Washington, D.C, pp. 3529-3538.

Kingsley Okoye, Syed Islam, Usman Naeem, Saeed Sharif, Muhammad Awais Azam and Amin Karami: *The Application of a Semantic-Based Process Mining Framework on a Learning Process Domain*. Intelligent Systems Conference (IntelliSys) September 2018. IEEE, 2018. Paper accepted for presentation at the Conference, London UK.

Okoye, K., Tawil, A. R. H., Naeem, U. & Lamine, E., 2015. *Semantic Process Mining Towards Discovery and Enhancement of Learning Model Analysis.* Proceedings of the 17th IEEE International Conference on High Performance Computing and Communications (HPCC), New York, USA, pp. 363-370.

Okoye, K., Tawil, A. R. H., Naeem, U. & Lamine, E., 2014. A Semantic Rule-based Approach Supported by Process Mining for Personalised Adaptive Learning. *Procedia Computer Science*, 37(C), pp. 203-210. In proceedings of the 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014) Halifax, Nova Scotia, Canada.

Okoye, K., Tawil, A. R. H., Naeem, U., Bashroush, R. & Lamine, E., 2014. *A Semantic Rule-Based Approach Towards Process Mining for Personalised Adaptive Learning*. In Proceedings of Algorithmic & Modelling workshops of the 16th IEEE International Conference on High Performance Computing and Communications (HPCC) Paris, France, pp. 929 – 936.

Okoye, K., Tawil, A. R. H., Naeem, U. & Lamine, E., 2016. *Fuzzy-BPMN Miner approach - Process Discovery Contest @ BPM 2016*. Technical Report, IEEE CIS Task Force on Process Mining discovery contest [1st Ed] in BPI workshop co-allocated with BPM 2016 Conference.

Chapter 1. Introduction

1.1 Motivation and Problem Statement

The need for novel approaches in design and integration of computational intelligence and technologies into everyday (e.g. business) processes, have sprout new insights and unceasing research investigations particularly on how to exploit such tools for use in improving the various organisational processes. In recent years, a common challenge with many business processes and operations has been on how to create techniques capable of providing platforms for exploring the additional, and most often, the monotonous tasks of managing the entire business process and quality of information by providing understandable and useful insights on the best possible ways to make the envisioned informations explicable in reality.

Nowadays, many organisations data collection systems and procedures for the captured data analysis is proving to be more and more complex. Such increasingly complexities spans due to the need for richer and more precise description of real-time processes (e.g. the business process) that allows for flexible exploration on how the various activities that makes up the business operations has been performed. Therefore, handling the large volumes of datasets extracted from the business operational processes and IT systems have raised intense debate within the industrialised community (mainly within the field of process mining) as well as many other IT experts, business process management and artificial intelligence companies. 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 and data complexities.

Moreover, researchers (Dou, et al., 2015; de Medeiros & Van der Aalst, 2009; Van der Aalst, 2016) have shown that a better way of attaining a closer look at any organisation's operational process is to consider the events log that are readily available in its process or database systems. Perhaps, an accurate exploration and/or analysis of the events log could provide vital and valuable information with regards to the quality of support being offered for the so-called organizations and their information knowledge-base at large. For example, revealing the underlying relations the process elements or resources share amongst themselves within the knowledge-base.

Recently, Process Mining (PM) (Van der Aalst, 2016) has become a valuable technique used to discover such meaningful information from the available event data logs. Moreover, the process mining field combines techniques from computational intelligence and data mining to process modeling and analysis, as well as several other disciplines to analyze large datasets.

Nonetheless, a shared challenge with most of the existing process mining techniques is that they depend on tags (i.e 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 in event log of the domain processes.

For that reason, this research work explores the technological potentials and prospects of using semantic-based approach to manage perspectives of process mining. In other words, the thesis addresses the challenges posed by the traditional process mining techniques by providing a method that focus on analysing the readily available events log based on concepts rather than the tags/labels in events log of the domain processes - through the proposed semantic-based process mining approach in this thesis that involves the combination of the *Process Mining* and *Semantic Modelling* techniques.

For all intent and purposes, this thesis refers to the *process mining* as such methods that pilots the structure of event logs by defining formats or viewpoints on the level of the systems and activities execution performance in relation to how a process has been previously performed and to determine the real process workflows (Van der Aalst, 2004; Ingvaldsen, 2011).

On the other hand, the *semantic modelling* and the resulting conceptual analysis techniques provide us with the opportunity to develop intelligent tools and algorithms that are capable of enhancing the resulting process models through explicit specification (often referred to as *conceptualisation*) (Balcan, et al., 2013; Montani, et al., 2017; Polyvyanyy, et al., 2017) in order to identify appropriate domain semantics and relationships among the process elements. Such an ontology-based approach is significant because, indeed, they involve semantic descriptions and/or reformulation of the meanings of the labels as well as their comparisons for the purpose of improving the usefulness and performance of the entire domain processes at large, particularly the information retrieval, processing, and extraction process.

Moreover, the common problem with process mining has been the technical focus of the event logs - where most of the existing process mining techniques depend on labels in the events

log information about the captured process to discover process models. As a result, the intellectual understanding or interpretation of the discovered models relates to the abstraction levels of the available event logs in view. 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.

Accordingly, the work in this thesis addresses the above challenges i.e. (i) lack of process mining 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 method of analysis displayed by the traditional process mining. The purpose is mainly as a way of providing formal structures for data used for process mining and enhancement of the analysis and/or interpretation of the derived process models.

Finally, the study applies the proposed method on a case study of learning process domain in order to demonstrate the usefulness of the proposed design framework - SPMaAF, algorithms formalizations and the resulting Semantic Fuzzy mining approach. This also includes comparisons and cross-validation of the proposed method against other existing benchmark algorithms and techniques used for process mining. Apparently, the proposed SPMaAF framework and its main application takes into account the different stages of process mining and analysis - from the initial phase of collecting and transformation of the readily available event data logs, to discovering of worthwhile process models. And then, expounds the traditional process mining method to semantically preparation of the extracted models for further analysis and querying at a more abstraction level. In terms of abstraction level, the thesis shows that the SPMaAF framework and the algorithms formalization is able to provide an easy way to analyse the datasets (i.e event logs and models) by allowing the meaning of the process elements to be enhanced through the use of property descriptions languages and schema - such as the Ontology Web-Rule Language (OWL) (W3C, 2012), Semantic Web Rule Language (SWRL) (Horrocks, et al., 2004), Description Logic (DL) queries (Baader, et al., 2003) in order to make available inference knowledge which are then utilized to determine useful patterns by means of the semantic reasoning aptitudes.

In essence, the proposed semantic-based approach in this thesis involves augmenting the informative value of the resulting process 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 short, the thesis shows how the work makes use of the case study of *Learning Process Domain* to demonstrate the capability of the SPMaAF framework and the sets of proposed algorithms by analysing the learning activity logs based on *concepts* rather than the *tags* (i.e attribute labels) found within the event logs about the process. In turn, presenting the process mining results at a much more conceptual level. The case study is based on running example of a *Research Learning Process*. Thus, the work describes how it uses the Research Process domain with focus on ascertaining by means of the conceptual method of analysis - the *successful* and *uncomplete* learner categories within the learning knowledge-base. Such conceptual method of analysis is described in more details in section 5.2 of this thesis.

In fact, the thesis implements the SPMaAF framework and the sets of semantically motivated algorithms in order to find out patterns/behaviour that describes or distinguishes certain entities (i.e process instances) within the learning knowledge base from some kind of others. Thereby, recognizes what *attributes* and/or paths some particular learners follow or have in common, or what attributes distinguishes the *successful learners* from the *uncomplete* ones. The purpose is not only to answer the specified questions by using the proposed approach in this thesis, but to show how by referring to the attributes (in a conceptual manner) and the application of semantic reasoning, it becomes easy to refer to a particular case (i.e. certain group of learners). Hence, the thesis focus is - on the use case distinction of the *Successful* and *Uncomplete* learners. In other words, the work utilize the events log about the *research process* to prove how the semantic-based method in this thesis is applied to represent and answer real time questions about the learning process in reality.

1.2 Research Context and Scope of Study

This section outlines the problems which the thesis pursues to address and how they are related in context of the research investigations.

1.2.1 Research Questions

Primarily, this research explores the best possible ways towards the:

> ***RQ1:*** *Use of process mining techniques to discover, monitor and analyse event logs about some domain process by discovering useful and worthwhile process models?* (Chapter 4 and 5)

Secondly, the research looks at:

> ***RQ2*** *How effective semantic modelling and reasoning methods can be used to enhance process mining analysis from the syntactic level to a much more conceptual level?* (Chapter 4 and 5)

Fundamentally, to address *RQ1*, the work uses the case study of *learning process* to show how one can efficiently mine and analyse the sets of unobserved behaviours/patterns (e.g. the process instances) that can be found within the process knowledge-base, and *RQ2* - through provision of formal structures (i.e. semantic representation) of the derived process models.

First, the research adopts the *process mining* technique to extract useful models from events log about the process domains. It is important to note for all intents and purposes, the work in this thesis refers to the *process domains* as specific class of problems which are identified and trailed to be resolved within a particular process in view or interest (e.g. case study of Learning Process described in this thesis). Accordingly, the extracted models from the process domains allows for the possibility to explore the process into multiple directions (i.e. process maps or workflow-nets) and to answer real-time questions about the process workflows or how the activities have been performed, and more importantly allows us to further model and hold inference reasoning to generate process improvement ideas along the way.

Secondly, through the *semantic modelling* approach, the thesis provides data inputs that are enriched with semantic annotations, and then links to concepts within an ontology designed for representing the derived process models in order to extract useful patterns by means of semantic reasoning. Moreover, the semantic modelling approach focuses on extracting useful information (semantics) from the sets of activities within the domain processes in order to generate rules and/or semantic assertions related to the process elements (e.g. how the activities have been performed), and in turn, enhances the informative values of the resulting process models and domain ontologies due to the rich semantic annotations and reasoning capabilities of the method.

Therefore, the two main research questions *RQ1* and *RQ2* forms the core validation study of this thesis and are addressed in Chapter 4 and 5. Grounded on those core validation areas and context of the thesis (i.e. process mining and semantic modelling) other research analyses has come up during the course of the research investigations directed towards achieving the main aim and objectives of the study as outlined in the next sub sections 1.2.2 and 1.2.3.

Thus, for the following research aim and objectives, the research has focus on ways towards achieving the proposed semantic-based approach which takes advantage of the rich semantics described in event logs of the domain processes, and trails to link them to concepts within a defined ontology in order to extract useful conceptual knowledge that allows for analysis of the event logs and models based on concepts rather than the event tags of the process.

1.2.2 Research Aim

The overall goal of the work carried out in this thesis is to:

"*extract streams of event logs from any given domain process (case study of the learning process) and describe formats that allows for mining and improved process analysis of the captured data*".

In other words, the focus of the research aim is to:

- apply process mining techniques to the domain of a learning process, and
- to provide real time semantic knowledge and understanding about domain processes (using the case study of the learning process) as well as useful strategies towards the development of process mining algorithms that are more intelligent with high level of conceptual (i.e semantic) reasoning capabilities.

1.2.3 Research Objectives

Practically, this thesis uses the case study of a learning process and data about a real-time business process to seek ways on *how* to do the following:

> ***RO1*** *Extract data from process domains to show how we semantically synchronize the event log formats for various process domain data* (Chapter 4)
>
> ***RO2*** *Semantically prepare the data through an ontology driven search for explorative analysis of a learning process activities and executions* (Chapter 4)
>
> ***RO3*** *Transform the extracted data into mining executable formats to support the discovery of valuable process models through the proposed technique for annotating unlabelled learning activity sequences using ontology schema/vocabularies* (Chapter 4 and 5)

RO4 *Provide techniques for accurate classification of unseen process instances (traces) within the process models* (Chapter 5)

RO5 *Monitor and enhance real-time processes through further semantic analysis of the discovered models.* (Chapter 5 and 6)

RO6 *Importance of semantics process mining to augment information value of data about domain processes: case study of learning process*. (Chapter 6)

The work carried out in order to achieve the target in *RO1* to *RO6* is centred on the two main components - (i) process mining, and (ii) semantic modelling - which are primarily focused on addressing the stated research question in *RQ1* and *RQ2*.

The aforementioned components and intents are characteristically based on the following four main application aspects:

(i) *Process Discovery:* to discover new process models based on the events data log about a *learning process* (section 5.1) and a *business process* (section 5.3.1) without any prior information on how those activities were performed.

(ii) *Conformance Check:* how much the data in the event logs matches the presented behaviour in the deployed models?

(iii) *Extension:* the need for both the process model and its logs to discover information that will enhance this model.

(iv) *Semantics:* even though the events data are captured and modelled with acceptable performance to accurately reflect the process executions, they are still limited for real-time analysis - because they lack the abstraction level required from a real world perspectives. Thus, the thesis show that analysis provided by using the traditional process mining techniques can be improved by adding semantic information to both the model and its logs.

For that reason, the semantic process modelling forms a key part of this thesis especially to guide the research efforts in achieving the main aim - which focus on extracting streams of event logs from a learning process and then describe formats that allows for mining and improved process analysis of the captured data.

Driven by such an effort, this work consequently studies *how* to propose the methods that highly influence and support the aforementioned objectives. Apparently, in an effort to

achieve the stated objectives i.e *RO1* to *RO6,* the research has made the presumption that *ontologies* can help in harmonizing the different datasets in line with the defined *concepts*. Moreover, the research also believes that *semantic annotations* and *reasoning* can help add useful conceptual *knowledge discovery* and *process querying* capabilities to the mining outcomes. Therefore, this research claims that through ontological modelling of the different process elements and use of the semantic rule-based approach that it is possible to make inferences over a process knowledge-base (e.g. case study of the learning process) that leads to automated discovery of useful learning patterns and/or behaviour.

Such proposals and claim is demonstrated in chapter 5 and 6 of this thesis. At the core of the proposed approach is the Semantic Process Mining and Analysis Framework (SPMaAF) presented in section 4.1, the sets of semantically motivated algorithms and its formalizations in section 4.3, and the Semantic-Fuzzy Miner implementation in section 5.3.2 which were all developed to help find answers to the research questions, and achieve the stated aims and objectives.

Consequently, the main aspects and motive for implementing the proposed SPMaAF framework, the sets of semantically motivated algorithms, and the Semantic-Fuzzy mining approach in this thesis is summarised as follows:

- *Event Logs* - to show how process mining can be applied to improve the informative value of learning process data.
- *Learning Model* - describe how improved process models can be derived from the large volume of event data logs found within the learning process domain.
- *Annotation* - describe how semantic descriptions (annotation) of the deployed model can help enrich the result of the learning process mining and outcomes through discovering of new knowledge about the process 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 (Semantic-Fuzzy Miner)* - 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 model - which in turn helps to address the problem of analyzing the learning process datasets based on concepts and to answer questions about relationships the learning objects (process instances) share amongst themselves within the knowledge-base.

1.3 Research Contributions

The main contributions of this thesis are summarised as follows:

1. Definition of a semantic-based fuzzy mining approach that exhibits a high level of semantic reasoning and capabilities (entire Thesis)

2. Design framework (SPMaAF) that highly influence and support the development of semantic process mining algorithms (section 4.1).

3. Sets of semantically motivated process mining algorithms that proves useful towards extraction, semantically preparation, and transformation of events log about any domain process (section 4.3 and 4.5).

4. A method for formal structures on how to perform and present process mining results in a more intuitive and easy way (section 4.2 to 4.8)

5. A process mining technique that is able to accurately classify and induce new knowledge based on previously unobserved behaviours (section 5.3)

6. An ontology-based system that is able to perform information retrieval and query answering in a more efficient and effective way compared to other standard logical procedures (section 5.2 and 5.3)

7. A series of case studies showing that semantic-based process mining can be used to enhance process mining results and analysis from the syntactic level to a much more conceptual level (Chapter 5)

8. Empirical evaluation of the impact of the Semantic Fuzzy mining approach and its outcomes compared to other benchmark algorithms for process mining (Chapter 6)

1.4 Research Methodology

The work in this thesis makes use of both the *qualitative* and *quantitative* research methods to carry out the investigations and proposals as represented in Figure 1.1. Moreover, the semantic-based process mining approach described in this thesis could be regarded as a fusion theory that is devoted to represent and analyse information in a qualitative and yet quantitative manner.

Figure 1.1 General overview of the Research Design and Methodology

As gathered in Figure 1.1, this thesis *qualitatively* shows by using a case study of the 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 processes through further semantic analysis of the discovered models. In addition, the research also *quantitatively* assess the level of accuracy of the classification results to predict behaviours of unobserved traces within the process

knowledge-base by determining which traces are fitting or not fitting the discovered model. In other words, the thesis also makes use of a training set and test log from a real time data about a business process from the IEEE CIS Task Force on Process Mining to implement and perform the experiments in this thesis.

In turn, the work employs both research methods for the purpose of validation and comparison by evaluating the level of impact and usefulness of the proposed research methodology in this thesis, validity of the classification results, and their influence compared to other existing benchmark algorithms and techniques that are closely related to the process mining field.

1.5 Thesis Structure

Chapter two presents background information in research areas related to the context and investigations of this thesis. It discusses and analyses the relevant theories and technological information within the field of process mining and semantic modelling that are fundamental for understanding the research outcomes and its contributions. This chapter also critiques existing works and identifies gaps that are addressed by the work done in this thesis.

Chapter three introduces the state-of-the-art and main components of the proposed method in this thesis. It presents the general approaches within the field of process mining research especially when applying the technique to any given process domain. The chapter starts by describing the event logs and the different sources of data for process mining. It continues by describing the relevant state-of-the-art - i.e tools, algorithms and techniques that are used to discover and analyse process models as well as the existing challenges and methods for managing the complications of the learned models. The chapter then concludes by looking at the ontological concepts/schema and how they can be applied to represent the discovered informations and/or models in a formal structure or manner.

Chapter four presents the main components of the SPMaAF framework which this study has developed for implementation of the proposed semantic-based process mining approach in this thesis i.e. the semantic fuzzy miner. It starts by describing in details the design specifications and methodology, and specifically, demonstrate the main tools and ideas behind the semantic-based approach. In addition, the chapter highlights the main functionalities offered by the SPMaAF framework and the consequential main application areas. It continues with a description of the semantic-based motivated process mining

algorithms and its formalizations, methods for semantically annotating the discovered process models, and the main components of the resulting domain ontologies - particularly the ontology schema and the key functions that allows the definition of the different domain classes, objects and data properties, including the process used to classify and query the resulting model. Finally, the chapter summarizes the presented method and components in relation to the work done in this thesis.

Chapter five introduces the use case studies and implementation scenario of the semantic-based approach (using the case study of the learning process and data about a business process) that supports the mining of real-time processes and events data at a much more conceptual level. The chapter starts by demonstrating the main functionalities of the SPMaAF framework and the resulting Semantic Fuzzy Miner (SFM) application including the essential requirements and different phases for implementing each of those component, and then finalise by presenting the main results in details.

Chapter six presents the cross-validation of the developed approach in the thesis. It qualitatively and yet quantitatively evaluate and discuss to what extent the work has achieved the target aim of the research, and addressed the research questions with regards to the stated objectives. The chapter also compares the outcome of the study to other relevant state-of-the-art benchmark approaches used for process mining, as a way of indicating its impact and contributions to knowledge.

Chapter seven concludes the thesis by pointing out the research achievements, novelty of the research findings and contributions including its influence and impacts. In addition, it provides an overall discussion on the research limitations and threats to validity including a road map on topics that could be investigated as future work.

Chapter 2. Literature Review and Background Information

This chapter of the thesis discusses and elaborates on some of the challenges and potential technologies used for process mining and the semantic modelling techniques. It describes some of the relevant, related areas in this research especially in terms of process discovery and pattern mining, information retrieval and extraction, semantic-based process mining and ontologies, interrelated data mining techniques such as the classifications method of data analysis as well as the fuzzy logics. It continues by highlighting the main components and mechanisms behind the development of the SPMaAF framework in this thesis and then describes the different challenges when handling and integrating those essential components. In other words, the chapter describes the tools, models and techniques that could be applied within the semantic process mining domain as described in the work in this thesis. In essence, the work looks at background informations that are essential for understanding the context and field area of this research. The idea of embracing process mining and applying its methods within the domain of learning process is discussed at first. It continues by describing the application of intelligent and adaptive educational process mining and other modelling approaches devoted to improving process analysis by acquiring and representing abstract knowledge about the actual processes in reality. The chapter also looks at the broader term of Business Intelligence (BI) and many other overlapping terms, such as the Business Process Management (BPM), Business Activity Monitoring (BAM), Process Aware Information Systems (PAIS) etc. that combines tools or methodology which are all aimed at offering useful information and insights that can be utilized to support real-time processes or decision making. Finally, the chapter discusses the Semantic Web Search technologies and Ontology-based Information Extraction (OBIE) systems which have also been seen as very useful approach used to support the extraction, mining and analysis of processes by influencing the level of real world (semantic) knowledge that can be derived from the readily available datasets about the domain processes in view. And then concludes by looking at the Process Querying (PQ) scheme, an emerging technique that concerns automatic methods for managing repositories of models of observed and/or unseen processes with the goal of transforming the process-related informations into decision making capabilities. Lastly, the summary section of this chapter includes a table listing the main relevant work in this area of research and are grouped by their various application domains which are most closely related to the process mining and semantic modelling techniques.

2.1 Related Areas and Existing works

Most of the existing techniques for analysing large growing knowledge bases focus on building algorithms to help the knowledge base automatically or semi-automatically extend. According to (Miani & Hruschka Junior, 2015) vast number of such systems constructing large knowledge bases continuously grow, and most often, they do not contain all of the facts for each process instance or elements, thereby, resulting in missing value datasets. Consequently, a well-designed information retrieval or mining system should present results and discovered patterns in a formal and structured format qua being interpreted as domain knowledge and to further enhance the existing knowledge base (Dou, et al., 2015).

Information Retrieval and Extraction: according to (Cairns, et al., 2014) one of the challenges with 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 the said problems, the authors (Cairns, et al., 2014) 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 (Cairns, et al., 2014) propose a semi-automatic procedure used to associate semantics to training labels. They used the Ontology Abstract Filter plug-in in ProM (Verbeek, et al., 2011) as input to a semantically annotated 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) (deMedeiros, et al., 2008) file format, and subsequently perform a control-flow mining using the Heuristic Miner algorithm proposed by (Weijters & Ribeiro, 2010; Weijters, et al., 2006) in order to extract the process models based on the concepts that has been derived.

Semantic-based Process Analysis: indeed, methods for semantic process mining and analysis focuses on information about resources hidden within a process knowledge-base, and how they are related (deMedeiros, et al., 2008; de Medeiros & Van der Aalst, 2009; Okoye, et al., 2016; Okoye, et al., 2017; Jareevongpiboon & Janecek, 2013). 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 for analysis of the extracted event logs based on concepts rather than the event tags or labels about the process. Currently, 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 (deMedeiros, et al., 2008; de Medeiros & Van der Aalst, 2009; Okoye, et al., 2017; Jareevongpiboon & Janecek, 2013). In (Okoye, et al., 2017; Okoye, et al., 2016), we show how semantic annotations and reasoning can be used to provide a more improved analysis (i.e enhancements) to process models and event logs through concept matching (i.e. ontology classifications). Specifically, in (Okoye, et al., 2016) we perform the semantic modelling and integration of the resulting process mappings with annotated terms and then present 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) (Horrocks, et al., 2007; W3C, 2012) and Semantic Web Rule Language (SWRL) (Horrocks, et al., 2004). Indeed, reasoning on the ontological knowledge plays an important role in semantic representation of processes (Calvanese, et al., 2017). 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 & Kamber, 2004; Han, et al., 2011) *classification* is one of the most common data mining (DM) technique that aims at finding models or functions that describes or distinguishes data classes or concepts. One of the main benefits of applying such DM technique in context of this thesis is to help annotate and explain the classification labels in line with the set of relations defined in an ontology especially for use in semantic enhancement of the captured datasets. 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., 2014; Balcan, et al., 2013). For instance, the authors in (Balcan, et al., 2013) 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., 2008) argue that classification is a fundamental task for a lot of intelligent applications, and that classifying through logic reasoning may be both too demanding and frail because of inherent incompleteness and complexity in the knowledge bases. However, the authors observe that those 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 resolve process modelling problems (Okoye, et al., 2016) using process description logics particularly those based on classification, clustering and ranking of individuals. Explicitly, the works in (d'Amato, et al., 2008; Okoye, et al., 2016) shows 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.

Pattern Discovery: accordingly, the authors in (Elhebir & Abraham, 2015) notes that pattern discovery algorithms uses 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 (Elhebir & Abraham, 2015) 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 algorithms) could perhaps be considered as a general solution for pattern discovery because they obtain better results compared to a single classifier as long as the components are independent or have diverse outputs.

Fuzzy Logics vs Fusion Theory: in principle, (Baati, et al., 2016) propose two kinds of possibilistic classifiers for numerical data namely: 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. Even more, according to the authors in (Baati, et al., 2016; Baati, et al., 2017) the Possibility theory, introduced by (Zadeh, 1965) and further developed by (Dubois, et al., 1988) is a fusion theory based on fuzzy sets theory and devoted to represent and combine imperfect information in a qualitative or better still 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., 2013). Reference (Baati, et al., 2017) also 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 such problem, the authors (Baati, et al., 2017) states that the Generalized Minimum-based (G-Min) algorithm proposed

in (Baati, et al., 2016) can be employed to avoid the 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 in (Baati, et al., 2017) the G-Min algorithm requires the matrix Π of possibilistic estimates and it's based on two main steps: the first, aims to build a set of possible decisions, whilst, the second aims to filter those set in order to find a final class with a high score of reliability (Baati, et al., 2016). 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.

Likewise, the work in this thesis presents a semantic-fuzzy mining approach that targets through the conceptualization method to turn process models and analysis into a classification task with a *training set* and a *test set* (Carmona, et al., 2016; Van der Aalst, 2016) 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) the model. Indeed, the utilized approach aims at making use of semantic annotations to link process instances found within the event data log and models with concepts they represents in an ontology. The purpose of the semantic annotation is to seek the equivalence between the *concepts of the fuzzy models* derived by applying the fuzzy miner algorithm on the datasets and the *concepts of the defined enriched domain ontology*.

Lastly, the method proposed in this thesis as opposed to the other benchmark techniques and/or algorithms within the field of process mining, makes use of semantics of the sets of activities within the domain process and models to generate rules and events relating to task, to automatically discover and ascertain the various process instances. The use case scenario of the learning process in this thesis together with the effort to address the aforementioned challenges with process mining techniques and analysis forms part of the contribution of this thesis. Interestingly, this kind of knowledge can 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 the proposed approach in this thesis is the ability to describe the semantics behind the labels within the model and events log about the domain processes (e.g. the learning process) considered useful for discovery of new knowledge about any domain

process. 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 those 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. To this end, the application of ontologies for any process management system should focus on generating new or improve existing methods, tools and techniques that support the different phases of the process management and its analysis.

Accordingly, the next sub sections of this chapter looks at the main fields and background information that are pertinent towards the development of the semantic-based process mining approach in this thesis by discussing the tools, models and techniques that could be applied within the domain area of the research.

2.2 Process Mining (Data Science in Action)

" ... I have come to feel that my central interest is in data analysis, which I take to include, among other things: procedures for analyzing data, techniques for interpreting the results of such procedures, ways of planning the gathering of data to make its analysis easier, more precise or more accurate, and all the machinery and results (mathematical) statistics which apply to analyzing data…" John Tukey (1915 – 2000) *wrote in 1962* (Tukey, 1962)

Wil van der Aalst (Van der Aalst, 2016), one of the most influential and the best known BPM researcher, in his recent book (Van der Aalst, 2016), referred to process mining as *"data science in action".* The author notes that *data science* has emerged as a new discipline in recent years, and further mention that the "*data science field includes data extraction, data preparation, data exploration, data transformation, storage and retrieval, computing infrastructures, various types of mining and learning, presentation of explanations and predictions, and the exploitation of results taking into account ethical, social, legal, and business aspects"* (Van der Aalst, 2016).

Accordingly, (Van der Aalst, 2016) describes *process mining* (seen as the missing link between model-based process analysis and data-oriented analysis techniques that allows for extraction of non-trivial information from event logs) as one of the main mechanisms of "data science". The author opines that process mining has the capacity to provide means towards bridging the gap between *data science* and *process science*. Apparently, *Process science* has

emerged due to the process-perspective that is missing in most *big data* initiative and the curricula of *data science*. Besides, the author in (Van der Aalst, 2011; Van der Aalst, 2016) argue that the data logs extracted and stored in many organisations IT system must be utilised to enhance the end to end process in reality by focusing on analysing the unseen behaviours based on the information that are present in the logs, thus, emergence of *process mining*.

Process mining (PM) research started at the Eindhoven University of Technology (TU/e) in 1999, and was first proposed by Wil van der Aalst (Van der Aalst, et al., 2003; Van der Aalst, et al., 2004). According to (Van der Aalst, 2016) as of then, there were limited availability of event logs, and the early methods used to perform process mining tasks at that time were exceptionally ineffective and naive. Interestingly, for the past few decades, the process mining tools and approaches has undisputedly matured (Maita, et al., 2017) because events data logs has become ever more available. Moreover, progress has been spectacular in the field and the technique is currently being supported by different tools and algorithms such as the one introduced in this thesis. Besides, the authors in (Maita, et al., 2017) thinks that the process mining research and challenges, such as balancing among robustness, simplicity, accuracy and generalization, would benefit from a larger use of such techniques.

Furthermore, whilst the initial attention was primarily on the *process discovery* method, process mining field have significantly widened - e.g. conformance checking, operational support, and multi-perspective process mining which has now grown into fundamental part of many tools and approaches that supports the extraction and interpretation of processes in reality. Particularly, ProM (Verbeek, et al., 2011) one of the leading process mining tool.

Nowadays, several organisations have focused on applying the PM technique to different aspects of their business processes. Moreover, the application of the process mining techniques are not only or limited to business processes, but also provides new means to discover, monitor, and enhance any given process domain. (De Leoni & Van der Aalst, 2013; Van der Aalst, et al., 2012). According to (Van der Aalst, 2011) there are two main drivers for the growing interest in process mining. First, data about many organizations business process are captured and stored at an unprecedented rate. Secondly, there is ever rising need to improve and support business processes in competitive and rapidly changing environments.

In short, process mining have likewise proved its relatability and application in some other field areas including: Health care (Rojas, et al., 2016), Government sectors (Van der Aalst, 2016), Banking and Financial industries (Jans, 2011; Van der Aalst, et al., 2010), Educational

organizations and settings (Cairns, et al., 2014; Okoye, et al., 2016), Airlines and Transportation industry (Van der Aalst, 2016) , ICT and Cloud Computing (Chesani, et al., 2016) etc. Indeed, the PM techniques uses event data from any these process domain to discover process models, perform conformance checking of the discovered models, analyse deviations, and even more, extend and predict future outcomes and/or developments.

Actually, many explanations of the process mining notion has been propose in literature (Van der Aalst, 2011; Cairns, et al., 2015; Ingvaldsen, 2011; Van der Aalst, 2016).

Reference (Van der Aalst, 2011) refers to the PM - as a young research field that makes use of the data mining (DM) technique to find out patterns or models from event logs, and predict outcomes through further analysis of the discovered models. According to the author (Van der Aalst, 2011; Van der Aalst, 2016) PM means extracting valuable, process-related information from event logs about any domain process.

The authors in (Cairns, et al., 2015) also mentions that the process mining term is concerned with analysis of the captured datasets (i.e. events log) from a process-perspective. Reference (Ingvaldsen, 2011) states that as soon as a particular process (e.g. business process) is being supported by some form of IT system, its operational transactions or activities executions can then be observed or recorded in the form of event logs. Likewise, references (Greco, et al., 2006; Van der Aalst, 2011) mentions that the process mining notion is an attempt towards extraction of meaningful and non-trivial information from recorded event logs.

Without any doubt, references (Adriansyah, et al., 2011; Ingvaldsen, et al., 2005; Ingvaldsen, 2011; Van der Aalst, et al., 2012) are even more specific about the focus of process mining in extracting, validation and extension of process models explicitly, and as such, also groups process mining into the following three types of process mining as discussed in the next sub section of this thesis. Thus, for the aforementioned reasons and explanations, *events log* could be utilized to perform the following PM techniques (Van der Aalst, 2016):

(i) Process Discovery
(ii) Conformance Checking, and
(iii) Model Enhancement

2.2.1 Process Discovery

The lion's share of attention in process mining has been devoted to *process discovery* i.e. extracting process models, mainly business process models from an event log (Carmona, et al., 2016). Process discovery has been lately seen as the main significant and furthermost challenge logically allied to the process mining term (Carmona, et al., 2016; Van der Aalst, 2011). Process discovery techniques aims to automatically construct process models, e.g., BPMN, Petri-nets, Fuzzy models etc. (Van der Aalst, 2016) from event log about a process, and describes causal dependencies between the individual activities as performed.

A typical *process discovery* method takes (as input) recorded event logs, and then produce (as output) a model without any prior information on how the activities has been formerly performed. Besides, in settings where the data sets (i.e. event logs) includes further information about resource (e.g. roles), it is also possible to discover resource-related models. For instance, a shared network representing how employees works collectively or collaborate within a particular organisation. Principally, one can make use of the process discovery methods to obtain models that describes reality.

Various algorithms have been developed in current literature with the capability of performing process discovery tasks, namely: the α-algorithm, Fuzzy miner, Heuristic miner, Genetic miner, Inductive miner (Van der Aalst, 2016) etc. Actually, those algorithms have also been made available in existing process mining tools such as the ProM (Verbeek, et al., 2011) and Disco (Rozinat & Gunther, 2012) etc. Moreover, some of the main application benefits, impact and limitations of the process discovery algorithms are discussed in details in chapter 3 of this thesis.

2.2.2 Conformance Checking

The *conformance checking* is the second type of process mining technique. The method focuses on determining (assessing) how fit the discovered process models describes the actual observation in the event logs (Ingvaldsen, 2011). In other words, a conformance check and analysis technique references an a-priori (i.e existing) process model and compares it with the events log of the specific (i.e. the same) process. Thus, such analysis is performed in order to check if in reality, the recorded data logs conforms to the deployed models (Munoz-Gama & Carmona, 2011; Adriansyah, et al., 2011; Rozinat & Van der Aalst, 2008; Weerdt, et al., 2011; Fahland & van der Aalst, 2012).

For instance, the output a conformance checking technique may imply that the discovered process model perhaps do not describe the executed process as supposed in reality, or is being executed in a different order (Fahland & van der Aalst, 2012; Van der Aalst, 2011). It could also mean that some of the process instance (i.e. individual activities) as observed within the discovered model are skipped in the event log, or may be the logs consist of actions (i.e., events) that are not necessarily defined by the process model (Fahland & van der Aalst, 2012).

Therefore, a well performed conformance check is relevant and significant especially from a business objective alignment or auditing perspective. For example, it is possible that the recorded logs could be reiterated (i.e model replay) against the derived models in order to discover unexpected deviation or bottlenecks that may impact the business process in general. In addition, the conformance checking could be utilized to measure the fitness of the models discovered by the PM tools, and could also be used to perform the repairing of the process models in reality.

Many existing PM algorithms capable of performing the conformance check has been proposed e.g. the Inductive Visual Miner (Leemans, et al., 2014), LTL checker (de Beer, 2005) etc., and has been applied also in different business process settings (Adriansyah, et al., 2011; Van der Aalst, 2011). These algorithms can be found in some of the existing process mining tools mainly ProM (Verbeek, et al., 2011).

Generally, in many settings the conformance checking algorithms targets to achieve the following functions:

- ✓ Business Alignment and Auditing
- ✓ Token Replay
- ✓ Comparing Traces or Footprints
- ✓ Model Repair
- ✓ Assessing Process Mining Algorithms, and
- ✓ Connecting Event Logs and Process Models

Most often conformance check is performed to show the replaying semantics (i.e token replay) for models in regards to the four quality criteria's - *Fitness, Generalisation, Precision,* and *Simplicity* (Van der Aalst, 2011). The research describes and show the need for those four quality criteria in chapter 5 of this thesis. Particularly, the method indicates how the fitness of the available events data logs are being measured in a qualitative and quantitative manner. For instance, the level or extent of behaviours within the event logs which happen to be actually possible according to the discovered process models.

In short, the conformance checking technique is utilised to balance between *traces* (i.e. observed behaviours or patterns) that are *overfitting* or *underfitting* the actual process as performed in reality (Carmona, et al., 2016; Fahland & van der Aalst, 2012).

2.2.3 Model Enhancement

Given the drawbacks and challenges identified with the previously aforementioned types of PM techniques, the last type of process mining (i.e. *model enhancement*) comes into play. The *model enhancement* aims at augmenting the process models with additional information extracted from the event logs with focus on extending or improving the actual behaviours or the learned processes as captured from the original logs (Ingvaldsen, 2011).

Therefore, whilst the *conformance checking* techniques measures the log fitness (e.g. alignment) between the process models and reality, the *model enhancement* trails to extend or completely change the original (a-priori) model. The method is used to manipulate and maintain compliance and/or to quantify deviations by making use of the informations that have been discovered (through the process discovery) and aligned (conformance checking) about the real process models learned from the event logs. Consequently, model enhancement comes in to play from two perspectives (Van der Aalst, 2016; Van der Aalst, 2011):

1) *Model Repair, and*
2) *Model Extension*

Moreover, *process discovery* and *conformance checking* approaches do not only limit their application to control-flows (or workflow management), but also allows for additional perspectives to be added to the methods by extending the process models, thus, the *model enhancement.*

To sum up this section of the thesis - various perspectives may be considered orthogonal to the three different type of PM techniques (i.e. process discovery, conformance check, and model enhancement) namely: *Control-flow perspective*, *Case perspective, Organizational perspective*, *Time perspective*, *Operational support*, etc. (Van der Aalst, 2011).

Clearly, the *process mining* plays an important role in many organisations. It spans its technical application from the fields of *data science* and *business process management* (BPM), and as such, we assume that to perform any process mining task that there has to be some kind of recorded data from an actual (i.e real-life) process.

Using the context of this thesis as example, the work shows in Figure 2.1 that the first step (i.e. starting point) for any given *process mining* project is to capture the *event data* logs about the process (e.g. learning process), and then generate *process model* to show in details how the activities has been performed and to reveal interesting connections between the different process elements (process instances). In consequence, the process mappings allows for an enhanced *analysis* and/or *extension* of the discovered process model. Indeed, such application of PM technique as described in Figure 2.1 is what the research has used to support the integration and implementation of the proposed semantic-based process mining and analysis (SPMaAF) framework and its real-time application in this thesis.

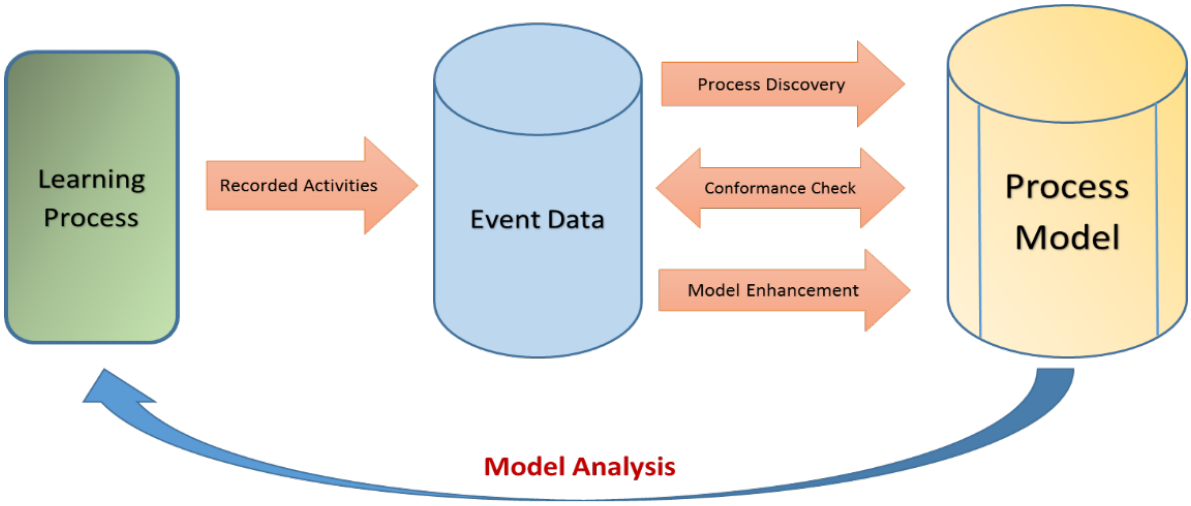


Figure 2.1 Application of process mining techniques

2.3 Educational Process Mining (EPM)

The need for relatable *automation and management of learning processes* in reality has led to increasing demand for methods and tools that supports the accumulative large volumes of data that are extracted from the various data-sources, stored in different forms, as well as in diverse granular levels in several educational organizations (Bogarín, et al., 2014; Trčka, et al., 2010). Indeed, those captured datasets can be exploited by process analyst and the owners to understand the behaviours or patterns of the intended users, including their level of performance and/or achieved goals.

Henceforth, such search for "*exploration and analysis, by automatic or semi-automatic means of managing the large quantities of datasets in order to discover meaningful patterns or*

rules" (Ingvaldsen, 2011) motivates the increasingly research interest in application of the process mining techniques in educational settings (Cairns, et al., 2014).

Educational Process Mining (EPM) is a new domain area within the wider context of business process management that aims to apply *process mining* techniques to find out user patterns or models from the captured sets of educational data, and then pursues to predict outcomes through further analysis of the discovered models. Thus, EPM refers to the application of process mining techniques within the education domain (Trčka, et al., 2010; Cairns, et al., 2014; Bogarín, et al., 2018). Reference (Bogarín, et al., 2018) are even more specific about the application areas of EPM. According to the authors, EPM is one of such areas that the process mining technique is currently being applied and is gaining attention in recent years. The authors notes that EPM means the application of process mining to raw educational data by taking into account the end to end processes rather than local patterns, as opposed to the Educational Data Mining (EDM) (Baker & Yacef, 2009; Dou, et al., 2015) which tends not to be process-centric and do not focus on event data (Van der Aalst, 2016) e.g. the rows (instances) and columns (variables) of a typical data file which does not have any meaning.

A number of researchers have directed their work towards the use and application of this new advanced aspect of process mining within the educational settings (Bogarín, et al., 2014; Cairns, et al., 2014; Trcka & Pechenizkiy, 2009; Trčka, et al., 2010; Okoye, et al., 2017). According to (Cairns, et al., 2014) EPM emerges from the Educational Data Mining (EDM) discipline, and the drive for its incentive is primarily to discover, analyse and improve the educational process based on the hidden informations within the databases or events log recorded by the existing systems (e.g. Schools, Colleges, Universities, or Professional Training Institutions).

EDM vs EPM: the difference between Data Mining (DM) and Process Mining (PM), is that whilst DM aims to mine and analyse event logs at *data-levels*, PM targets to mine and analyse event logs at *process-levels* (Holzhüter, et al., 2013). Likewise, those levels of analysis also applies to the context of Educational Data Mining (EDM) and Educational Process Mining (EPM). Hence, whilst EDM aims to mine and analyse educational data at *data-levels*, EPM pursues to mine and analyse educational data at *process-levels.* Nonetheless, (Bogarín, et al., 2018) notes that both the EDM and EPM apply specific algorithms to data in order to discover hidden patterns and/or relationships. In fact, whichever tool one chooses to adopt, the key focus should be on achieving the purpose of adopting such techniques.

For example, one may pursue and focus on developing approaches that reproduces or analysis learner activities especially to improve learning efficiency and/or provide useful knowledge about how the individual process elements interact with each other within the learning knowledge base such as the one described in this thesis. Besides, the enhancement of learning processes and its effectiveness and performance is capable of providing sound arguments or point of analysis to perceive and/or identify the benefits for different learning scenarios.

Moreover, the authors in (Cairns, et al., 2015) studies the potential benefit of the process mining techniques within the educational domain by proposing a two-step clustering approach to extract the best training paths depending on an employability indicator. According to them the embracing of abstract filtering or clustering technique could assist in reducing complexities within the learned models to improve the application of process discovery and analysis techniques in an educational setting.

However, (Cairns, et al., 2014) observes that existing methods for extracting models within the educational processes are limited to some extent because the approaches depend on traditional process mining techniques that are purely syntactic in nature (i.e. based on the labels in event logs) to discover the process models. In so doing, the developed systems do not technically gain from the real world knowledge that describes the processes as performed in reality, and as consequence, the actual *semantics* behind the event log remains missing and sprouts the need for who have to interpret them.

Therefore, in practice, *process mining* tools poses some certain issue of *semantics* that limits its efficiency when handling the large volume of events log from the complex educational systems as well as their analysis at conceptual levels (de Medeiros & Van der Aalst, 2009; Cairns, et al., 2014). However, (Cairns, et al., 2014) thinks that the *semantic process mining* method appears to be a promising area that can be explored in order to resolve those issues of *understanding* the learning patterns or trace heterogeneity, and as such, to extract streamlined models that fits or represents the actual processes as performed in reality. Moreover, the authors (Cairns, et al., 2014) believes that *semantic annotation* of the captured datasets can also be utilised to address the challenge of interpreting the processes in question. Henceforth, to benefit from the actual semantics behind those event tags or labels, *semantic process mining* (de Medeiros & Van der Aalst, 2009; deMedeiros, et al., 2008) which enforces mining and analysis of processes at a more conceptual levels has to be employed.

In view of that, the work in this thesis pursues to provide a semantic process mining approach directed towards the discovery and enhancement of process models with emphasis on the case study of the learning process domain. The work shows using data about a learning process - how event data from various process domains particularly the educational process 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. The aim is not only to extract streams of event logs from the learning execution environment, but to also define formats that allows for mining and improved analysis of the captured datasets by semantically annotating the process elements with concepts they represent in real time, and then linking them to an ontology built for representing the learning processes. The method proves to allow for analysis of the extracted event logs based on concepts rather than the event tags of the process. Moreover, the semantic-based analysis allows the meaning of the learning objects and the resulting models to be enhanced through the use of property characteristics and classification of discoverable entities, to generate inference knowledge which are then used to determine useful patterns and improve analysis of the resulting models to a much more conceptual level as opposed to the syntactic nature of analysis displayed by the traditional process mining techniques.

2.4 Intelligent and Adaptive Educational Learning Systems (IAELS)

Intelligent and adaptive educational learning systems (IAELS) is classed under the umbrella of Educational Learning Systems (ELS) (Peña-Ayala, 2013). ELS represents the broad range of computer-based approaches dedicated towards the support and widespread of educational services for learning through the use of information and communication technology (ICT). The development and acquisition of ELS takes into account Artificial Intelligence (AI) or even the latter, Augmented Intelligence systems (AIs) i.e through a combination of computer generated methods and human input - to show the extent of intelligent functionality when they are used to support educational services. For example, the process of acquiring and representing knowledge, making inferences and/or automation of the learning process in real-time.

With such requirements at the central motive of educational services, IAELS could be defined as ELS technologies or tools that exhibits some kind of intelligent and adaptive functionality (Peña-Ayala, 2013). Many researchers and practitioners in the field of education, ICT or AIs have carried out investigations, and in turn, propose systems, methods and approaches that

intelligently provides enhanced services for the learners. According to (Peña-Ayala, 2013) amongst many of the contributions are the main key components of IAELS systems as follows: *User Modelling*, *Content Representation, Virtualization,* and *Metacognition and Case studies Application*. A typical example, is the work in reference (Tadlaoui, et al., 2013) which demonstrates the importance of the IAELS to perform learning content management and adaptation, most notably, in terms of *Knowledge on the Entity Relationship model* i.e. understanding domain independent knowledge useful in identifying the different users and adaptation based on shared properties.

Actually, with regards to the context of investigations of this thesis, such parameters could be raised to be paramount to the idea of using the process mining in combination of semantic modelling techniques to manage perspectives of the learning process domain. Especially when the available data is made up of such attributes, and the proposed approach in this thesis is also developed to link to those concepts as well-defined within the process models.

Prediction: another important feature of the IAELS is the *predictive* aspect of the systems they support. Reference (Peña-Ayala & Sossa, 2013) notes that a pre-emptive educational model seeks to predict possible future activities in order to accomplish better student's training as well as overcome likely issues. Interestingly, the authors in (Peña-Ayala & Sossa, 2013) propose a Casual and Fuzzy Student Model (CFSM) which describes several user attributes to anticipate the sequencing of activities for the students through the acquisition of domain knowledge about the students themselves. For example, the process of determining the sets of individuals that are classified as *successful* or *uncomplete* learners within the learning knowledge-base as described in section 5.1 and 5.2 of this thesis.

The main component derived here and pertinent for use in this thesis, is that those user attributes are semantically described in form of concepts within ontologies. Thus, the described concepts sets casual relations amongst the individual elements, and as such represents a certainty on how an attribute utilizes the properties and/or triggers another attribute. Moreover, those descriptions of attributes and casual relationship consist of fuzzy-values and rule-base. In turn, the fuzzy rule-based system could then be analysed using mining set of rules to interpret and predict future behaviours. For example, the incorporation of the *Fuzzy Student Model* in (Sevarac, 2006) that enables classification of users based on qualitative observation of the ascertained properties.

Fuzzy Mining and Reasoning: perhaps, instead of using a *neuro fuzzy* or *probabilistic reasoning* method, the CFSM (Peña-Ayala & Sossa, 2013) makes use of a *fuzzy casual reasoning* method to deal with the qualitative observations or information. The main discussion here, is that the approach organises *ontologies* which are used in identification and definition of the meanings of the concepts, casual relations, fuzzy rules bases, universe of discourse and other relevant items through the use of the ontology web-rule language (OWL) declaration sentences such as *Classes*, *Datatype Properties, Functional Properties,* and *Process instances* etc. Moreover, ontology also sits at the heart of the work in this thesis, and a description of how we have utilized the schema is expounded in later chapters (Chapter 3 and 4) of this thesis.

Learning Process Automation: the authors in (Holzhüter, et al., 2013) states that *process mining* is the discipline capable of offering useful tools and sets of concept to follow such optimization and adaptation of learning process. Besides, automated learning process means supplying simulations, contents and interactive maps in a unified and well-structured manner, and this is where the process modelling in combination with process mining techniques is capable of providing useful method for better identification and analysis of learning problems. Thus, the main reason why the proposed approach in this thesis is grounded on process mining and semantic process modelling.

Furthermore, in a survey correspondingly conducted by (Baker & Yacef, 2009) adapted from (Dżega & Pietruszkiewicz, 2013), the authors examines the application of some DM methods in e-learning based on the *Proceedings of Educational Data Mining* in 2008 and 2009. The results of their work reveals the following statistics as shown in Figure 2.2.

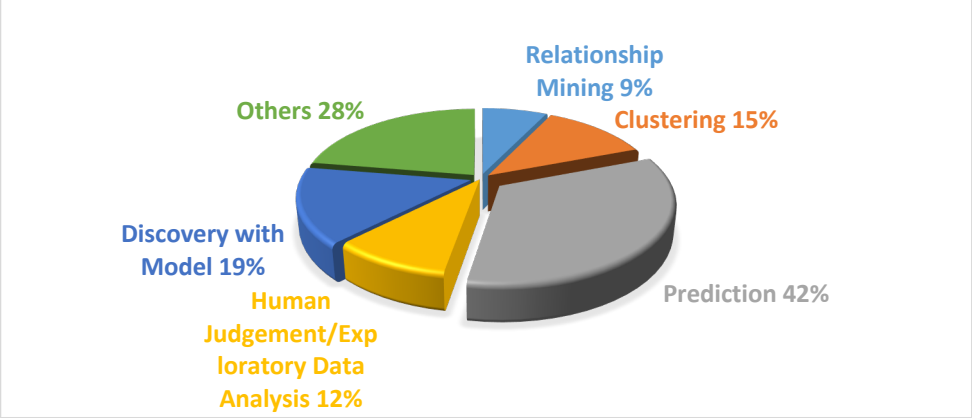


Figure 2.2 Application of Data Mining in e-learning settings

It is important to note that the numbers in Figure 2.2 do not add up to 100%, as some sources uses multiple methods that were counted in multiple categories. Moreover, the authors (Dżega & Pietruszkiewicz, 2013) observe that largest group of the DM applications were aimed at *observations, control, and prediction* of user behaviour or performance. Nonetheless, the authors suggests that learning process automation irrespective of the settings in which they are being used in (be it in the educational settings or any other domain), should be offered as specialized computer-generated services and such analysis have to also integrate some form of business sub-processing tools such as the process mining techniques.

Learning Process Management vs BI: obviously, process mining belong to such collections of tool within the Business Intelligence (BI) as well as other overlapping terms such as BAM, BPM, PAIS etc. (Van der Aalst, 2011; Van der Aalst, 2016) that uses, if not all, the data mining techniques described in Figure 2.2 to find out patterns or models from event logs (e.g. the learning process), and predict outcomes through further analysis of the discovered models.

For example, the authors in (Cesarini, et al., 2004; Perez-Rodriguez, et al., 2008) introduces various approaches for learning pattern control through a workflow management system (WFM) but does not relate a devoted strategy for the process analysis such as the process mining technique. On the other hand, (Nguyen & Phung, 2008) sparks the potential benefits of enhancing learning process model particularly within the context of Adaptive Educational Hypermedia Systems (AEHS) by constructing a framework for exploiting learner models. Their method combines process mining techniques with the concepts of learning patterns.

Learning Patterns Discovery: similarly, the authors in (Trčka, et al., 2010; Pechenizkiy, et al., 2009) have worked on approaches that applies the process mining technique in context of e-learning process. The authors analysed and points out tools that are being used to perform process mining tasks which qualifies better in support of e-learning processes. Whereas, the authors in (Holzhüter, et al., 2013) argues that a way of supporting learners within an e-learning setting is to adopt the combine approach of using the process mining techniques with concepts of the discovered learning patterns. In turn, the learning patterns describes how leaners sees, interacts or responds to a learning process, and therefore measures the differences between individual properties, or better still, helps provide useful information on how to recommend certain paths for new learners that may have or share the same properties.

Learner Models: typically, a learner model records learners behaviours, and as such represents and interprets to certain extent the characteristics of the learners within the learning

knowledge base (e.g. user ID's, task preferences, roles, abilities, categories, date and timestamps etc.). These representative properties of the learners is significant, and could be related to some kind of particular user-subgroups (e.g. by means of the classification method) within the learning knowledge base. In other words, such procedures classifies the learners as belonging to a specific or even different groups, as well as integrates their usual properties within the domain models.

Learning Process Analysis and Classifications: according to (Holzhüter, et al., 2013), to perform the classification or assumptions of learner patterns, datasets need to be extracted from a learning system (i.e the data sources). Moreover, in view of carrying out the process analysis - the extracted datasets needs to be prepared and transformed into formats that allows for the pattern recognition to follow. Consequently, prior to performing any other further analysis or use of the discovered informations, an evaluation of the resulting learning patterns is carried out, and then the results are presented and interpreted as shown in Figure 2.3.

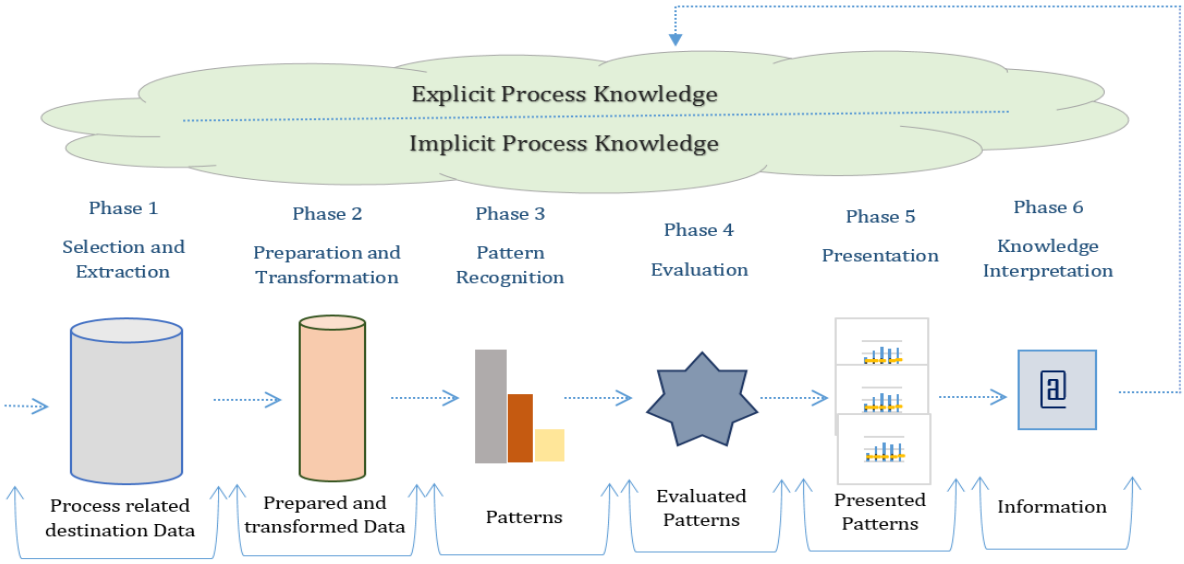


Figure 2.3 Process Mining and Analysis Framework (Holzhüter, et al., 2013)

As gathered in Figure 2.3 such type of analysis, allied to the *process mining* techniques, improves explicit process knowledge e.g. the learning processes that are complemented by some implicit knowledge which may also be discovered inadvertently (Grob, et al., 2008). In other words, process mining could be adopted to analyse and improve learning processes and models, or even more, used to recommend future learning patterns and/or behaviours.

In summary, reference (Holzhüter, et al., 2013) introduces the notion of process mining technique in e-learning settings to highlight the need of the approach in offering the capability of improving information values of those systems. The authors opines that process mining in

combination with learning concepts could be a promising tool for learning process automation and modelling. Their arguments are concerned with how learning processes can be improved through the use of process modelling and rule-based controls, as well as how process models can be generated in details taking into account the concept of learning styles or patterns. The outcome of their research (Holzhüter, et al., 2013) interestingly reveals that the implementation of rule-based controlled methods (e.g. workflows) into systems that support learning process or management still rests an important field of further investigation.

For that reason and to address such gap in literature - this thesis has introduced the SPMaAF framework and its main application in real-time setting which is perceived as a semantic-based PM approach (i.e extended or semantically enriched version of the Figure 2.3) directed towards the discovery and enhancement of the resulting process models with its custom on the learning process domain in order to show the influence and usefulness of the approach.

2.5 Business Intelligence (BI)

Business Intelligence (BI) is a broad term that combines tools or methodology which aims at providing actionable information that can be used to support decision making (Van der Aalst, 2011). Most of the time, BI are used in place of the overlapping term such as the Business Process Intelligence (BPI) (Ingvaldsen, et al., 2005; van Dongen, et al., 2016). Many other terms has also been used under the umbrella of BI such as the Business Process Management (BPM) (Santoro, et al., 2016), Business Activity Monitoring (BAM) (Van der Aalst, 2016), Corporate Performance Management (CPM) (Van der Aalst, 2016) etc. According to (Van der Aalst, 2016) these terms has emerged as a result of many vendors who offer BI software merchandises and their attempts to verbally distinguish themselves from other competitors as well as to show additional functionalities their product is capable of offering.

In most cases, the main functionalities offered by those BI products includes but are not limited to the following five purposes as observed in (Van der Aalst, 2011; Van der Aalst, 2016):

- ✓ *Extract, Transform and Load* (ETL)
- ✓ *Ad-hoc Querying*
- ✓ *Reporting*
- ✓ *Interactive Dashboards*
- ✓ *Alert Generation*

Furthermost, some example of the state of the art BI systems (de Leoni, et al., 2012; Gatner, 2010; Van der Aalst, 2011) are as follows: IBM Cognos Analytics[1], Oracle Business Intelligence[2], SAP Business Intelligence[3], WebFOCUS[4], SQL Server BI[5], MicroStrategy[6], Necto Panorama Software[7], Qlik[8], SAS BI & Analytics[9], TIBCO Spotfire[10], TIBCO Jaspersoft[11], Pentaho Business analytics[12], and Tableau[13] the recently trend in business data visualization etc.

---

1. https://www.ibm.com/analytics/us/en/technology/products/cognos-analytics/
2. https://www.oracle.com/solutions/business-analytics/business-intelligence/index.html
3. https://www.sap.com/uk/products/analytics/business-intelligence-bi.html
4. http://www.informationbuilders.co.uk/products/intelligence
5. https://www.microsoft.com/en-us/sql-server/sql-business-intelligence
6. https://www.microstrategy.com/us/products/capabilities/advanced-analytics
7. https://www.panorama.com/
8. http://www.qlik.com/us/products
9. https://www.sas.com/en_gb/software/business-intelligence.html
10. https://spotfire.tibco.com/
11. https://www.jaspersoft.com/business-intelligence-solutions
12. http://www.pentaho.com/product/business-visualization-analytics
13. https://www.tableau.com/

Indeed, process mining (Van der Aalst, 2011) has been professed and proves to be a new collection of the BI techniques, and has been closely ranked as a worthwhile and powerful technique within the wider context of BPM, and industrial concepts of BAM.

BAM: refers to the real-time monitoring of business processes (Van der Aalst, 2011). The method allows and supports many organisations in extracting business performance metrics and relates those measures to the business operations. BAM tools as observed by the authors in (Van der Aalst, 2016; Ingvaldsen, 2011) are different from the process mining term because they assume a pre-defined or causal business process models. Every now and then, the resulting process models are out-dated and/or unfitting. Besides the BAM tools pursues to accomplish the business communication problems rather than the entire details and/or aspects of a real-life or actual business process (Adriansyah, et al., 2011).

BPI: is another term that is used to refer to BI. According to the focus of one of the main discussion in BPI technologies and applications (van Dongen, et al., 2016), BPI is an area that spans different aspects of process mining - from process discovery to conformance checking, model enhancement to predictive analytics and many other techniques which are being

investigated, and altogether, are gaining attention within the research industry. In other words, the BPI notion refers to application of many different measures and analysis methods within the area of BPM (Santoro, et al., 2016).

Moreover, the BPI is characterized by a common goal in relation to the process mining field. This means that algorithms or tools that supports the BPI also operates on event logs (i.e captured datasets) in relation to the sequence of activities or traces as performed within the so-called processes (e.g. business process). In practice, BPI embodies methods used for handling process executions and values by proposing several features such as optimization, monitoring and control to prediction and analysis etc.

Nonetheless, references (Ingvaldsen, 2011; Erdmann, et al., 2000; Van der Aalst, 2016) observes that considering the analytical capabilities of current systems that supports BPI in practice, they are not always very intelligent and suffers from some kind of functionality limitations. Thus, such restrictions can be viewed from different perspectives as noted in (Ingvaldsen, 2011) as follows:

- *Performance perspectives*: where logs are not structured enough for data analysis. Process logs are designed to ease and not delay the job of the process engines.
- *Data quality perspectives*: which includes - noise, inconsistencies or missing data, and special codes that are not intuitive and easy to solve.
- *Semantics perspectives*: many times, even though the datasets are being extracted and modelled with acceptable performance to accurately reflect the execution of the processes, they may still be useless or unusable for many process analysis purpose, because they lack the abstraction level required from a real world perspective.

Indeed, researchers, software vendors, and many large scale organizations has been pursuing industriously over the years in developing methodologies and tools which targets at overcoming such limitations with the BPI.

A typical example is the annual BPI workshop (van Dongen, et al., 2016) which primarily strives to achieve but are not limited to the following goals: *Process discovery, Intelligent Process analysis, Prediction, Handling Decisions and Exceptions, Optimization of Static processes, Optimization of Dynamic processes* etc.

BPM: another significant aspect of process mining is that it complements existing approaches to BPM. Interestingly (Van der Aalst, 2016; Van der Aalst, 2011) refers to BPM as the "*discipline that combines knowledge from information technology and knowledge from management sciences, and applies this to operational business process*" (Van der Aalst, 2004; Weske, 2007). Moreover, BPM heavily rely on process modelling techniques, and has been perceived as an extension of WFM systems (Van der Aalst, 2011; Van der Aalst, et al., 2004).

In any case, both the BPM and classical WFM systems (Van der Aalst, 2011) provides facts or key information on how activities are being performed within a process, and as such, are being derived from the captured datasets, often referred to as *event logs* (Gunther, et al., 2008; Van der Aalst, 2011).

Other Overlapping Terms: according to (Van der Aalst, 2011) also related to BI are management tools and techniques which includes: Six Sigma (mainly used for improving operational performance), Total Quality Management (TQM) and Continuous Process Improvement (CPI). These tools commonly have the characteristics that a process is put under scrutiny in order to check if further enhancements can be done or is possible.

Remarkably, (Van der Aalst, 2011) opines that BI systems and the many overlapping terms that are used to refer to the notion are not actually *intelligent* as they appear to be, and tends not to entirely encompass all the capabilities of the process mining techniques. According to the author there is a problem with existing BI tools because they focus more on fancy looking dashboards or reports rather than carrying out an in-depth analysis of the captured datasets. To this end, reference (Van der Aalst, 2016; Van der Aalst, 2011) mentions that all of such BI systems are too *data centric* and are solely not aware of the process the captured datasets are being used in. Even though, the author in (Van der Aalst, 2011) states that a typical BI system provides data mining capabilities, such as Clustering, Regression, Classification, Association Rule Learning etc.

Perhaps, such capabilities are not all, but one part of the techniques used for process mining to perform its tasks (e.g. the process mapping, trace classifications etc.).

In short, nowadays process *aware* and truly *intelligent* BI systems are potentially credible thanks to the advances within the *process mining* field (deMedeiros, et al., 2008; Van der Aalst, 2011; de Medeiros & Van der Aalst, 2009; Okoye, et al., 2017) which refers to techniques that are capable of extracting the real knowledge (i.e. semantics) behind the labels in events logs and the derived models, thus, *semantic process mining*.

2.6 Semantic Web Search Technology

Semantic web search technology refers to tools that trails to combine the notion of *information extraction* (IE) (Calvanese, et al., 2016) and *information retrieval* (IR) (Manning, et al., 2008) to find meaningful information or files from large collections of databases, and then present the output/results to users (search initiator) based on some pre-specified information need. Therefore, whilst IR systems focuses on finding useful materials (e.g. documents) from a large collection of unstructured data (e.g. the internet), the IE pursues to present the specified informations in form or state in which the users are interested in by providing the output in a structured format. Thus, this is the mechanism upon which the semantic web search methods such as the semantic-based approach proposed in this thesis are built. In fact, *semantic web search* simply means finding a set of text or information that are relevant to the user query (Ingvaldsen, 2011).

Interestingly, and in context of this thesis (Cunningham, 2005) opines that semantic web technologies targets to add machine tractable and/or repurposeable layer of annotations that are relative to *ontologies.* The method is used to match or complement the overwhelming (omnipresence) web of natural language hyper-text (Fensel, et al., 2002; Bechhofer, et al., 2004) by creating semantically annotated terms, and then linking the resulting pages to ontologies. Moreover, such process turns out to become automatic or semi-automatic in nature due to the formal design, development, and inter-relation of the ontologies.

According to (Cunningham, 2005) a typical example of such tools that supports semantic-based web search is the Knowledge and Information Management system (KIM) (Popov, et al., 2004) which offers IE-based facilities for metadata creation, storage, and semantically enriched web browsing or search. Equally, many other tools exist in literature e.g. SemTag system (Dill, et al., 2003) and Magpie (Domingue, et al., 2004) an add-on for the browser which relies on the fact that it makes use of ontologies to provide precise or tailored perspectives of the web pages which the user might be interested in (or wishes to browse) etc.

Interestingly, OWL (W3C, 2004) has emerged as the standard format for defining the semantic web ontologies, and has since in recent years, widely been accepted and used towards advanced structuring of information and knowledge engineering for the purpose of enriching the datasets and depiction of inference rules. For example, as utilized in chapter 4 and 5 of this thesis to support the process descriptions (assertions) and semantic reasoning of the derived process models at a more conceptual level. Moreover, as sets of annotated terms

and relations, the resulting *ontologies* supports information extraction particularly likened to the *Ontology-based Information Extraction* (OBIE) systems (Wimalasuriya & Dou, 2010).

Over the next sections, the works looks at the Ontology Based Information Extraction systems (OBIE) and how they could be used to support meaningful information extractions and management of processes in different context, especially within the process mining field.

2.7 Ontology-Based Information Extraction (OBIE) Systems

Ontology Based Information Extraction systems (OBIE) refer to tools and approaches that pursues to identify and extract information in form of texts or concepts, and describes the relations or properties which are relatedly expressed in an ontology for any particular process domain (Yankova, et al., 2008). The OBIE notion spans and is inspired by the *Information Extraction* (IE) terminology. According to (Cunningham, 2005) a typical IE systems takes as input - texts and even at times speech, to produce fixed format explicit data as outputs. Clearly, such method of information extraction means that IE systems only presents relevant (specific) information or knowledge in form in which the users are interested in.

Apparently, this feature is where OBIE systems draws its incentive due to the fact that *ontology* is one of such tools that has the capacity of providing information in a structured format. For example, the automatic population of ontologies in OBIE applications is capable of identifying instances within a text document or process models that belongs to a particular class, and trails to add those learned instances within their correctly inferred locations.

(Yankova, et al., 2008) observes that such method for information aggregation has proved to be advantageous by increasing the confidence of extracted information and storage of updated information within the process knowledge bases. For instance, if a person age is added to his/her description in an OBIE system, it is expected that the age restriction will be added to a new identity criterion, and not necessarily changing the entire function of the system.

However, (Yankova, et al., 2008) reveals that one fundamental problem to be addressed when providing a structure for distribution of conceptual knowledge such as the OBIE system, is the issue of *identification* and *integration* of the entities (instances) which are extracted from different data sources. Apparently, the process should aim at identifying newly extracted facts (e.g. from texts, models, knowledge-bases) and linking them to their previous references. To this end, (Cunningham, 2005) states that OBIE poses two main challenges which are directed towards:

(i) identification of concepts from the ontologies, and
(ii) automatic population of ontologies with instances in the texts or databases.

Moreover (Cunningham, 2005) observes that if the ontology in question is already populated with instances, the task of an OBIE system may perhaps be simply to identify and integrate those instances from the ontology in the text or data sources. Indeed, such methodology could be more useful as opposed to the traditional IE systems because they makes use of an ontology rather than a flat gazetteer (Cunningham, 2005).

In principle, (Cunningham, 2005) notes that for such systems which are *rule-based*, the procedures are pretty direct (straightforward). But for the systems which are *learning-based*, it appears to be somewhat challenging because a *training dataset* is most often necessary. In turn, the collection of the training dataset is likely to be a constrain and/or bottleneck. Therefore, to address such problem - new training datasets has to be manually or semi-automatically created, which appears to be time-consuming and are burdensome task. Although, new approaches and systems are currently being developed with intent to help support such metadata creation. Besides, a number of OBIE supported systems has been proposed and used in different settings in current literature (De Giacomo, et al., 2018; d'Amato, et al., 2008; deMedeiros, et al., 2008; Okoye, et al., 2016; Yankova, et al., 2008).

In summary, unlike *traditional* information extraction (IE) where the extracted facts or knowledge are only classified as appropriate for pre-defined data types, a typical OBIE system must seek to discover structures or resolutions which targets at generating reference links between objects (process instances) that are inherent within the knowledge-base of the systems as well as their mentions within the contextual domain (Cunningham, 2005). In fact, any *ontology-based* systems should not only contain the representations of the specific domains, but should also provide information about the identified instances (entities) as well as their properties. Thus, an ontological knowledge base system must contain a set of well-defined entities (e.g. the process instances and class) with their full semantic descriptions.

2.7.1 OBIE in context of Process Mining

In another application domain, reference (Calvanese, et al., 2016) relates OBIE within the context of process mining by highlighting the extreme challenges encountered when extracting event data, and then reveals the necessity for appropriate methods that are deemed beneficial towards extracting events log from relational databases. According to the authors

(Calvanese, et al., 2016), *information extraction* processes spans across several levels of abstractions - from the high-level (i.e. the domain-independent notions which are characterized at the conceptual level by the so-called domain ontology), to coming down to the concrete level at which sets of data are effectively stored.

(Calvanese, et al., 2016) notes that several tools such as the XESame (Verbeek, et al., 2011), ProMimport (Verbeek, 2014; Gu¨nther & Van der Aalst, 2006), and ProM (Verbeek, et al., 2011) which all supports event log extraction, and commercial tools such as Disco (Rozinat & Gunther, 2012) and many other overlapping tools such as MinIt, Celonis etc. that makes it easier to transform Excel or CSV files into an eXtensible Event Streams – XES (IEEE 1849-2016, 2016) or Mining eXtensible Markup Language – MXML (deMedeiros, et al., 2008) log have already been developed. However, the authors observes that none of such tools or platforms, in reality, considers the domain ontologies in the loop. In consequence, the process of extracting useful informations are a lot of the time *ad-hoc* because in such settings, the data may be duplicated for dissimilar interpretations, and the semantics of the available datasets perhaps cannot be traced back in most cases.

Moreover, (Calvanese, et al., 2016) notes that some works have similarly been done on semantically annotated events log (deMedeiros, et al., 2008; Allahyari, et al., 2014; Erdmann, et al., 2000; Calvanese, et al., 2017; Ingvaldsen, 2011) which focus on exploiting such ontological information during data analysis, but yet, do not put profusely in consideration the process of extracting the event logs. Even though, to overcome such challenges, (Calvanese, et al., 2016) argues that it could be theoretically applied only if realistic datasets (i.e. event logs) that follows the accepted standards (e.g. XES) are available, and in view of that, propose a novel framework that supports domain experts in the extraction of XES event log information from legacy relational databases (Calvanese, et al., 2016).

More so, to demonstrate the capability of the ontology-based framework in context of the process mining, the authors (Calvanese, et al., 2016) resort to a well-established Ontology-Based Data Access (OBDA) model which allows one to link the raw data to the underlying domain ontologies (i.e. hierarchical datasets or taxonomy) and overcome the impedance mismatch. In so doing, the process analysts can focus more on the ontological levels only, while the associations within the underlying knowledge-base/datasets are managed automatically by the OBDA system (De Giacomo, et al., 2018; Poggi, et al., 2008). In general, the ontology-based approaches provides basis for the development of process mining tools

and algorithms that are capable of extracting conceptual informations - either by explicitly materializing it or by retrieving the informations on demand.

2.7.2 OBIE in context of Knowledge Extraction for BI

Business Intelligence (BI) as discussed earlier in section 2.4 of this thesis - are gateways of information systems which assist business analysts with the tasks of discovering, gathering, aggregating, and analysing of information or searching for documents. Often, it is up to the user of such system to dig into the large amounts of the readily available information to find relevant facts that supports the so-called decision making processes e.g. credit ratings, measuring probability of the business success, discovering appropriate business partners, or getting up-to-date facts about companies, places, people etc. (Yankova, et al., 2008).

The authors in (Yankova, et al., 2008) explains the application of OBIE systems within the Business Intelligence (BI) context through their project named MUSING project (Yankova, et al., 2008). The project focuses on integrating Human Language technologies with Semantics applications in order to address the problem of identifying and integration of the process elements that are being extracted from the various data sources which are every now and then related to OBIE supported applications.

Indeed, the works in (Yankova, et al., 2008; Calvanese, et al., 2017; De Giacomo, et al., 2018) shows that the answer to such problem relies on applying process modelling and natural language processing (NLP) (Maynard, et al., 2007) approaches to the supposed *unstructured-data sources*. Apparently, this should aim at allowing the transformation of the data extracts into a well-structured representation that fits the said higher level (conceptual) of analysis. For example, the MUSING project (Yankova, et al., 2008) has been developed with key focus on identification of newly extracted business facts, e.g. from text or models, and pursues to link them to their prior references through formal structures (i.e. population of ontologies). The authors (Yankova, et al., 2008) has developed the project using the GATE platform (Cunningham, et al., 1995) a general architecture for text engineering developer tool that offers a wide range of applications that are useful towards the development of OBIE systems, especially with its ability to support ontology-based projects. The main component and lesson here (i.e. the key element of the system) is the *annotations* created by the method through encoding of ontological informations (*mention* annotations) which makes references to targeted ontologies, as well as, the ontological concepts referenced by the strings of texts or labels such as the one introduced in this thesis.

2.7.3 Measuring Performance and Flexibility in OBIE Systems

To recapitulate the importance and relevance of OBIE applications, the process of its classification tasks, and the underlying logics: reference (Maynard, et al., 2008) discuss methods for measuring the performance of OBIE systems. The authors looks at why the traditional *precision* and *recall* evaluation metrics used for systematic information extraction methods are somewhat insufficient when ontologies are involved. In consequence, they propose the Balanced Distance Metric (BDM) (Maynard, et al., 2006) a new metrics which measures flexibility and takes similarities between ontology-based systems into account.

Specifically, the experimentations in (Maynard, et al., 2008) based their arguments on the reason why the traditionally discovered entities (classes) does not incorporate one another? While in an ontology based settings, there exists *subclasses* or *superclasses* types to reason or integrate. Thus, discrepancies amongst correct and incorrect (i.e. true or false) classes is not clear. In turn, the authors (Maynard, et al., 2008) observes that with traditional or standard IE systems, an instance (i.e entity) that is recognised as a *person* may either be true or false when measured using *precision* method. However, when measured by *recall*, the instances which should have been recognised as a *person* are either identified or not at all. Moreover, the dissimilarities are fuzzier when making ontological classifications after all. Nonetheless, (Maynard, et al., 2008) argue that such method of evaluation can at times be conventionally realised by assigning a half-weight to *things* (e.g. the entities) considered to be partially correct, yet, are still not enough to provide proper differentiation between the levels of accuracy or precision. Even though, credit should be given for partial correctness.

Surely, the BDM follows some of the guidelines proposed by (King, 2003). Henceforth, an ontology-based application performance metrics should:

- reach its highest value for perfect quality
- reach its lowest value for worst possible quality
- be monotonic
- be clear and intuitive
- correlate well with human judgement
- be reliable and exhibit as little variance as possible
- be cheap to set up and apply
- be automatic

Indeed, the preliminary observations in (Maynard, et al., 2008) shows that two-fold (binary) decisions are not appropriate for evaluation of ontology-based systems, especially in settings where objects or class hierarchy (taxonomy) are being considered. According to the results, both the Balanced Distance Metrics (BDM) and Learning Accuracy (LA) metrics are more useful than a distance-based or flat metric when evaluating information extraction based on hierarchical rather than a flat structure (Maynard, et al., 2008).

In summary, the most relevant outcome of the evaluation in (Maynard, et al., 2008) with regards to the context of this thesis - is the usefulness of such performance measures in *population of ontologies*. In essence, the evaluation metrics were specifically centred on structures of an ontology to analyse similarities between the concepts defined within the ontologies. Moreover, since based on *description logic*: the ontologies are able to allow semantic interpretations, and can also measure concepts similarities by using the underlying process descriptions and languages. According to (Maynard, et al., 2008) this type of semantic similarity would make more sense than the structure-based measures particularly for complex systems or ontologies that contains different kinds of relations. Especially, to help provide more flexibility and support for task specific processes and systems which are believed to be *aware* of the processes they trail to support as discussed in the following section.

2.8 Process-Aware Information Systems (PAIS)

Process-Aware Information Systems (PAIS) comprises of systems that provide more flexibility and support for task specific processes (Dumas, et al., 2005; de Leoni, et al., 2008). The traditional WFM systems are typical example of PAIS. Many other process control and management systems can be classed under the umbrella of PAIS. Systems such as SAP (Oracle), Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), WebSphere (High-end middleware), Rule-based systems, Call Centre softwares etc.

Even though PAIS systems tends not to necessarily control the process via some generic *workflow* engine, they share a common attribute which is that the information systems are entirely *aware* of each and every process they trail to support, and also, for the fact that there exists an explicit process view or interpretations (Van der Aalst, 2011). According to (Van der Aalst, 2011) many database systems or programs may well be utilized for executing activity sequence (i.e. steps) within a given process (e.g. business process). However, those systems are most often not *aware* of the methods/processes they are being utilised for. For

that reason, they appear not to be dynamically involved in the transposition or management of the processes they are involved in.

Therefore, the more flexibility (generalization) a PAIS allows for, the greater diversity of behaviour that can be derived from the supported processes. In other words, only in situations where the process in question shows great level of flexibility, would the resulting process models offer the best values when compared to other methods used to reveal or support task specific processes such as the BAM or BPM.

Totally, the process mining technique embodies PAIS, because process mining aims to mine and analyse event logs at *process-levels,* and at the same time, are entirely aware of the *facts* and to great extent details of how the various activities has been performed. This form of analysis is interrelated to the notion of the proposed SPMaAF framework in this thesis, in the sense that the approach makes use of semantic annotations to add meanings to the discovered and/or existing process models, and therefore enables automated inference (e.g. through process querying) of knowledge from the domain processes with the goal of bringing the process-related information to a much more level of human (i.e real world) understanding. Thus, the method is classed to support *machine-understandable* system rather than just *machine-readable* system.

2.9 Process Querying

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 (Polyvyanyy, et al., 2017; Polyvyanyy & et al, 2016). According to (Polyvyanyy, et al., 2017) the process querying technique concerns automatic methods for handling (e.g. filtering or manipulating) repositories of models of observed and unseen processes as well as their relationships, with the intension of transforming the process-related information into decision making capabilities.

In practice, reference (Polyvyanyy & et al, 2016) notes that 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. (Polyvyanyy & et al, 2016) 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 (Polyvyanyy & et al, 2016) one reason for such growing interest, is that ontologies permits the adding of semantics to

discovered or existing process models which enables the automated discovery or 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 (Polyvyanyy, et al., 2017) 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. According to the authors, process querying methods need to address those gaps. For instance, organizations often fail to convert the high volume of data recorded in their various information system into strategic and tactical intelligence. This is due to the lack of dedicated technologies that are designed to effectively manage the information on the processes encoded within the envisioned process models or data records, in order to better support strategic decision-making and to provide the next generation of Business Intelligence. Interestingly, the proposed framework listed in (Polyvyanyy, et al., 2017) 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 thesis, the research focus is particularly on the *Process Querying with Rich Annotations* (Polyvyanyy & et al, 2016) which studies the use of rich ontology annotations of process models for the purpose of process querying. Besides, (Montani, et al., 2017) notes that a trace abstraction technique for any semantic-based process mining and model analysis should present methods or design frameworks which are able to convert actions found within the discovered traces into higher level concepts based on the domain knowledge, thus, the term *conceptualization*.

2.10 Summary of Related Works

As conversed in this chapter and in the following Table 2.1, the research reviewed and discusses the theoretical backgrounds that needs to be laid in order to allow the readers have an inclusive view and understanding of the work that have been done in this thesis.

Table 1.1 Systematic review of the related works and findings in relation to the research investigations, aim and objectives

Work/Paper	Field relevant to Thesis	Field	Findings relevant to Research questions, aim and objectives	If yes, then what Design approach was used?	Tools
(Bogarín, et al., 2018)	Yes	Educational Process Mining, Intentional mining, Sequential pattern mining and Graph mining	Yes, the authors observes that the application of process mining to raw educational data takes into account the end to end processes rather than local patterns, as opposed to the Educational Data Mining. They suggest that Semantic concepts can be layered on top of existing learner information assets to provide a more conceptual analysis of the processes in reality	The paper reviews existing techniques used for EPM and elaborates on some of the potential of these technologies. It describes some of the relevant, related areas and highlights the components of an EPM framework. It also describes the data, tools, techniques and models used in EPM	Process Discovery and Conformance Checking Techniques, Dotted Chart and Social Network Analysis Techniques, Massive Open Online Courses (MOOCs), Learning Management System (LMS), Hypermedia learning environments, Curriculum Mining, Computer-Supported Collaborative Learning, Software Repositories etc.
(Cairns, et al., 2014)	Yes	EPM, Process Discovery and Information Retrieval, Semantics particularly ontologies, Learning Process Automation	Yes, the authors observe one of the problem with existing process mining techniques is that they rely exclusively on the syntax of labels in the databases or events log, and are very sensitive to data heterogeneity, label name	The paper propose a semi-automatic procedure used to associate semantics to training labels by extracting process models annotated with semantic information in order to provide a more accurate mining and compact analysis of the	Ontology Abstract Filter plug-in in ProM, Semantically Annotated Mining eXtensible Markup Language (SA-MXML), Heuristic Miner algorithm

			variation and frequent changes. As a result, majority of the process models are discovered without some kind of hierarchy or structuring.	processes at different levels of abstraction	
(deMedeiros, et al., 2008)	Yes	Semantic Process Mining, Events Logs Annotation, Ontologies	Yes, the authors opines that the three core building blocks i.e Annotation of events logs, Ontologies and Semantic reasoning; if adequately utilized, could cater for a much more robust and accurate process mining and analysis technique as opposed to the traditional means of process mining.	The paper propose the Semantic LTL Checker Algorithm by extending the existing LTL Checker conformance and analysis plug-in in ProM in order to exploit semantic annotations. Their approach applies concepts in an *ontology* as input to parameters of a Linear Temporal Logic (LTL) formulae to formulate and answer questions about process elements (instances) by making use of the WSML2Reasoner to infer all the necessary associations.	Conformance analysis plug-in *LTL Checker* in ProM, Annotation, Semantic Reasoning, and LTL formulas and template.
(Han, et al., 2011; d'Amato, et al., 2008; Elhebir & Abraham, 2015)	Yes	Classification, Ontology, Data Classes and Concepts.	Yes, the works all observe that Classification is one of the most common data mining (DM) technique that aims at finding models or functions that describes and/or distinguishes data classes. Explicitly, the works shows that the problems of modelling	Combination of multiple classifiers (i.e hybrid algorithms) and Ontologies which are incorporated as consistency constraints into multiple related classification tasks, clustering and ranking of individuals. And also, pattern discovery algorithms that uses	Data Mining techniques, Semantic labelling, Process Description logics and various Classifiers.

			domain processes can be resolved by transforming ontology population problem to a *classification* one where for each entity within the ontologies, the concepts (i.e classes) to which the entities belongs to have to be determined, hence, classified.	statistical and machine-learning techniques to build models that predicts behaviour of captured data, and	
(Baati, et al., 2017; Baati, et al., 2016; Zadeh, 1999; Peña-Ayala & Sossa, 2013)	Yes	Fuzzy Logics, Fuzzy Sets, Fusion Theory, Fuzzy Mining and Reasoning	Yes, the works observes that information imperfections treated as possibility theory may represent information or data uncertainty due to variability of observations, the uncertainty due to poor information, ambiguity, or information imprecision etc., and in turn, the quality of decision may be seriously altered since the final classification tasks is likely to be inaccurate.	The works makes use of the Fusion theory (i.e combination of two or more algorithms, classifiers etc.) based on fuzzy sets theory that are devoted to represent and combine imperfect information in a qualitative or yet quantitative manner. Most of the approach organizes *ontologies* which are used in identification and definition of the meanings of concepts, casual relations, fuzzy rules bases, universe of discourse etc.	Extended Bayesian classifiers, Generalized Minimum-based (G-Min) algorithm, OWL Schema and declaration sentences such as Classes, Datatype Properties, Functional Properties, and Individuals etc.
(Peña-Ayala, 2013)	Yes	Intelligent and adaptive educational learning systems (IAELS), Artificial Intelligence (AI), User Modelling,	Yes, the author looks at collection of methods and parameters that could be raised to be paramount to the idea of using the process mining in combination of	The work introduces various approaches for the user attribute which are semantically described in form of concepts within ontologies particularly by enabling	Computer-based and Business Intelligent (BI) tools, Workflow management system (WFM),

		Content Representation, and Case studies Application	semantic modelling techniques to manage perspectives of the learning process domains.	classification of users based on qualitative observation of the ascertained properties. In other words, Concepts and Entities Relationship Prediction	Learner Models, OWL and the property descriptions.
(Van der Aalst, 2016; de Leoni, et al., 2012; van Dongen, et al., 2016; Ingvaldsen, 2011)	Yes	Process Mining, BI, Business Process Management (BPM), Business Activity Monitoring (BAM), Corporate Performance Management (CPM), Process-Aware Information Systems (PAIS) etc.	Yes, the works looks at ways towards the Extraction, Transformation and interpretation of events logs about any given process domain. However, the works observe that all of such BI systems are too data centric and are solely not aware of the process the captured datasets are being used in	The works includes various approaches and methods that combines tools or methodology which aims at providing actionable information that can be used to support decision making particularly in terms of process mining.	Process Mining e.g. PROM, Disco etc. Data mining tools e.g. Clustering, Regression, Classification, Association Rule Learning, Predictive Analytics and WFM systems, BI tools e.g. SAP, WebFOCUS, SQL Server, TIBCO, Pentaho, Tableau etc.
(Calvanese, et al., 2017; Yankova, et al., 2008; Cunningham, 2005; Calvanese, et al., 2016; Maynard, et al., 2008)	Yes,	Semantic Web Search Technologies, Ontology-based Information Extraction (OBIE), Information Extraction (IE), Information Retrieval (IR), Process Mining, and Database Management.	Yes, the works seeks ways on structuring of information and knowledge engineering for the purpose of enriching the datasets and depiction of inference rules. Particularly in terms of identification of concepts from the ontologies, and automatic population of ontologies with	The authors developed tools and methods that supports semantic-based process analysis by offering either OBIE, IR or IE based facilities for metadata creation, storage, and semantically enriched queries and/or search.	XESame, ProMimport, eXtensible Event Streams (XES), Ontology-Based Data Access (OBDA) model, OWL, Knowledge and Information Management system (KIM), SemTag system, Magpie etc.

			instances within the knowledge-bases.		
(Polyvyanyy, et al., 2017; Montani, et al., 2017; Polyvyanyy & et al, 2016)	Yes	Process Querying, Trace Abstraction and Classifications, Ontologies for process management, BPM, BI, Semantic Modelling and Annotations.	Yes, the authors studies techniques which concerns automatic methods for handling (e.g. filtering or manipulating) repositories of models of observed and unseen processes as well as their relationships, with the intension of transforming process-related information into decision making capabilities.	The authors propose 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.	Ontology annotations, Process Models etc.
(Okoye, et al., 2017; Okoye, et al., 2016)	Yes	Process Mining, Semantic Modelling, Process Models and Events Log Annotation, Ontologies, Semantic Reasoning, Fuzzy Mining BPMN notation etc.	Yes, the authors observes that most of the existing process mining techniques appears to be vague or limited when confronted with unstructured data because they depend on the tags (i.e labels) within the events logs to analyse the process rather than concepts, and therefore, seeks ways on how best to use semantic-based approach to manage the perspectives of process mining.	The authors introduces a framework and sets of semantically motivated algorithms for construction of semantic-based process mining technique that exhibits a high level of intelligence and conceptual reasoning through semantic – labelling (annotation), representation (ontology) and reasoning (reasoner).	Process Mining tools such as Disco and PROM, Process Modelling tools such Bizagi Modeller, Fuzzy and BPMN Models, Ontologies and Process description Languages such as OWL, Semantic Web Rule Language (SWRL), Description Logics (DL), Reasoners e.g. Pellet, OWL API etc.

In summary, the work discussed and analysed in this chapter of the thesis (particularly the systematic review in Table 2.1) the main motives that have led to the emergence and use of the process mining technique, its trailed theories, and influence especially within the educational domain. In addition, since process mining techniques builds on computational intelligence and data mining techniques, which has led to its significant influence on how the process owners and the analysts perceive and analyse the readily available large volumes of data captured from their various IT systems, respectively. The research deemed it necessary to look at the other overlapping terms within the process mining field and main areas of its application that are pertinent for the proposed approach in this thesis. To this end, the work investigates the practical use of the techniques in relation to information analysis and semantic modelling. The study reviews approaches that are entirely aware of the several process which they trail to support and methods that are used to extract meaningful patterns from the event logs captured about those processes, and ways of transforming and analysing the datasets in order to provide real knowledge (semantics) and understanding of the processes in reality. In other words, the thesis looks at means to semantically improve the information values of the readily available process logs as well as their analysis at a more conceptual levels.

Indeed, most organisation's processes are complex, because the amount of data available today has outgrown both the human expectations and processing capabilities of the various IT systems. Yet, the opportunities, investigations, and good news from the available literature reviews and systematic analysis remains that there are solutions, as researchers are working resolutely to meet this expectations. Certainly, if the process analysts, software vendors and IT experts (whose task are to provide methods and tools to manage such increasingly business processes and organizational data) can understand these progressively more and unprecedented large volume of data with *advance level of intelligence*, then they can start to realise the power of *process mining* especially when integrated with the *semantic modelling* techniques. Clearly, such methods could be employed for diagnosis and subsequently utilized to improve those complex processes in reality, only if the analysts should take the additional step of providing the real knowledge (*semantics*) that describes the said processes. Besides:

"Process Mining" *represents* "Data Science in Action"
"Data Science" *represents* "Augmented Intelligence" (AI)
"AI" *represents* "Computing + Human Input"
"Human Input" *represent* "the Process Analyst, Software vendors & IT experts"
Agreeing to (Van der Aalst, 2016) "Start Today"

Chapter 3. State of the Art Components, Process Mining and Analysis Methods

This chapter looks at the state of the art components, tools and methods as it relates to the research investigations and main areas of interest. It describes the key components of the process mining and semantic modelling methods, and the tools which enables the practice and application of the techniques. In other words, the chapter explains the main mechanisms applied in this thesis: ranging from event logs to available process mining techniques, tools and algorithms used to discover the process models and to help improve their interpretations and analysis. Also, given the fact that process mining techniques are not possible in absent of appropriate events data log, the chapter discusses and looks in details - the need for event data, the informations expected to be existing in the data logs, and data quality challenges that may be encountered in reality as well as how those challenges can be addressed. In addition, the chapter also looks at current tools that supports the semantic-based process mining approach – ranging from the annotation of event logs to ontological representation of the resulting models and semantic reasoning, and then illustrates how it can be applied to perform effective analysis of the event logs and process models at a more abstraction level. Finally the chapter summarizes the presented state of the art components and approaches.

3.1 Event Logs

Process mining algorithms use event logs to learn and reason about processes by coupling in a technical manner: *event history data* and *process models* (Van der Aalst, 2011). Indeed, data logged in IT systems can be utilized towards provision of a better understanding or insights about real-time processes by improving the quality of the discovered models, analysis of the individual process elements, or detecting deviations. In fact, process mining combines techniques from computational intelligence and data mining to process modelling and analysis, as well as several other disciplines to analyze the captured datasets.

Many approaches that incorporates such use of data mining techniques to interpret datasets has been proposed in existing literatures. On one hand, references (Dou, et al., 2015; De Leoni & Van der Aalst, 2013; Han, et al., 2011) refers to data mining as the techniques that are used to analyse recorded datasets in order to find unpredicted relations, and then trails to summarise or interpret the data in a more novel way that are both meaningful, understandable, as well as beneficial to the data owners.

Likewise, process mining allows for the same practice as data mining, but aims to analyse the recorded event data at *process-levels* (Van der Aalst, 2016). Such an advanced analysis at process-levels helps to address the problem of determining unswerving connections amid the low level events log about the processes in view, and the discovered process models in reality. This means that the process mining techniques are not limited to automatic discovery and interpretation of patterns within processes, but also are built on *data mining* and *process modelling* techniques because many of the existing data mining approaches appears to be overly data-centred in providing an inclusive or full understanding of the end to end processes in execution (e.g. from a business process perspective).

In essence, process mining assume that a typical dataset (i.e. the event logs) consist of at least information about a *single process* and every event within the data logs has to refer to a single process *trace*, by specifying the process as group of *activities* such that the *life-cycle* (or sequence of activities) of a single process trace is established. In turn, the *event log* serves as the first step (i.e starting point) for process mining. Moreover, the event logs are a multiset of *traces* where each trace describes the life-cycle of a particular case (i.e. a *sequence of activities*) in terms of how they have been executed within the said process. Thus:

- ✓ The "*Case ID*" and "*Activity*" represents the bare minimum to perform a process mining task (Van der Aalst, 2016).

Other additional information may be required for ample implementation of process mining:

- ✓ *Event ID* - for ordering of information to discover casual dependencies in process models.
- ✓ *Timestamp* – useful when analysing performance related properties (for instance, the waiting time between two activities).
- ✓ *Resources* – the persons executing the activities
- ✓ *Other Attributes* – e.g. Cost, Roles, Abilities, Preferences, Place etc.

In summary, *event logs* describe executed operations for a given process, and classically contains *timestamps* as to the periods during which the operations were performed including the *names* and *identifiers* of the activities that were performed. There could also exist references to the sets of *resource* that was involved during the execution of the mentioned process. Moreover, the resources could be the different departments that are involved in the process, the users that carries out the operations, the documents and products etc.

3.2 Data Sources

One of the essential and most integral phase to any process mining task is the *data extraction* process. According to (Van der Aalst, 2016) societies, people and organizations are *Always On* i.e. data are collected *about anything*, *at any time*, and *at any place*. Thus, the notion of performing process mining is to analyse those captured datasets from a process-oriented perspective aimed at answering questions, and to provide insights and actionable procedures about the operational processes in view.

Many organisations have IT systems with more or less event logs that are stored as audit trails, history or transaction logs etc. (Goedertier, et al., 2009). However, most of those systems tends to store the informations in an unstructured format. For instance, event logs that are distributed or spread across different tables or are needed to be extracted from another system.

Notably, (Van der Aalst, 2016) mentions that the process of extracting the datasets have to be inspired as a result of questions that needs to be answered or resolved, rather than, the presence of large amount of event logs. In so doing, process analysts are able to answer and provide solutions to a wide range of data-to-process driven questions. For instance, the following questions could be established (Van der Aalst, 2016):

- What really happened in the past? (reporting)
- Why did it happen? (diagnosis)
- What is likely to happen in the future? (prediction)
- When and why do organisations and people deviate? (alignment)
- How to control the process better? (recommendation)
- How to redesign a process to improve its performance? (extension)

Moreover, it is also important to note that those datasets may come in form of simple flat file e.g. Excel or CVS spreadsheets, Tables within Databases, Transactional logs, web pages, e-mails, Documents in form of text or pdf etc., and quite often the datasets are sometimes not well-structured. Even now and then, the event logs may also be distributed across the various data sources due to technical or organisational reasons, and most of the time, additional effort is being required in order to collect the relevant data.

Therefore, it is indispensable and fundamental that process mining techniques relate to events within the captured data often referred to as *trace* (i.e. *sequence of related events*). Moreover, the measurement or condition of what makes the events to be associated (i.e relates) relies

entirely on the prospective process mining task as well as the trailed domain processes of interest. For example, in more or less situations, events about similar items or product may well describe or determine the resulting traces or sequence of related events. Besides, traces are found and are always present in any process mining task particularly when trailing to determine how the events or operations are executed to reach some business goal or decision.

Irrespective of the questions or viewpoint chosen by the process analysts and/or the concerned organisations, there may also be need for data cleaning (often referred to as *filtering*) i.e. removing events of a particular type. Filtering is an iterative process that corresponds to drilling down (i.e. fine-grained scoping) of the available event logs based on an initial analysis of the extraction process. For instance, the process analyst could choose to pay more attention to the activities that are performed most frequently in order to keep the model simple particularly in line with the Occam's razor principle (Thorburn, 1918; Hiroshi, 1997). Of course, the Occam's razor principle follows the natural rule that process analyst must aim for the "simplest models" which are capable of explaining the behaviours or what is observed within the datasets. Basically, in a format that takes its own advice "keep things simple".

3.3 Standard Format for Storing Event Logs

A number of events data log format have been created to standardise the data inputs used for the purpose of process mining. In practice, the standard format for storing event logs is by using the XML-based formats. The XML-based format are standard event log layouts used by many process mining algorithms: where process and activity names are normally assumed to be unique by assigning a *case identifier* and/or using both *start* and *end* times to obtain activity durations respectively. Over the next sub-sections, the thesis looks at some of the relevant file formats currently used for process mining as well as their individual attributes, benefits and limitations.

3.3.1 Mining eXtensible Markup Language (MXML)

MXML is one of such XML-based file format used for event log representation (deMedeiros, et al., 2008). The MXML format first emerged in 2003 and since then have been adopted by a number of process mining tools such as ProM. The most important part (or unit) of an MXML log file is the *Workflowlog* (Van der Aalst & Van Hee, 2004) which assemblies some group of process elements (i.e entities) and contains a well-ordered lists of *AuditTrailEntry* events where each single event is expected to have a name, thus *WorkflowModelElement*.

Also, another compulsory attribute present in the MXML file layout is the *EventType* which identifies the lifecycle transition (i.e. states if an event denotes a task that is in its start state, or complete state etc.). Other optional attributes that are present in MXML files are the event *Timestamps* (i.e. the precise dates and time the events occurred), and *Originator* (i.e. the name of the resources that initiated the events) etc. The ad-hoc extensions of the MXML format such as the Semantic Annotated Mining eXtensible Markup Language (SA-MXML) (de Medeiros & Van der Aalst, 2009) unveils the fixed format limitation of the files. Nonetheless, the shortcomings has spanned the advancement of the XES format as explained in the following sub section.

3.3.2 eXtensible Event Stream (XES)

XES is the successor of MXML (Van der Aalst, 2011). XES has been proven to be a reliable and trustworthy XML-based format used for process mining because they are less restrictive and truly extendible (Verbeek, et al., 2011; IEEE CIS Task Force on Process Mining, 2016). The standard first emerged in 2010, and in the meantime has been accepted by the process mining community as standard format for process mining since the standard can only define the attributes which could explicitly be identified practically in any settings as well as enables the capability of interchanging event data logs across different application domains or tools. Currently, the XES standard format is being used and supported by a lot of process mining tool including the ProM (Verbeek, et al., 2011; Verbeek, 2014), Disco (Rozinat & Gunther, 2012), XE-Same (Verbeek, et al., 2011) and OpenXES (IEEE 1849-2016, 2016) etc.

According to (IEEE CIS Task Force on Process Mining, 2016) the following objectives have been used as guiding principles in design of the XES standard format or layouts namely: *Simplicity* - *Flexibility* - *Extensibility* – and *Expressivity.*

Interestingly, the author in (Rozinat, 2016) notes that one of the most frequently issues with the use of XES or any of the other data types for process mining, remains with *semantics*. For instance, in settings where there exist no devoted distinctive *activity name* field within a dataset - how do the process analysts attach meaning (semantics) to the available data? or in other words, know what the exact attributes means in reality?

Nevertheless, XES has introduced the file *extension* concepts primarily for the purpose of the semantic issues. According to references (Van der Aalst, 2011; Rozinat, 2016) XES *extension* defines a number of *standardized attributes* for each *level* in the hierarchy (e.g. log, trace,

attributes) together with their *type* (e.g. string, Boolean etc) and their specific attribute *keys*. Currently, there are five standard extensions of XES defined in terms of the (i) *Concept, (ii) Life-Cycle, (iii) Organizational, (iv) Time* and (v) *Semantics* (IEEE Standards, 2016).

For instance, the extension *Concept* may describe a *Case* or *Activity* attributes of types: $case\_id$ and $activity\_name$ respectively. Notably, such extension matches the $\#case\_id(e)$ and $\#activity\_name(e)$ attribute as utilised in this thesis to classify the individual traces within the log in section 5.3.1. On one hand, the *classifier* defines the *identity* of the event by simply defining the sets of data attributes by their *attribute keys*. On the other hand, two events are considered to be similar, if they have the same values for each of the attributes.

In fact, one of the usefulness of XES format is that - it does not only provide semantics for commonly used attributes but also provides *semantic extensions* which are capable of using the underlying informations about the event logs to create new knowledge, or even more, utilized to enhance existing ones (Van der Aalst, 2011). References (Van der Aalst, 2016; Verbeek, et al., 2011; IEEE CIS Task Force on Process Mining, 2016) opines that if a log can actually describe a specific process in terms of their various domains, then IT developers or process analysts can also easily define their own domain-specific extensions. The authors notes that, indeed, additional attributes that are not defined by any extension are always allowed. Hence, the reason why the pursuits and streams in XES extensions has also inspired the advent of the latter - XESEXT (Gunther, 2009) as discussed in the next subsection.

3.3.3 eXtensible Event Stream Extension (XESEXT)

XESEXT (Gunther, 2009) has been developed as an extension to the XES file format. The XESEXT can have essential child element tags which includes - *log, trace, event,* and *meta.* The XESEXT functions serves as an enfolding container for the attributes tags definition: where the attributes that are defined in the tags is applicable to their conforming entities (i.e process elements) in line with the structure of the XES core standards.

In other words, whilst the XES core standard do not necessarily have a generic understandable purpose (i.e. does not have semantics and is mainly focused on describing the general structures of the event data logs), the actual informations enclosed within the logs are stored in the attributes. Therefore, to define the semantic information for an event log, the XES core standard makes use of the extension interface to perform such functions, whereas XESEXT permits the definition of explicit extension to the XES standards. Apparently, this results in a fixed sets of attribute for any structural levels on the XES core standard (Gunther, 2009).

3.3.4 Semantic Annotated Mining eXtensible Markup Language (SA-MXML)

The SA-MXML format (deMedeiros, et al., 2008; de Medeiros & Van der Aalst, 2009) is a semantic annotated version of the MXML file standard that incorporates an additional attribute called the *modelReference* for all elements in the log except for *AuditTrailEntry* and *Timestamp*. The SA-MXML standard format is inspired by the MXML extension. The extension incorporates *reference* between elements in the events log and concepts within an ontology - which is a great way to define or compliment the way we look at processes by associating meaning to tags or labels within the datasets (i.e. event logs).

In other words, the *modelReference* attribute points and links between instances in the log, and a list of concepts within the ontologies in order to provide a standard structuring particularly useful towards implementing semantic-based process mining approaches. The SA-MXML format is supported by tools such as ProM 5.2 (Verbeek, 2014) and ProMImport (Gu¨nther & Van der Aalst, 2006) open source frameworks for process mining.

3.4 Problems with Data Quality for Process Mining

The quality of data is imperative to any data processing or analysis procedures. According to (Rozinat, 2016) the co-founder of Fluxicon.com and Disco process mining tool (Rozinat & Gunther, 2012) in one of her article published in the Flux capacitor Blog (Rozinat, 2016) - the outcomes of process mining algorithms in relation to quality of the recorded data can be likened to the longstanding computing phrase *"Garbage in, Garbage out"*. Therefore, it is important that the event logs which serves as *input* to process mining tools/techniques are relatively of high quality in order to ensure the quality of the *outputs*. (Rozinat, 2016) opines that for an effective analysis of data or process mining – that *the quality of the underlying data is important*, or else, process analysts may run the risk of drawing the wrong conclusions.

Perhaps, in addition to resolving some of the data quality problems (i.e Incorrect logging, Insufficient logging, Semantics, Correlation, and Timing) as discussed in the following subsections 3.4.1 to 3.4.5, this thesis claims in section 6.2 that to 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 outcomes of the classifier over the given data sets) rather than focusing on the seen (observed) process instances, which in turn, is useful to further predict good classification for unseen (unobserved) instances within the available

datasets. The work have used the test event logs in (Carmona, et al., 2016) with complete total of 200 traces to validate the accuracy, error-free rate, recall and precision of the proposed classification method in this thesis by using the standard Percent of Correct Classification (PCC) as described in (Baati, et al., 2017) to assess the performance of the classifiers. Such classifications and the data cross-validation process is explained in details in section 6.2.

Furthermore, reference (Rozinat, 2016) also notes that "*the value of data is reflected in the value of decisions made*" written by Mark Norton in his comment on a recent blog post about the monetary value of data by Forrester Analyst - Rob Karel (Karel, 2011) who states that:

"…If you don't have the data, decisions can't be made (by definition), and if decisions can't be made, the organization cannot create value. So there is also an 'opportunity cost' associated with non-existent or bad data…" Mark Norton *wrote in* (Karel, 2011)

Therefore, in the following sub-sections, this thesis looks at the prevailing problems with quality of data for process mining in more details. The work looks at some of the issues in relation to data quality which the process analyst may come across during any process mining or analysis task, and consequently look at how those data quality problems can be resolved.

3.4.1 Incorrect Logging

Noise is used in the process mining field to refer to exceptional behaviours not present in the actual log (Van der Aalst, 2016). Therefore, if a specific process mining tool has the capability to abstract low frequently behaviours by displaying only the most frequent (main) process flows, then it is classed to have the ability to deal with noise. Likewise, in most cases process mining algorithms finds it difficult to differentiate between incorrect loggings from the frequent events. Thus, *incorrect logging* means that the *recorded data is wrong* (Rozinat, 2016). Besides, the issue here implies that the available datasets tends not to mirror the underlying certainty about the process in question, but instead provides wrong information about the process in reality. For example, data logs from an invoice document in a typical ERP system may be automatically scanned. Nonetheless, due to mistakes during the process of scanning the document, the "invoice ID" may be misunderstood (i.e misinterpreted) as "invoice Date" and consequently, the activities with timestamps referring to years with numbers (e.g. 2013, 2016, 2020 etc.) may appear in the logged data.

Another typical example of incorrect logging is the inconsistencies within the logged data due to human errors or alterations. For instance, an employee in an organisation could mistakenly

press the *complete* button in a workflow system right at the start of a task, or another employee may press the *start* button at the end (complete) stage of a task. Perhaps, such kind of problems could be addressed only during the process analysis phases when the employees becomes aware of such inconsistencies. Therefore, according to (Rozinat, 2016) it is imperative when carrying out a process mining task, that the process analysts and/or owners are cautious about datasets which are manually generated because they are often less reliable than data created automatically. Obviously, the data quality should be firstly inspected prior to performing the process mining or analysis task, even if there exist no doubts about the reliability or consistency of the data. Moreover, as noted in the previous section 3.4 that the results of process mining algorithms in relation to the quality of the recorded data can be compared to the longstanding computing phrase *"Garbage in, Garbage out",* the accuracy of the classification results i.e. the outcome of the classifier over a given datasets (e.g. as shown in section 6.2 of this thesis) may as well indicate that the input data is reliable and/or at the same time consistent.

3.4.2 Insufficient Logging

Modern tools for data collection and management in many information systems are often updated and simply overwritten, and in so doing, past entries or loggings are lost. Moreover, in most settings the databases only makes available the informations about current state of the systems they support, but not the complete (i.e entire) history about what has occurred in the past. Characteristically, many of such systems employs *batch logging* processes: where activities may be entered all at once at the end-of or in a day. In turn, every change that were made intermediately (i.e. halfway) are lost, and the collection of when and what has happened (i.e the process history) may not be remodelled. Therefore, whilst *incorrect logging* is about wrong data, *insufficient logging* is about missing data (Rozinat, 2016; Van der Aalst, 2011).

For example, (Van der Aalst, 2011) notes that the minimum requirement for any process mining project includes the: *Case ID*, *ActivityName* and/or *Timestamp*s to be able to model the *history* of the events or activities as performed in reality. On the other hand, (Rozinat, 2016) observes that most data mining or online analytical processing (OLAP) systems does not need the entire history of a particular process to analyse data, so for that reason, data extracted from many databases (i.e the warehouse) every so often does not carry or include all the information (i.e. data fields) that are necessary for process mining tasks.

Interestingly, the quality of outcome of any process mining task does not depend on logging too much data but exclusively on the minimum data requirement to carry out the process mining task. Although, for some particular type of process mining and analysis - additional data may be needed. For instance, when analysing a specific organisations business operations: "person" and/or "department" that executed a particular set(s) of activity needs to be present within the extracted data logs. Also, it is expected that both "start" and "end" timestamps are available within the logs, for any process mining algorithm to be able to compute the *execution-times for activities.*

3.4.3 Semantics

Predominantly, one of the biggest encounters when performing a process mining task is to discover the correct informations and to comprehend (i.e understand) what they mean (Rozinat, 2016; Carmona, et al., 2016; Rozinat, 2010). According to (Rozinat, 2016) it could be anything between really easy or very complicated to figure out the semantics (i.e metadata) informations from existing logs in many organisations databases or IT systems. Moreover, such outcomes mainly depend on how distant the logs are from the actual business logic. A typical illustration is a case where a specific business process operations could be logged directly in relation to the corresponding activity names as performed, or in settings where a process analyst may as well require process mappings between an actual business activities and some kind of hidden action code in order to be able to analyse the process.

However, according to (Rozinat, 2016) and evidently from current researches in the area of semantic process mining (deMedeiros, et al., 2008; de Medeiros & Van der Aalst, 2009) and business process intelligence (Ingvaldsen, et al., 2005; Van der Aalst, 2004; van Dongen, et al., 2016), it is best practice to work alongside process analyst who are able to mine the correct datasets as well as interpret the implication of various components of the process in question. Ultimately, in context and objectives of the process mining techniques and use of tools that supports such methods, reference (Rozinat, 2016) mentions that it helps not to try to understand everything at once but instead to focus first on the three essential elements:

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

According to (Rozinat, 2016) when these essential three elements have been identified and addressed, subsequently, one may further look for additional informations (i.e meta data) that might help improve the process mining outcomes and analysis from specific domain of interest or different perspectives.

3.4.4 Correlation

Correlation is about stitching all the element that are contained in the event log together, and in the correct way. Since process mining techniques are based on the *history* of processes, and in such manner, each and every element (i.e. the process instances) has to be mapped (modelled) from the available event data logs. The author in (Rozinat, 2016) observes that correlation is a very important factor when considering the quality of data for process mining. The author looks at correlation of data from the following examples:

- Business processes often span multiple IT systems, and usually each IT system has its own local IDs. Therefore, one needs to correlate the local process IDs to combine log fragments from the different systems (e.g. local ID from system-1 and local ID from system-2) in order to get a full picture of the process from start to end.
- Also, even within the same system, correlation may be necessary. For example, in an ERP purchase-to-pay process, purchase orders are identified by purchase order IDs and later on the invoices are characterized by invoice IDs. To get an end-to-end process perspective, the corresponding purchase order IDs and invoice IDs need to be matched.
- From time to time, there are hierarchical processes which means that the activity instances need to be distinguished to correlate lower-level events that belong to those (activity) sub processes.

In general, to resolve the problem of correlation, it makes sense to start in a simple manner in order to identify the low-level information needed to demonstrate the value of the process mining, often allied to the Occams Razor's Principles (Hiroshi, 1997).

3.4.5 Timing

Timing is another important factor for many process mining tasks used especially for ordering the events found within the captured logs. Such ordering of activities is possible due to the fact that process mining tools and algorithms calculates the history of the process elements (instances) or activities as they are sequentially performed. Thus, if the recorded timestamp(s)

are incorrect or not sufficiently accurate, it then becomes problematic for the process mining tool to produce the right order (i.e. sequence) of events history from the available data.

Accordingly, reference (Rozinat, 2016) identifies more or less of the complications with timing as follows:

- When the timestamp resolution is too low. For example, only the date of a performed activity (but not the time) is recorded. But even if the time is recorded, it may be necessary to record it at least with millisecond accuracy if many events follow each other in automated systems.

- Different timestamp granularities on different systems. For example, the timestamps in one system may be rounded to minutes. Whereas, in another system (which is also executing a part of the process) records events with 1-second resolution. Therefore, when the extracted logs from the two different systems are put together, the order of some of the events may be wrong due to the granularity difference.

- Different clocks on different systems. For example, if multiple computers record data, then these computers can have different system clocks settings. In turn, when the logs from the different computers are merged, the time differences perhaps can create problems, since they void the correct order of events.

Essentially, in an ideal world, timing of event logs must not be synchronised or summed up especially if different systems are involved, but should be exact and accurate (Rozinat, 2016).

To summarize this section of the thesis, it is certain that the quality of data is an indispensable problem that has to be firstly addressed even before commencing the process mining task. In fact, it is an undertaking which must not be ignored especially when the quality of analysis and/or output are of paramount. Besides, to discover proper or fitting models, one expects that the available datasets must consist of a *descriptive* example behaviour (Van der Aalst, 2011; Cairns, et al., 2015) which implies that most often the problems encountered while using any type of process mining algorithm or carrying out a process modelling task, is closely related to the available *event logs*. For all intents and purposes, it could take even more time to perform pre-processing of data than actual time required to perform the mining task in question. Moreover, it is important to note that not every datasets are incorrect and it's beneficial to start simple (Rozinat, 2016; Hiroshi, 1997).

Likewise, the work in this thesis focuses primarily on the *semantics* aspect of data quality to provide an easy and accurate way to effectively carry out process mining tasks. The study lays emphasis on how to differentiate the process instances (i.e. classifications) for the purpose of the research - which is to extract streams of event logs from a learning execution environment and describe formats that allows for mining and improved process analysis of the captured datasets. To this effect, the thesis describe how data from various process domains (using the case study of Learning Process) 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. As introduced in this thesis, the semantic viewpoint is captured by exploring the elements (i.e the process instances) within the event logs based on two types of probes and/or analyses, thus: (i) how to make use of the semantics that describes the available data? and (ii) how to mine the semantic information? (deMedeiros, et al., 2008).

3.5 Process Mining Algorithms, Tools and Support

Many software vendors and IT experts offer intelligent tools that provides support towards the implementation or carrying out of process mining tasks or projects. Amongst those many supporting tools is the ProM (Verbeek, et al., 2011) open source tool that supports altogether the practices and aspects of process mining. Several other commercial tools exist which supports process mining such as Disco by fluxicon (Rozinat & Gunther, 2012), including other vendors like the Celonis[14], MinIt[15], ProcessGold[16], My-Invenio[17], Worksoft[18], QPR ProcessAnalyzer[19] etc.

In this section of the thesis, the work provides insight on the strength and weakness of some of the different algorithms that supports the right use, interpretation, or extension of the PM task and resulting process models. The work explains the main ideas behind the development of the existing algorithms, their successful application towards process mining and analysis of event logs, and the problems one may need to deal with when using any of such algorithms.

14. https://www.celonis.com/en/Proactive-Insights/
15. https://www.minit.io/
16. http://processgold.com/en/process-mining-software-processgold/
17. https://www.my-invenio.com/
18. https://www.worksoft.com/products/process-mining-sap
19. https://www.qpr.com/solutions/process-mining

3.5.1 Alpha Algorithm (α-algorithm)

One of the very first algorithm developed for the primary purpose of process mining is the *Alpha algorithm (α-algorithm)* (Van der Aalst, 2011). It was first put forward by (Van der Aalst, et al., 2004), and since then many extensions of it has been proposed. A typical α-algorithm takes an event log and produces a *Petri net* that can replay the discovered model by explaining the behaviours (traces) recorded in the log.

On the other hand, *Petri net* (Van der Aalst, 2016) is one of the oldest and best investigated process modelling language that allows for concurrency which consist of *places* and *transitions* that are run by the firing rules *tokens* through the *split* and *join* notations, e.g., AND, XOR, OR gateways etc., and the discovered patterns (often referred to as footprints) are represented as Workflow nets (WF-net) (Van der Aalst & Van Hee, 2004)

According to (Van der Aalst, 2011), α-algorithms are able to discover huge amount of workflow nets that are perceived to be valid. However, there is also some limitation when using the algorithm. For example, there may exist several workflow nets that appears to have similar behaviour (trace equivalent) but the models can be *structurally* different. This means that even though the models are capable of representing the behaviours as observed within the event logs, the resultant Workflow nets may still be unnecessarily complex (i.e. representational bias).

3.5.2 Heuristic Miner (HM)

The Heuristic Miner (HM) (Weijters & Ribeiro, 2010; Weijters & Van der Aalst, 2003) is one of the existing process mining algorithm which makes use of representations similar to Casual nets (C-net) (Van der Aalst, 2016) to construct process models by taking into account *event frequency*. The main idea in heuristic mining remains in its ability to ignore paths which are not frequently executed within the process. Besides, the HM approaches are additionally robust than majority of existing process mining techniques due to its ability to focus on the frequent paths, and thus, overcomes the problem of representational bias provided by C-nets. Moreover, (Van der Aalst, 2016) notes that even though in a practical sense, casual nets are more expressive and intuitive than the traditional heuristic nets, process models discovered by the heuristic miner still appears to be highly structured and rather sequential.

Furthermore, reference (Buijs, 2014) notes that the heuristics miner (HM) has been developed to be more resistant to exceptional behaviour than most other process discovery algorithms.

For instance, when compared to outputs of the Petri nets, the author in (Buijs, 2014) observes that the resulting process tree from the heuristic miner can correctly replay all behaviour of the event log because there exist a lot of silent transitions that allow for different combinations of activities being executed as opposed to the Petri nets.

However, the author (Buijs, 2014) notes that the results of the heuristics miner are in general relaxed sound, which for most analysis techniques is not sufficient. In other words, although the HM is able to handle exceptional behaviour, the approach results in low replay fitness scores in settings where the behaviour is slightly more complicated. Besides, heuristics miner trails to consider *precision* because the behaviour of the process model is restricted, and as such, comes at the cost of *generalization*. To this end, the resulting process models are not easy to interpret since the different relationships between activities are encoded with separate transitions (Buijs, 2014).

3.5.3 Inductive Miner (IM)

Inductive Miner (IM) algorithms supports a wide-range of *process discovery* methods which are represented in form of *process trees*, i.e., notations used to represent *block structured models* which are perceived to be sound in representation (Van der Aalst, 2016; Leemans, et al., 2015). According to (Van der Aalst, 2016) whereas many other process discovery models such as the WF-nets, Petri nets, BPMN models, EPCs, YAWL models, UML etc. might suffer from dead-locks, live-locks, and other anomalies; *process trees* discovered by the inductive miner are sound by construction.

Also, reference (Van der Aalst, 2016) notes that the IM frameworks are extremely extendible and also permits for a lot of variations of the elementary method. Moreover, the *family of the inductive mining* techniques (Leemans, et al., 2013; Leemans, et al., 2014; Leemans, et al., 2014a; Leemans, et al., 2015) have variations which are capable of handling behaviours that are not frequent, and also deals with large amount of models whilst making sure the proper correct measures are conformed. For instance, the capability of rediscovering the original models within the actual limit. Thus, the IM is capable of discovering a much-wider class of events or activities, and learns sound models in settings where the α-algorithm including several other methods fails.

Currently, many extensions and refinements of the IM algorithms have been developed following the basic ideas presented through the family of IM techniques. Example of such

family of algorithms which are also available in ProM (Verbeek, 2014) are as follows: Inductive Miner-infrequent (IMF), Inductive Miner-incompleteness (IMC), Inductive Miner-directly-follows based (IMD), Inductive Miner-infrequent-directly-follows based (IMFD), and Inductive Miner-incompleteness-directly-follows based (IMCD) (Leemans, et al., 2013; Leemans, et al., 2014; Leemans, et al., 2014a; Leemans, et al., 2015; Van der Aalst, 2016).

Perhaps, IM is currently one of the leading process discovery techniques (Van der Aalst, 2016) due to its ability to discover process models that are deemed to be sound in nature. Thus:

- ✓ Flexible and Scalable
- ✓ Formal or Fitness guarantees
- ✓ Ability to convert the resulting process tress to other notations. For example BPMN models or Petri nets etc.
- ✓ Simplicity - due to its block-structure (i.e. activities are not duplicated)

On the other hand, one of the limitations of the IM is the *fall-through* which may create underfitting models in settings where there exist no process trees without silent or duplicate activities during the process of creating the observed behaviour. Moreover, the experiment carried out in (Buijs, 2014) shows that IM algorithms takes a constructive approach where more emphasis is put on replay *fitness* than on *precision*.

3.5.4 Genetic Process Mining

Genetic process mining algorithms are *search methods* that mimics the process of evolution in biological systems to discover process models by randomly distributing a finite number of points into the search space (de Medeiros, 2006; Van der Aalst, 2016). According to (Van der Aalst, 2016) whilst the α-algorithm and other techniques (e.g. the fuzzy and heuristics miner) provides models in a deterministic and direct way, the genetic algorithms appears not to be fixed but instead depends on unsystematic (random) approach to discover new methods.

(Van der Aalst, 2016) also notes that the genetic mining steps are very general, and certain choices needs to be made when actually implementing the technique. For instance, the essential choice of representation of Individuals, Initialization, Fitness function, Selection strategy (i.e. tournament and elitism), Crossover, and Mutation etc. On one hand, (de Medeiros, 2006) observes that even though such choices may be necessary, they are not sufficient enough since generally there may be more than one individual that are capable of

reproducing the behaviours within the log. On the whole, with genetic process mining, there also exist the risks of discovering underfitting (*over-general)* or overfitting (*over-specific)* individual populations.

Likewise, reference (Van der Aalst, 2016) notes some limitations with the genetic process mining. According to the author, realization of the main ideas beneath the genetic mechanisms (i.e. crossovers and mutations) is not as simple as they may suggest. Usually, in many settings model repairs are often required when those crossovers and mutations are completed. Besides, (Van der Aalst, 2016) observes that the approach appears not be somewhat effective particularly when large amount of models are being considered, and it could take longer computational times to derive models which have a fitness that is satisfactory.

On the other hand, the authors in (de Medeiros, et al., 2007) argues that such algorithms which mimics the process of evolution could be potentially used to mine and analyse event logs by trailing to make use of the conformance checking techniques to choose the representative models which are then used to produce the next generation of process models. In essence, the genetic mining approach can be used to repair the process models to reflect reality (Fahland & van der Aalst, 2012). Even though, reference (Fahland & van der Aalst, 2012) observes that similarity of the repaired model to the original model (including simplicity of the repaired model in general) is harder to achieve and may require trade-offs (e.g. allowing for all possible noisy behaviours) with respect to other quality dimensions for comparing process models, and can consequently result in spaghetti-like or complex models.

Nonetheless, genetic process mining approaches have some benefits when compared to other process mining algorithms. (Van der Aalst, 2016) observes that a combination of the heuristic miner with the genetic approaches are quite meaningful and advantageous. Consequently, to demonstrate the benefit of such mixture or hybrid algorithms, reference (de Medeiros, 2006) observes in the experiments that combination of the heuristics with genetic process mining algorithm is superior when compared to situations where it is used without the heuristic, or heuristics without the genetic operators. In the setup, the author (de Medeiros, 2006) tested the genetic algorithm on event logs from 25 different process models assumed to be noise-free (Van der Aalst, et al., 2004). The author (de Medeiros, 2006) set up four scenarios while running the genetic algorithm:

- *Scenario 1* - without heuristics to build the initial population and without genetic operators

- *Scenario 2* - with heuristics, but without the genetic operators
- *Scenario 3* - without heuristics, but with genetic operators
- *Scenario 4* - with heuristics and genetic operators.

Interestingly, results of the experiment with *heuristics and genetic operators* (scenario 4) indicate that the hybrid (combined) algorithm is superior to the other scenarios. Owing to the fact that - the *scenario 4* combines the strong ability of the heuristics to correctly capture the local causality relations, with the benefits of using the genetic operators (especially mutation) to introduce the non-local causality relations.

In general, the set up in scenario 4 (i.e. *hybrid genetic algorithm*) produces much more complete and precise models than the other approaches. Likewise, the proposed approach in this thesis appears to be a fusion theory that pursues to integrate the fuzzy models with other tools in order to enhance the information values of such type of models by carefully integrating and tuning the semantics metrics that those models lack.

3.5.5 Fuzzy Miner (FM)

Fuzzy Miner (FM) algorithms are practically used in discovering process models in a more or less precise way and to visualize complex processes. In other words, flexible and less-structured models (Rozinat, 2010; G¨unther, 2009). According to (Rozinat, 2010) fuzzy miner algorithms are applied with the goal to show understandable models for very unstructured processes. Specifically, the author in (Ingvaldsen, 2011) mentions that the FM is a one of the many existing algorithm which aims to address the problem of mining complex processes (which are unstructured in nature) by utilizing a mixture of clustering and abstraction techniques. This means that models discovered as a result of applying the fuzzy miner algorithm are able to abstract from details and aggregate behaviours that are not of interest for the process analysts (i.e visual noise) into cluster nodes (Rozinat, 2010). Even though, by referring to unstructured process, we refer that the fuzzy miner algorithms are used to produce simplified models to directly address the problems of large numbers of activities and/or highly unstructured data or behaviours, but nevertheless, tends to lack some kind of formal description (for instance, successive pattern recognition such as the *simple choice* i.e. (OR split), *parallel choice* (i.e. AND split), or *multiple choice* (i.e. XOR split) or Formal representation) when compared to the class hierarchy classifications or better still *taxonomy* described using the semantic-based fuzzy mining approach in this thesis.

Consequently, (Rozinat, 2010) notes that the results of fuzzy miner algorithms are *relaxed* in nature especially when compared with the semantics of other process modelling languages such as Petri nets or BPMN. The author also observes that if a task in a Fuzzy model has multiple successor tasks, then all of those successors will be activated once the task has been executed, even though, they do not need to be executed. In so doing, there is no explicit distinction possible between the *simple choice*, *parallel choice* and/or *multiple choice*s. Even though, (Rozinat, 2010) further notes that those patterns may emerge implicitly, but are not necessarily enforced by the model semantics.

In tools that supports the FM algorithm (e.g. ProM and Disco), users has the capacity to control the levels of detail displayed by the resulting process maps through the attributes sliders (e.g. Frequent Paths or Performance indicators) that are used to set the threshold-values. Noticeably, the resulting models 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/or on variable levels of abstraction.

However, according to (Rozinat, 2010) one of the main strengths of the fuzzy models is that they are conceived to be easily adaptable (i.e. extendible). Thus, with fuzzy miner, nodes and edges can be automatically removed or clustered by moving a slider along a particular connotation or correlation threshold scale. To this end, the relaxed nature or characteristics of the fuzzy model semantics can be summarized as follows in terms of the workflow patterns:

- *Instantiation* – the process may start at any arbitrary node in the fuzzy model, i.e. there is no exclusive starting point.
- *Branch semantics* - every node in fuzzy models has an AND split semantics, i.e an executed node enables all successor nodes.
- *Join semantics* - every node has memory-less XOR join semantics, i.e. it can be executed as soon as it has been enabled by any of its predecessor nodes, but in most cases does not *recall* how often it has been enabled.
- *Termination* - the process terminates implicitly whenever no further nodes are executed, i.e. there is no exclusive ending point, and possibly, remaining enabled nodes are ignored.

Unfortunately, even with the relaxed execution semantics e.g. as explained above, 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 (Van der Aalst, 2016) if they would be precisely represented (G¨unther, 2009). Therefore, fuzzy models are ambiguous and tends to lack the real descriptions (semantics) behind the event logs. Moreover, such approach 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 execution, but does not actually describe the semantics about relationships the process elements share within the process knowledge base which shows the limitation of such hierarchical decomposition.

Yet, fuzzy mining approaches are useful especially in settings where the analyst is interested in process discovery algorithms that are capable of providing simplified process models. Moreover, the proposed approach in this thesis 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 the analysis of the resulting models to a more conceptual level, thus, the semantic-fuzzy mining approach. Besides, because fuzzy models are conceived to be easily adaptable (i.e. extendible) by utilizing a mixture of clustering and abstraction techniques that are similar to the classification and abstraction (conceptual) method of analysis used in this thesis, the work attempts to then resolve the semantics limitation with the fuzzy algorithms by trailing to integrate semantic knowledge to the resulting fuzzy models through the series of experimentation and the proposed semantic fuzzy miner. In other words, this work has shown that it is possible to improve the information values of such type of models to some greater extent by carefully integrating and tuning the semantics metrics that those models lack.

3.6 Semantic Process Mining

In recent years, semantic technologies and its application have gained a significant interest within the field of process mining (de Medeiros & Van der Aalst, 2009; De Giacomo, et al., 2018). Such interest has prominently spanned the notion of *semantic process mining* which is currently being embraced and technically applied as a tool towards the extension and enhancement of event logs and/or models derived from traditional process mining techniques.

In other words, semantic process mining technologies trails to utilize the semantic information (i.e. metadata) about the process instances that are found within the event logs or process models to create new techniques for process mining and/or improve existing ones to better

support humans in obtaining a much more detailed and accurate results that are closer to human understanding. Indeed, this means that the *semantic-based analysis* helps to present process mining outcomes at a much more conceptual level that can be easily grasped by process owners, process analysts, or IT experts.

In principle, *semantic process mining* techniques takes the advantage of the rich semantics (Polyvyanyy & et al, 2016; De Giacomo, et al., 2018) described in event logs or models (i.e. tags or labels) of any given process, and links them to concepts in an *ontology* in order to extract useful patterns by means of *semantic reasoning*. Semantic reasoning is supported due to the formal definition of ontological concepts and expression of relationships that exist within the event logs of the process. To this effect, useful information (semantics) about how activities depend on each other in a process execution environment has been made possible, and essential for extracting models capable of creating new knowledge.

Moreover, the semantic-based approach has emerged due to the limitations identified with majority of existing process mining techniques which depend on tags or labels in event logs information, and perhaps, to a certain extent are limited because they do not technically gain from the real knowledge (semantics) that describe the tags.

To cater for this problem, the semantic-based process mining approaches prompts the main benefits provided by its utilization - which is the ability to describe the semantics behind the tags or labels in an event log or model considered useful for discovery of new knowledge. To this end, the semantic process mining approach is purely grounded on three basic building blocks (deMedeiros, et al., 2008) namely:

- ✓ Annotated event logs or models
- ✓ Ontologies, and
- ✓ Semantic Reasoning (ontology reasoners)

Notably, semantic process mining is a new area in the field of process mining and there are few existing approaches that demonstrates the capabilities of the technique.

In this thesis, the work introduces a semantic process mining approach that is directed towards the discovery and enhancement of the sets of unobserved behaviours that can be found within a process execution environment. The technique is developed in order to address the problem of determining the presence of different patterns or traces within a process knowledge base. Moreover, the study focuses on identifying meaningful information about different process

elements within the available event logs, and enriching the information values of the resulting models based on the proposed semantic-based algorithms, design framework, and application.

3.6.1 Semantic LTL Checker Algorithm

The Semantic LTL Checker (deMedeiros, et al., 2008) is one, if not the only existing process mining algorithm that trails to analyse event logs based on concepts by presenting the analysis of the processes in view at a more abstraction levels. The Semantic-LTL checker applies concepts in an *ontology* as input to parameters of a Linear Temporal Logic (LTL) formulae to formulate and answer questions about process elements (instances) by making use of the WSML2Reasoner (Bishop, et al., 1999) to infer all the necessary associations.

Reference (deMedeiros, et al., 2008) proposed the algorithm to support the development of semantic process mining tools and its applications by extending the existing LTL Checker conformance and analysis plug-in (de Beer, 2005) in ProM in order to exploit semantic annotations.

Essentially, the standard LTL Checker can be utilized for verification of properties that are defined in terms of Linear Temporal Logic (LTL) particularly beneficial when auditing logs. However, the standard LTL Checker plugins only works on labels or tags in the event logs. This means that setting values for the parameters in the LTL Checker interface is perceptibly based on event logs tags (or labels) that produces results which are syntactic in nature. In consequence, the standard LTL Checker does not benefit from the actual semantics behind those labels, which if adequately utilized, could cater for a much more robust and accurate process mining and analysis technique.

For that reason, the Semantic LTL Checker algorithm is developed with the primary aim of extending the original LTL Checker through the addition of the choices (i.e. options) to provide concepts as input to parameters of a Linear Temporal Logic (LTL) formulae.

In principle, the authors in (deMedeiros, et al., 2008) has modified the LTL Checker algorithm in the following way:

- The input formats were extended to support semantic annotations; paving the way for further development of semantic process mining techniques.

- The Semantic LTL checker has been integrated with the WSML2Reasoner framework (W2RF). The authors chose the WSML2Reasoner because their work is part of the SUPER European project, in which ontologies are defined in WSML format (Lausen, et al., 2005).

Therefore, the Semantic LTL checker (i.e. a semantic version of the conformance analysis plug-in *LTL Checker*) was developed based on the aforementioned reasons. The Semantic-LTL algorithm is only available in previous versions of ProM (i.e. ProM 5.2). The algorithm was initially built to provide support for the analysis of business processes to carry out semantically augmented auditing of event data logs. However, they can now be applied to analyse any process domain of interest or context as long at the recorded datasets contains the minimum requirement to perform process mining and are in the standard format for storing event logs for process mining and analysis.

Accordingly, to illustrate how the Semantic LTL Checker works and could be applied for the purpose of semantic process mining and analysis, this thesis have utilized the algorithm to demonstrate the need for the three basic building blocks for any semantic-based process mining approach (i.e. *annotated event logs, ontologies, and a reasoner*). For example, the work utilized the *Semantic LTL Checker* algorithm to provide answer to a real world question by pointing to concepts within the defined ontologies as shown in Figure 3.1.

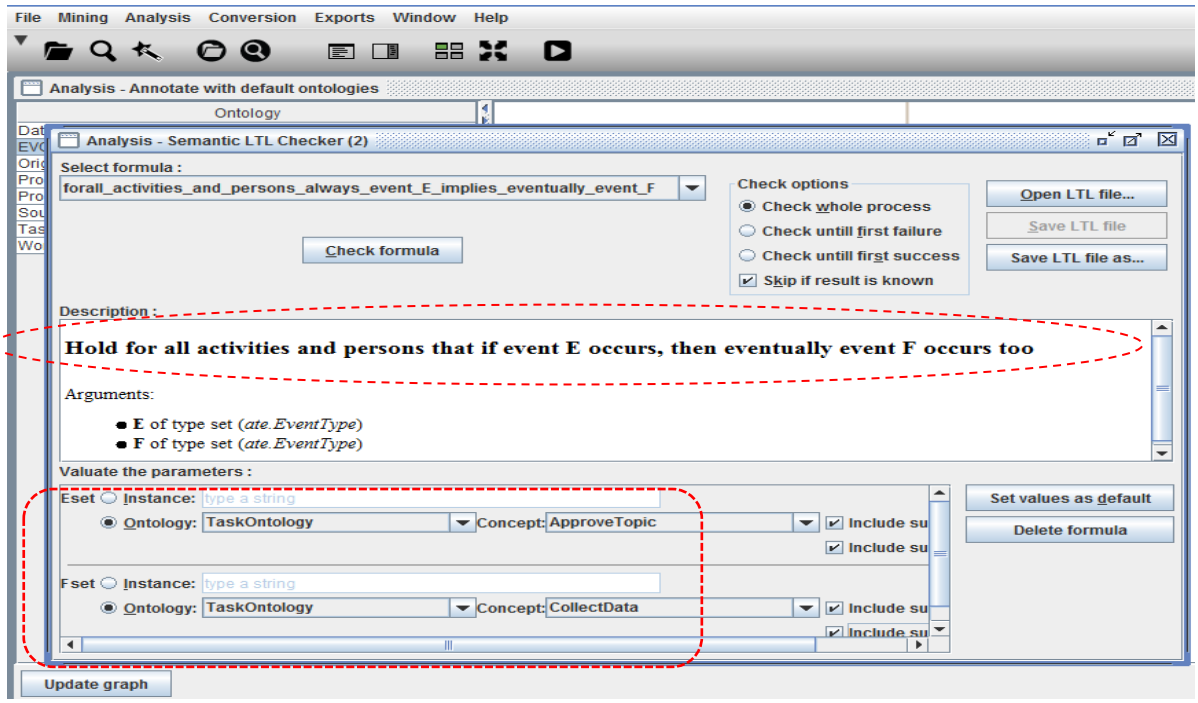


Figure 3.1 Learning Process model analysis using the Semantic LTL Checker

Therefore, by *checking* (computing) the formulae in Figure 3.1

i.e. "*forall_activities_and_persons_always_event_E_implies_eventually_event_F*"

where *Parameters* for the process model; **E** points to the Ontology (TaskOntology) and Concept (ApproveTopic), and **F** points to the Ontology (TaskOntology) and Concept (CollectData).

Apparently, the conceptual matching of the various concepts and the parameter settings provides us with meaningful information about the association that holds for all *Activities* and *Persons* within the process base that *IF* event E (i.e. ApproveTopic) occurs *THEN* eventually event F (i.e. CollectData) occurs too.

Indeed, such information can be of paramount importance especially in measuring the progress of a learner within the learning knowledge-base. For instance, since the *ApproveTopic* is within the *Define Topic Area*, it can be logically deduced that there exist a *necessary condition* for a Learner to complete the task *ApproveTopic* before they can eventually perform the task *CollectData* etc.

Notably, the associations or references to the *concepts* within the ontologies reveals interesting connection among the domain entities. Besides, it also provides a better understanding of how the different elements within the learning process base relate and interact with each other.

Apparently, such conceptual analysis is based on the incentives of the semantic process mining which forms the primary focus and is central to the objective of this thesis. Indeed, the ontological *references* associate meanings to labels (i.e. objects/data types, individuals, strings etc.) in the event log by pointing to the defined concepts in the ontologies. Moreover, the *reasoner* supports reasoning over the ontologies in order to derive new knowledge (deMedeiros, et al., 2008; Okoye, et al., 2016; De Giacomo, et al., 2018).

Over the following sub section of the thesis, the works looks at the ontological concepts and its main functions that allows for the implementation of the semantic-based process mining and analysis framework (SPMaAF), the semantically motivated algorithms and its formalizations, and subsequently describe in chapter 4 of this thesis how the work has utilised the ontology schema to develop the proposed semantic fuzzy mining approach.

3.7 Ontologies

As a collection of *concepts* and *predicates*, ontology has the ability to perform logic reasoning and bridge the underlying challenges (*semantic gaps*) beneath the event logs and models discovered especially through conventional process mining techniques with rich semantics. To make such *semantic knowledge* available, ontologies are incorporated with the process models (such as the fuzzy models used in this thesis) to pre-determine the model structure. Moreover, the method also serves as a way of representing or bridging the distances between the labels in the models and concepts within the defined ontologies.

Consequently, 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.

Moreover, for any ontological approach, the ability to make *consistency inference* is usually represented as constraints. Besides, the sets of constraints driven by the ontology have the capacity to recognize inconsistent data and outputs particularly during the pre-processing stage, the algorithm execution stage, filtering and/or interpretation stage, and the results generation.

Perhaps, 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 (Hashim, 2016) 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, (Gruber, 1995) refers to the ontological term as a *formal explicit specification of a conceptualization*, 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 the use of expressions or terms.

Thus, ontology performs the following three functions, namely: *Formal* - *Explicitness* - *Conceptualisation* – to provide hierarchical structures and representation of informations or knowledge.

In principle, ontology helps in description of the various concepts as well as the relations or associations that holds amongst those concepts in 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 and applications that are used for information retrieval and extraction, information management and integration of systems, scientific-knowledge portals, including e-commerce and web services.

Indeed, ontology has been broadly used in many other sub-fields of computer science and AI, particularly in areas that concerns information retrieval and extraction e.g. IR (Manning, et al., 2008), IE (Cunningham, 2005), OBIE (Calvanese, et al., 2016; M¨uller, et al., 2004; Hosseini, et al., 2013), database management systems (Alkharouf, et al., 2005; Calvanese, et al., 2009), information management and intelligent systems integration (Seng & Kong, 2009; De Giacomo, et al., 2018), knowledge representation (Brewster & O'Hara, 2007; Kumar, et al., 2011), and in context of this thesis, semantic-based process mining (Ingvaldsen, 2011; deMedeiros, et al., 2008; de Medeiros & Van der Aalst, 2009; Okoye, et al., 2017).

Clearly, the representation of knowledge using ontologies helps in organising *datasets* of complex structures such as the fuzzy models. Moreover, the work in this thesis 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 informations.

Even more, the metadata descriptions means that the syntactic datasets are semantically synchronized with information about the process elements which are encoded in form of entities within the ontologies (Sheth, et al., 2002; Okoye, et al., 2016; De Giacomo, et al., 2018).

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 (Dolog, 2007). Apparently, such reasoning aptitude ensures that the process 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 (Yarandi, 2013; Gruber, 1993).

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 more conceptual levels.

Furthermore, encoding ontologies with specific process domains often requires the use of formal languages (Brewster & O'Hara, 2007; Yarandi, 2013). Besides, ontological languages/vocabularies are the foundation upon which the construction of domain specific knowledge-bases are built, and most of the time, consist of semantic rules that supports the automatic reasoning and capabilities of such systems.

In the meantime, many languages or vocabularies have been technically proposed over the years primarily for the purpose of constructing ontologies. For example,

- the Knowledge Interchange Format (KIF) which are based on logic expressions (Obitko, 2007)
- LOOM8 which are based on description logics.
- FLogic (Kifer, et al., 1995)
- Cycl7

and many other advanced languages which are particularly based on the XML syntax, namely:

- the Resource Description Framework (RDF) Schema (Horrocks, et al., 2003; W3C, 2004)
- Ontology Exchange Language (XOL) (Karp, et al., 1999),
- SHOE (Heflin & Hendler, 2000) etc.

Interestingly, over the past few decades, three additional languages have been developed on top of the families of the RDFs with the intention of improving its features and to overcome its weaknesses.

(i) Ontology Inference Layer (OIL) (Fensel, et al., 2000)
(ii) (ii) DAML+OIL (Horrocks, 2002), and
(iii) (iii) OWL (Patel-Schneider, et al., 2004; Horrocks, et al., 2007)

For example, one of the limitation with the RDFs which the three additional languages trails to overcome - is the lack of proper explicit specification of resource in ontologies, property descriptions limits, and/or the domain ranges.

Currently, there exist different kinds of ontology editors that makes use of the different schema and vocabularies to describe and represent knowledge. Some examples of the ontology editors includes: Protégé[20], NeOn Toolkit[21], SWOOP[22], Neologism[23], TopBraid Composer[24], Vitro[25], Knoodl[26], Anzo for Excel[27], OWLGrEd[28], Fluent Editor[29], Semantic Turkey[30], VocBench[31] etc.

---

20. http://protege.stanford.edu/
21. http://neon-toolkit.org/wiki/Main_Page.html
22. http://www.mindswap.org/2004/SWOOP/
23. http://neologism.deri.ie/
24. https://www.w3.org/2001/sw/wiki/TopBraid
25. http://vitro.mannlib.cornell.edu/
26. http://www.knoodl.com/
27. http://www.cambridgesemantics.com/
28. http://owlgred.lumii.lv/
29. http://www.cognitum.eu/Semantics/FluentEditor/
30. http://semanticturkey.uniroma2.it/
31. http://vocbench.uniroma2.it/

Moreover, software developers or editors can create their own ontology, if they cannot find a relevant vocabulary or the existing languages are not good enough, or suitable for the use case scenarios and/or domains of interest.

In the next sub sections, the thesis specifically looks at the OWL ontology schema and the main different types used by many ontology-based approaches for formal structuring of processes. Besides, the work in this thesis also makes use of the OWL schema for implementation of the proposed semantic-based process mining and analysis technique.

3.7.1 OWL Ontologies and Schema

For the purpose of the work in this thesis, the focus is primarily on the Ontology Web-Rule Language (OWL) (W3C, 2012; Obitko, 2007; Horrocks, et al., 2007) because it is the current state of the art *logical layer* upon which semantic architectures are currently built in literature

(Lisi, 2008; Lisi & Esposito, 2007). In spite of the different facilities being offered by various ontology languages, the OWL (started by the World Wide Web Consortium) (W3C, 2012) has proved itself to be the most recent and widely accepted standard for ontology development.

According to (Lisi & Esposito, 2007) whilst debate around a unified language for rules e.g. Semantic Web Rule Language (SWRL) (Horrocks, et al., 2004) is still ongoing, the OWL mark-up language is already undertaking its standardization process at W3C.

One of the main benefit of the OWL is that the ontologies are capable of declaring the different classes and object/data properties within any given process domain. In turn, it classify those classes and properties in a taxonomy (i.e *subClass* and *subProperty* hierarchy) by assigning the *domains* and *ranges* in the same way as the RDF schema (W3C, 2004; Yarandi, 2013).

Indeed, the OWL has a better-off set of operators than many other type of ontologies. For example, the *union, intersection*, and *inverse* properties which makes it possible for *concepts* to be logically *defined* for any domain process. 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 (Kumar, et al., 2011).

Also, it is important to note that the OWL is based on *Description Logic* (DL) (Baader, et al., 2003) which makes it possible for a complete realization of the meanings of the propositions or schema. Besides, this is due to the fact that building ontologies that are fitting, especially for performing semantic-based process analysis and deploying them in an application - requires also a system with high level of *reasoning* capability. Moreover, the application of OWL based on DL (Baader, et al., 2003) allows the use of some of the existing reasoners that supports DL, especially the Pellet reasoner (Sirin & Parsia, 2004) which indeed have proven to be very effective in its ability to reason particularly at a more abstraction level.

Lastly, 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 (implication) of their developed ontologies and designs (Horrocks, 2008; De Giacomo, et al., 2018).

3.7.2 Types of OWL Ontologies

The OWL ontologies supports three expressive sub languages, namely (i) OWL Lite (ii) OWL DL, and (iii) OWL Full (Obitko, 2007) which are specifically designed by either a particular community of software developers or the end-users for use in development of their ontology projects. In turn, each of those sublanguages have its own different computational *complexities* and *expressiveness* (Horrocks, et al., 2007) as described in Table 3.1

	OWL Lite	**OWL DL**	**OWL Full**
User Support	Supports those users primarily needing a classification hierarchy and simple constraint features.	Supports those users who want the maximum expressiveness without losing computational completeness and decidability of reasoning by the systems.	Meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees.
Property Restrictions	whilst OWL Lite supports cardinality constraints, it only permits cardinality values of 0 or 1	OWL DL includes all OWL language constructs with restrictions such as type separation, thus, a class cannot be an individual or also a property, or a property cannot be an individual or also a class. OWL DL is so named due to its correspondence with Description Logics (i.e. first order logic).	In OWL Full, a class can be treated simultaneously as a collection of individuals, and as an individual in its own right. Thus, it differs from the OWL DL in the sense that an owl:DatatypeProperty can be marked as an owl:InverseFunctionalProperty . Moreover, OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary.
Functionality (Expressiveness & Decidability)	It should be simpler to provide tool support for OWL Lite than its more expressive relatives (i.e. the OWL DL and OWL Full), and provide a quick migration path for vocabularies and other taxonomies.	OWL DL was designed to support existing DL reasoning. It has decidable inference, which means, that the formal reasoning facilitates the use of deduction to infer new knowledge from the information explicitly available in an ontology due to its ability to automatically compute classification hierarchies as well as perform inconsistency checks.	OWL-full uses all OWL language primitives. Besides, it is syntactically and semantically an extension of RDF and RDFS. With such feature, the expressivity of OWL-full is more than the other two sub-languages which leads to it being undecidable. It is unlikely that any reasoning software will be able to support every feature of OWL Full.

Table 3.1 Computational Complexities and Expressiveness of the different types of OWL ontologies.

In summary, each of the OWL sub-language types in Table 3.1 is characteristically an extension of its simpler predecessor - both in what can be legally expressed and in what can be validly concluded. Perhaps, it is imperative for the developers or users (for instance, the OWL ontology with Description Logics as used in this thesis) to put into consideration what type best suits their requirements. For example, the OWL DL proves to provide the users with a more expressive restriction constructs than the OWL Lite. Besides, the optimal choice between the OWL Lite and OWL DL relies exclusively on the level of requirement by the users in terms of expressiveness.

For instance, as gathered in this thesis, particularly in section 4.6 - the work makes use of the OWL as it concerns protégé (Musen, et al., 2015) to provide additional new functions that allows for formal descriptions and structuring of concepts i.e object property assertions for any process domain of interest (using the case study of the learning process domain). The process especially as defined in section 4.6.1 to 4.6.3 involves the use of the main components of the OWL schema, i.e, Classes, Properties, Instances (individuals), and Reasoning capabilities to support the semantic modelling of the various elements (i.e different entities) that makes up the learning process, and then perform the automatic classification and/or inference of the different concepts as defined within the learning knowledge-base.

Moreover, as shown in the Table 3.1, the choice between OWL DL and OWL Full depends mainly on the level of requirement by the users in terms of meta-modeling functionalities of the RDF Schema. In essence, OWL DL is not fully compatible with the RDFs as opposed to the OWL Full. This means that not all RDF text is necessarily a legal OWL DL document, whereas on the contrary, every OWL DL document is in-fact an RDF text.

Lastly, *reasoning* support appears to be less likely when using OWL Full as opposed to the OWL DL. For example, when defining superClasses in ontologies as shown in section 4.6.1 of this thesis. Therefore, the OWL DL ontologies have lesser expressiveness influence (i.e. a reduced amount of computing complexities) than the OWL full. On the other hand, reasoners used by the OWL DL e.g. the Pellet (Sirin & Parsia, 2004) and FaCT++ reasoner (Tsarkov & Horrocks, 2006), when dealing with a decidable sub-language appears to be subject to more worst-case complexities than reasoners for OWL Lite that frequently have a desirable computational property. Even though, the Pellet reasoner have been proven to be very effective in reasoning particularly at a more conceptual level.

3.8 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* (Bechhofer, 2003) are 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.

Indeed, an additional function offered by the reasoner especially as used in this thesis is *consistency checking* of process elements and parameters. This means that based on the process description/attributes within the ontology, the reasoner is able to use the underlying informations to *check* if it is possible for any instances (individuals) to become a member of a class. Thus, 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 (Van der Aalst, 2011) 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 essence, 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.

Henceforth, 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.

Currently, there exist different kinds of reasoners capable of classifying the entities within an ontology, such as Pellet (Sirin & Parsia, 2004) which have proven to be very effective in reasoning particularly at a more conceptual level. Other kinds of reasoners has also been developed, namely: Racer (Haarslev & Möller, 2001), FaCT++ (Tsarkov & Horrocks, 2006), WSML2Reasoner (Bishop, et al., 1999; de Bruijn, et al., 2006) etc.

3.9 Summary

In short, the following Table 3.2 summarizes the various algorithms as discussed in this chapter of the thesis, and then highlights some of the benefits, limitations and impact of those algorithms as well as how are they related to this thesis.

Table 3. 2 Table of the various process mining algorithms with some of the benefits, limitations, and adaptability for the research purpose.

Algorithm	Proprietary Author(s)	Process Models	Benefits	Limitations	Are Benefits and/or Limitations adaptable for the Research purpose?
Alpha algorithm (α-algorithm)	(Van der Aalst, et al., 2004),	Petri Nets represented as Workflow nets (WF-net) (Van der Aalst & Van Hee, 2004)	Ability to discover huge amount of workflow nets that are perceived to be valid. Relatively intuitive and Simple technique that is able to deal with concurrency.	The discovered models can be structurally different, i.e, despite the similarity (trace equivalent), the resultant Workflow nets may still be unnecessarily complex (i.e. representational bias).	Yes, the thesis has focus on providing a method for accurate analysis of the process models which are intuitive and easy to understand.
Heuristic Miner (HM)	Weijters & Van der Aalst, 2003	Heuristic nets which makes use of representations similar to Casual nets (C-net) (Van der Aalst, 2016)	Ability to focus on frequent paths by ignoring paths which are not frequently executed within the process, thus, overcomes the problem of representational bias provided by C-nets and Petri nets. Thus, HM is less sensitive to noise and the incompleteness of logs. Moreover, the discovered models appears to be highly robust, structured, and rather sequential	The results of the HM are in general relaxed sound, which for most process analysis techniques is not sufficient because the resulting models are not easy to interpret and results in low replay fitness scores, i.e, the HM trails to focus more on *precision* which comes at the cost of *generalization*.	Yes, the thesis makes use of the object properties assertions and relationships just like the HM which uses the likelihood of events or activities by calculating the frequencies of relations between the tasks (e.g., causal dependency, loops, etc.)

			than most of the existing process discovery algorithms.		
Inductive Miner (IM)	(Leemans, et al., 2013; Leemans, et al., 2014; Leemans, et al., 2014a; Leemans, et al., 2015)	Process trees, i.e., notations used to represent block structured models	Produces Sound models i.e whilst other process discovery models such as the WF-nets, Petri nets, BPMN models, EPCs, YAWL models, UML etc. might suffer from dead-locks, live-locks, and other anomalies, *process trees* discovered by the IM are sound by construction. Hence, IM produces process trees which are (i) Flexible and Scalable. (ii) Formal or Fitness guarantees. (iii) Ability to convert the resulting process tress to other notations. For example BPMN models or Petri nets etc. and (iv) Simplicity - due to its block-structure (i.e. activities are not duplicated)	One of the limitations of the IM is the fall-through which may create underfitting models in settings where there exist no process trees without silent or duplicate activities during the process of creating the observed behaviour. This is because the IM algorithms takes a constructive approach where more emphasis is put on replay fitness than on precision.	Yes, unlike the IM algorithms, the Semantic-Fuzzy mining approach in this thesis considers both the replay fitness and precision when discovering the process models.
Genetic Process Mining	de Medeiros, 2006	Randomly distributed models or *search methods* that mimics the	The Genetic process mining approach tackles problems such as noise, incomplete data, non-free-choice constructs, hidden activities, concurrency, and duplicate activities	On the whole, with genetic process mining, there exist the risks of discovering underfitting (*over-general)* or overfitting (*over-specific)* individual	Yes, a combination of the genetic approaches with other techniques (e.g. the Heuristics miner) are quite meaningful and advantageous towards provision

		process of evolution.	which are often associated to the spaghetti-like or complex models. Moreover, unlike the α-algorithm and other techniques (e.g. the fuzzy and heuristics miner) which provides models in a deterministic and direct way, the genetic algorithms appears not to be fixed but instead depends on unsystematic (random) approach to discover new models.	populations. This is due to the fact that realization of the main ideas beneath the genetic mechanisms (i.e. crossovers and mutations) is not as simple as they may suggest. Usually, in many settings model repairs are often required when those crossovers and mutations are completed. Moreover, when large amount of models are being considered, it could take longer computational times to derive models which have a fitness that is satisfactory.	of much more complete and precise models that demonstrates the benefit of such *mixture* or *hybrid algorithms*. Likewise, the proposed approach in this thesis appears to be a fusion theory that pursues to integrate the fuzzy models with other tools in order to enhance the information values of such type of models by carefully integrating and tuning the semantics metrics that those models lack.
Fuzzy Miner (FM)	(Rozinat, 2010; G¨unther, 2009; (Rozinat & Gunther, 2012))	Fuzzy Models which are flexible and less-structured.	FM is one of the newer process discovery algorithms to directly address the problems of large numbers of activities and highly unstructured data and/or behaviours. FM algorithms are applied with the goal to show understandable models for very unstructured processes by providing means to explore the	Most often the fuzzy models are relaxed in nature especially when compared with the semantics of other process modelling languages such as Petri nets or BPMN. Hence, there is no explicit distinction possible between simple choice (i.e. OR split), parallel choice (i.e. AND	Yes, just like the work carried out in this thesis - FM approaches are useful especially in settings where the analyst is interested in process mining algorithms that are capable of providing simplified process models. Besides, Fuzzy models are conceived to be easily adaptable

			complex processes in an interactive manner and/or on variable levels of abstraction. Thus FM algorithms are used to produce simplified models.	split), or multiple choice (i.e. XOR split) which indicates why the FM models are ambiguous and tends to lack the real descriptions (semantics) behind the event logs.	(i.e. extendible) by utilizing a mixture of clustering and abstraction techniques that are similar to the classification and abstraction (conceptual) method of analysis used in this thesis.
Semantic LTL Checker	(deMedeiros, et al., 2008)	Semantically Annotated Models	The Semantic LTL Checker pursues to analyse event logs based on concepts by presenting the analysis of the processes in question at a more abstraction levels by making use of the WSML2Reasoner to infer all the necessary associations.	The Semantic-LTL checker applies concepts in an ontology as input to parameters of a Linear Temporal Logic (LTL) formulae to formulate and answer questions about process elements (instances) which requires the systematic knowledge of how the LTL template is applied and can only be utilized for verification of properties that are defined in terms of the LTL Logics.	Yes, just like Semantic LTL checker, this thesis introduces the Semantic-Fuzzy miner which aims to analyse events logs based on concepts by integrating the three main building blocks - semantic annotation, ontologies and reasoner. Besides, the Semantic-Fuzzy mining approach 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 as explained in details in section 6.1 of this thesis.

In summary, this chapter of the thesis has considered the main components of the process mining and semantic modelling techniques, including the tools which enables the practice and application of the techniques. Primarily, the work has identified that the application of semantic-based process mining and analysis techniques (for example, as shown with the semantic LTL Checker in section 3.6.1 and the summary Table 3.2) must focus on feeding the semantic-based algorithms with:

- ✓ *Event Logs* or *Models* which elements have references to concepts in ontologies, and
- ✓ *Reasoners* that can be invoked to reason over the ontologies used in those logs or models.

Therefore, the work in this thesis shows how semantic concepts (knowledge) and annotations can be layered on top of extracted information asset (using the case study of learning process) to provide more enhancements to the resulting process model and analysis through concept matching (i.e. *ontology classifications*) and *semantic reasoning*. Besides, semantic reasoning is supported due to the formal definition of ontological concepts and expression of relationships that exist between the process elements within the knowledge base. Thus, such conceptual information analysis and how the research has designed and implemented the framework is explained in details in the following chapter of the thesis.

Chapter 4. SPMaAF Framework Design and Main Components

The work in this thesis claims that the quality augmentation of process models is as a result of employing mining approaches that encodes the envisaged system with the three rudimentary building blocks - semantic labelling (annotation), semantic representation (ontology), and semantic reasoning (reasoner). In this chapter, the work introduces the SPMaAF design framework, and subsequently, show how the proposed frameworks and algorithms are utilized for ample implementation of the research method. The chapter then concludes with a practical description of the SPMaAF framework and the resulting semantic fuzzy mining approach and its main components including the different stages of its implementation, and then, look at use case scenario of the learning process to show how the component's integrate and is capable of analysing process models and event logs at a more conceptual level in chapter 5.

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

The design of the Semantic-based Process Mining and Analysis Framework (SPMaAF) is primarily constructed on the following building blocks as shown in Figure 4.1

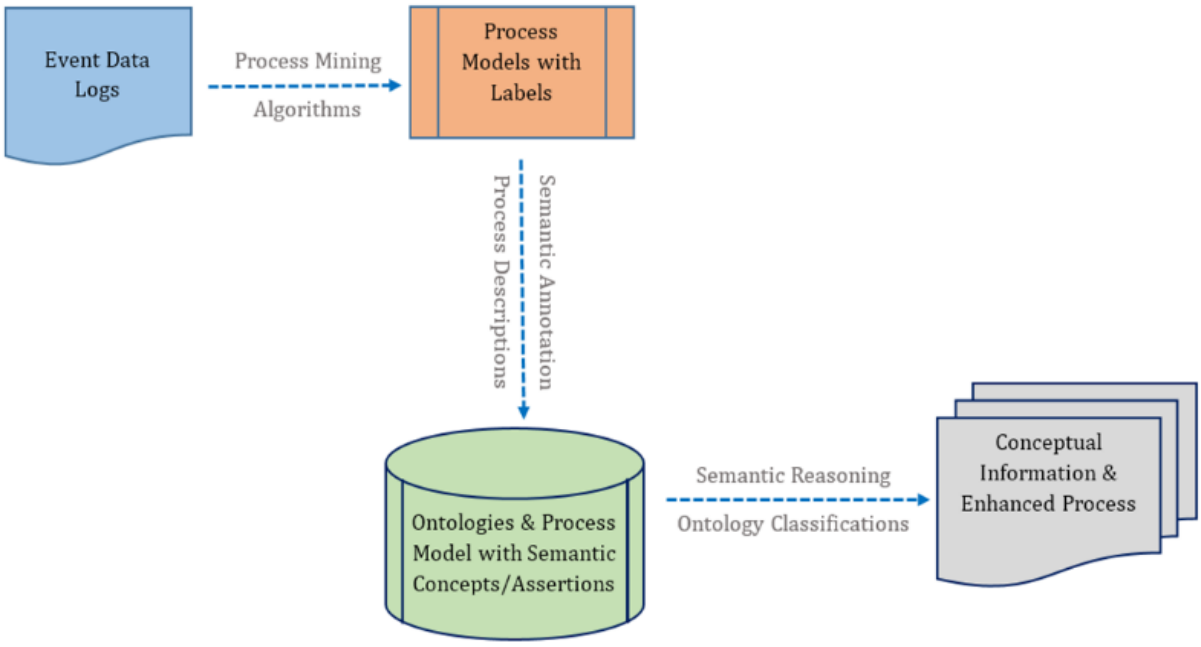


Figure 4.1 The Semantic-based Process Mining and Analysis Framework (SPMaAF)

In Figure 4.1 the work describes the proposed framework for the semantic-based process mining and analysis technique in this thesis (SPMaAF) which constitute the following processes:

- *extraction of process models from event data logs:* the derived models are represented as a set of annotated terms that links and relates to defined terms in an ontology, and in so doing, encodes the process logs and the deployed models in the formal structure of ontology (semantic modelling).
- *the inferred ontology classifications:* helps associate meanings to labels in the event logs and models by pointing to concepts (references) defined within the ontology.
- *the Reasoner* (inference engine): is designed to perform automatic classification of task and consistency checking to validate the resulting model as well as clean out inconsistent results, and in turn, presents the inferred (underlying) associations.
- *the conceptual referencing*: supports semantic reasoning over the ontologies in order to derive new information (or knowledge) about the process elements and the relationships they share amongst themselves within the knowledge base.

In short, to summarize the design framework, the key step to the application of the semantic-based process mining and model analysis approach is to focus on connecting the mining algorithms with two key core elements:

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

Indeed, the use of such semantic-based framework and its application has gained a significant interest within the field of process mining. On the one hand, the SPMaAF framework focus on making use of the semantics captured in event data logs (i.e. metadata) to create new techniques for process mining or better still support the enhancement of existing ones in order to assist humans in gaining a novel and more accurate results at a higher conceptual level mapping to the domain context as opposed to the traditional process mining techniques that tends to analysis data at the syntactic level. On the other hand, because of the semantic level of analysis, the outcome of the technique can be understood easily by the process owners, process analysts, or IT experts. 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 interpret or make use of such semantics across the different process domains.

For that reason, this thesis shows through the instantiation of the SPMaAF framework, the algorithms formalizations and the resulting semantic fuzzy mining approach - 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 (Dou, et al., 2015; deMedeiros, et al., 2008). Moreover, with this kind and level of semantic-based process mining and analysis method, 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 valuable and conceptual information.

In fact, the main difference between the SPMaAF framework described in Figure 4.1 and the traditional process mining framework in Figure 2.3 is that whilst the traditional process mining framework in Figure 2.3 tends to analyse the extracted events logs to derive some explicit and/or implicit information about the processes they support in reality without considering the *semantic* aspects of the information that are contained in the events log, the SPMaAF framework (Figure 4.1) focus on semantical integration and extension of the method in Figure 2.3 by taking into account the semantic gap that is missing with the traditional process mining framework in terms of the extracted events log and the derived process models. In other words, whilst the traditional process mining technique trails to analyse the events data logs at *syntactic levels* (i.e. labels or tags in the event logs), the SPMaAF pursues to extend and analyse the available events data logs and derived process models at a much more *conceptual level* (i.e based on concepts defined within the model)

To this end, the next section of this thesis describe the main architecture of the proposed SPMaAF framework in details, as well as, explain how this work have used the method to support the implementation of the semantically motivated algorithms and the semantic-based process mining application, thus, Semantic-fuzzy miner.

4.2 Main Components of the Semantic-based Process Mining and Analysis Framework (SPMaAF)

This section of the thesis looks at the general architecture of the proposed SPMaAF framework and how the main building blocks (annotated logs/models, ontology, and semantic reasoning) has been integrated in the development of the system. Clearly, the work summarizes in Figure 4.2 and 4.3 the various components of the semantic-based process mining approach and its implementation as follows:

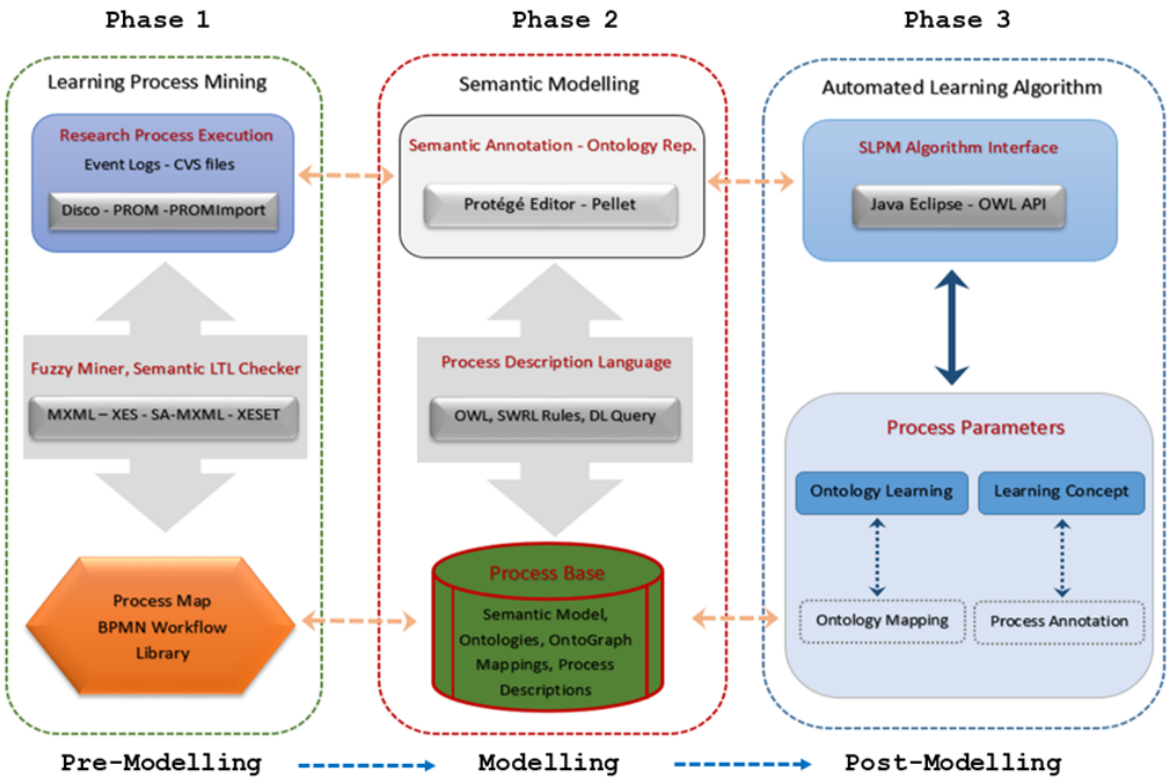


Figure 4.2 Main Architecture of the SPMaAF framework and its implementation

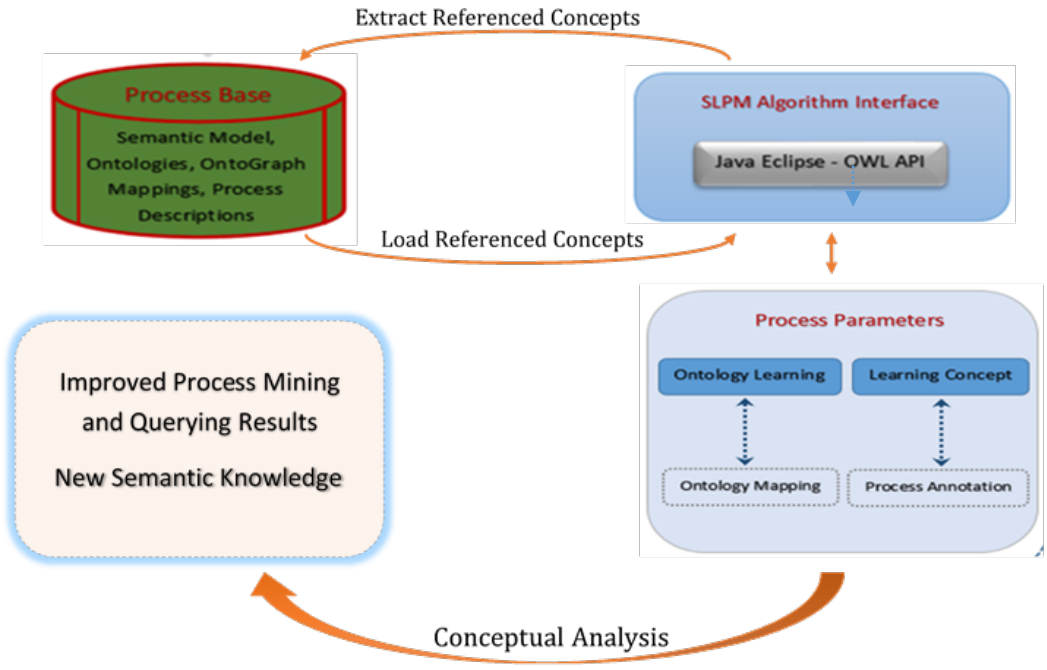


Figure 4.3 Practical aspects of implementing the proposed system and its main functions

Figure 4.2 and 4.3 represents an overview of the various components of the SPMaAF framework in this thesis including the different stages of its development and implementation. The application of the SPMaAF framework is in 3 phases as follows:

In Phase 1: the work 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 (Rozinat & Gunther, 2012) as described in sections 4.4.1 and 5.3.1. The main reason is that the resulting process map allows us to quickly, and interactively explore the processes into multiple directions and to show the individual activities workflow, and then provide platform for semantic annotation of the different process elements within the knowledge base. The proposed *Algorithm 1* in this Section 4.3 describes the procedures on how the work have implemented this phase of the process.

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 which sits at the core of the approach by using process description languages such as the OWL (Horrocks, et al., 2007) and SWRL (Horrocks, et al., 2004). In addition, the process also makes use of the Reasoner i.e. Pellet (Sirin & Parsia, 2004) - to infer the different process instances and the ontological representation (taxonomy) of the learning process model in reality. The proposed *Algorithms 2* and *3* in this section 4.3 explains the steps the work have taken to implement this stage of the process.

In Phase 3: the work implements the application used for extraction and automated 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 by linking to inferred concepts within the defined ontology. Example of the inferred underlying concepts using the OWL API and the resulting ontologies is as shown in section 5.3.2 of the thesis, whereas the Figure 4.2 and 4.3 shows the general architecture of the different phases of implementing the semantic-based process mining technique.

In short, the purpose of all the different phases of implementing the SPMaAF framework and algorithms formalization 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 learning ontology.

4.3 Proposed Algorithms and its Formalizations for Implementating the SPMaAF Framework.

This section presents and explains the different algorithms that have been proposed in this thesis as well as how this work have used the procedures to support the implementation of the SPMaAF framework and its main application in this thesis.

4.3.1 Algorithm 1

The work describe in this section the proposed *Algorithm 1* and how it have used the method to perform the process mining and model discovery (Phase 1) step in order to discover useful process models from the events data log. The process proves useful towards generation and mapping of the individual traces that makes up each of the process executions. Sections 4.4.1 and 5.3.1 describes how the proposed *algorithm 1* is implemented using process mining tools such as the Disco based on Fuzzy Miner Algorithm (Rozinat & Gunther, 2012) to generate and map the process models from the event logs for conformance checking and analysis of the individual Cases i.e. the classified traces and the sequence of activities executions.

In essence, the following *Algorithm 1* describes how this work discovers and generates process models and traces from any given event data log as follows:

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

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

Ultimately, from the proposed *Algorithm 1*, and as previously explained in section 3.1 of this thesis, we recognize that:

- A typical process model, ***M*** consists of Traces, ***T*** (i.e. Cases)
- A Trace (Case), ***T***, consist of events, ***e,*** such that each event relates to precisely one case.
- Events, ***e***, within a Trace are ordered, most often in a sequential order
- Events for any process mining task must have atleast a Case identification Id (*Case_id*) and Activity Name (*Act_name*) attributes to allow for the process model discovery.
- Other additional information may be required for ample implementation of process mining e.g. Event ID, Timestamp, Resources, Cost, Roles, and Places etc.

4.3.2 Algorithm 2

The semantic depiction (*representation*) of processes in an ontological form (Phase 2) is a very important step in this thesis that ensures the ample implementation of the SPMaAF approach. The method is aimed at unlocking information value of the Event Data Logs, ***EDL*** and the derived process Models, ***M*** by way of finding useful and previously unknown links between the process elements and the deployed models.

In fact, the purpose of the algorithm and its implementation is focused on augmenting the informative value of the resulting models by semantically annotating the process elements with concepts they represent in real time, by linking them to an ontology in order to allow for analysis of the extracted data logs and models at a much more conceptual level.

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 an enhanced conceptual analysis of the process instances.

Therefore, the following *Algorithm 2* describes how this work generates the ontology from the process models and event logs:

Algorithm 2: 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 (Gruber, 1993) ontologies, i.e $Ont \in Onts$, are formal explicit specification of shared conceptualization that can be applied in any context as exploited in this thesis to model the research 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 2*, 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 (Gruber, 1993; Gruber, 1995; Lautenbacher, et al., 2008; Lautenbacher, et al., 2009). 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

or fitting) especially during the semantic-based analysis of the process elements and/or models. Therefore, to achieve this importance step in this thesis, 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.

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 thesis. Thus, to show 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 results and its analysis from the syntactic level to a much more conceptual level.

This means that *semantic annotation* is another essential component in realizing such an approach that supports semantic-based process mining by automatically conveying the formal semantics of the derived process models and extracted logs (Lautenbacher, et al., 2009; Lautenbacher, et al., 2008). In other words, the annotated process models or logs are necessary for the semantic-based analysis and model enhancement.

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 (Lautenbacher, et al., 2009; Kiryakov, et al., 2004; Lautenbacher, et al., 2008); 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 (Born, et al., 2007) to generalize the individual entities within the domain process in view. Therefore, a *semantic annotated graph* as shown in the example Figures 4.4 which forms part of the semantic model that have been developed in this thesis is defined as follows:

Gsem = (*Nsem, Esem,Onts*) with *Nsem* = *{(n, SemAn(n))|n ∈ N}* and *Esem* = *{(nsem, n_sem)|nsem* = (*n, SemAn*(*n*))*∧n_sem* = (*n_, SemAn*(*n_*))*∧*(*n, n_*) *∈E}*. (Lautenbacher, et al., 2009).

Further details about the different Ontograph and main components of the semantic model used for the work in this thesis can be found in section 4.6 and 5.3.2 of this thesis.

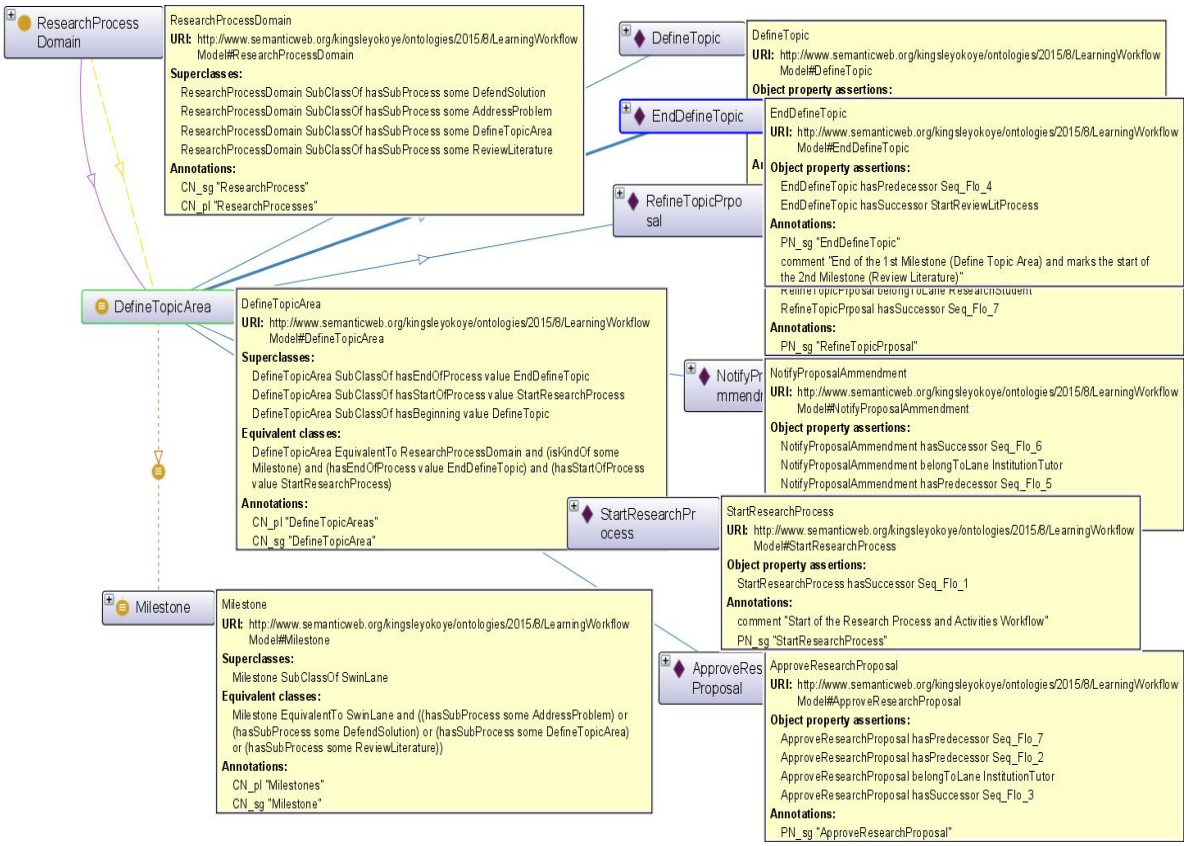


Figure 4.4 Example of semantic annotated graph with process descriptions and assertions for the different nodes (i.e. concepts) in the graph.

In fact, semantically planning of any ontology-based system such as the SPMaAF requires that all process actions within the defined ontology must include some form of semantic annotation as described in Figure 4.4.

Thus, according to the definition in (Lautenbacher, et al., 2009) Let *A* be the set of all process actions. A process action *a ∈A* is characterized by a set of input parameters *Ina ∈P*, which is required for the execution of *a* and a set of output parameters *Outa ⊆P*, which is provided by *a* after execution. All elements *a ∈A* are stored as a triple (*namea, Ina,Outa*) in a process library *libA*.

4.3.3 Algorithm 3

Accordingly, the last essential component in realizing the SPMaAF framework and its main application 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 resulting model. This means that based on the process description (i.e. assertions) within the domain ontology, the reasoner is able to use the underlying informations 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.

To this end, the following Algorithm 3 describes how this work makes use of the reasoner to classify and infer the necessary associations to produce the outputs:

Algorithm 2: Reasoning over Ontologies and Classification of Parameters 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 3, *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.

In summary, the use of ontologies and 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 (Gruber, 1993; Wimalasuriya & Dou, 2010; deMedeiros, et al., 2008; Okoye, et al., 2016). The main idea is that for any 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. Clearly, such method for semantic process mining and analysis such as the SPMaAF focuses on information about resources hidden within a process knowledge-base, and how they are related (Jareevongpiboon & Janecek, 2013; Okoye, et al., 2016; deMedeiros, et al., 2008). Indeed, reasoning on the ontological knowledge plays an important role in semantic representation of the various processes (Calvanese, et al., 2017) by allowing for extraction and conversion of explicit information into some implicit information as described in the Algorithms 1, 2 and 3. Thus, these main components as noted (Annotated logs/models, Ontology, and Semantic Reasoning) is the foundation upon which the SPMaAF framework is developed.

To this end, this work describes in details over the next sections - the steps for semantically annotating the learning process domain models which are used to demonstrate the proposed approach throughout this thesis including the sematic-based planning and the algorithms implementation.

4.4 Method for Semantic Annotation and Lifting of Process Models

A semantic-based process mining approach should present discovered models or patterns in a formal and structured manner. The primary aim must be to interpret the mining results in order to provide domain knowledge (semantics) that can help improve or further enrich the derived process models. Such type of *conceptualisation* tactics could be referred to as *semantic lifting of process models*.

This work demonstrates how the main components described in the design framework in previous section 4.1 fit and rely on each other in carrying out semantic enhancement of the discovered process models: At first, the extracted logs/models from the standard process mining techniques are represented as a set of annotated terms which links or relates to defined terms within an ontology. This makes it straightforward to represent the extracted information in an easy and yet accurate manner.

Perhaps, the *ontology* provides the means to represent the annotated terms in a formal and structured way by defining the associations (relationships) between the different process

elements in the model, and also ensures that the various range of tasks (activities) conforms naturally to the event logs and model representations. By encoding the deployed models in the *formal structure of ontology* (semantic modelling), we can then further expound the existing model.

To end with, the *Reasoner* (inference engine) is designed to perform the semantic reasoning and ontology classifications in order to validate the resulting model and clean out inconsistent outputs, and consequently, presents the inferred (underlying) semantic associations in a structured manner.

4.4.1 Annotation of Fuzzy Learning Model

Clearly, the first step towards achieving the semantic annotations is aimed at making use of process description languages/assertions to link elements in the learning models with concepts that they represent in an ontology. The purpose of the semantic annotation method is to seek the equivalence between *the concepts of the fuzzy models* derived by applying the fuzzy miner algorithm on the learning process logs and the *concepts of the defined learning domain ontology*. The method is particularly focused on making use of the process descriptions languages and notations to represent the extracted models through semantic labelling (i.e annotation) as previously explained in section 3.5.5 and the summary Table 3.2 in section 3.9 of this thesis.

To this end, in order to realize semantic annotation of the learning process models, the work applies the following process mining technique specifically as a way to achieve the target goal in Phase 1 of the SPMaAF framework and *Algorithm 1* in section 4.3.1 as follows: first, it analyse the extracted event log for the learning process using the fuzzy miner (Günther & Van der Aalst, 2007) as shown in Figure 4.5. The method involves the extraction of the process history data, specifically, the *research process* (as represented in Figure 4.6 - 4.10) by submitting the resulting event streams format to the process mining environment in Disco (Rozinat & Gunther, 2012) to help in discovery of the fuzzy model in Figure 4.5.

In turn, the approach provides us with reliable and extendible results about the data sets in order to create a process map that describes the individual traces or sequences (workflow) of the learning activities based on the proven framework of the Fuzzy Miner (Gu¨nther & Van der Aalst, 2006).

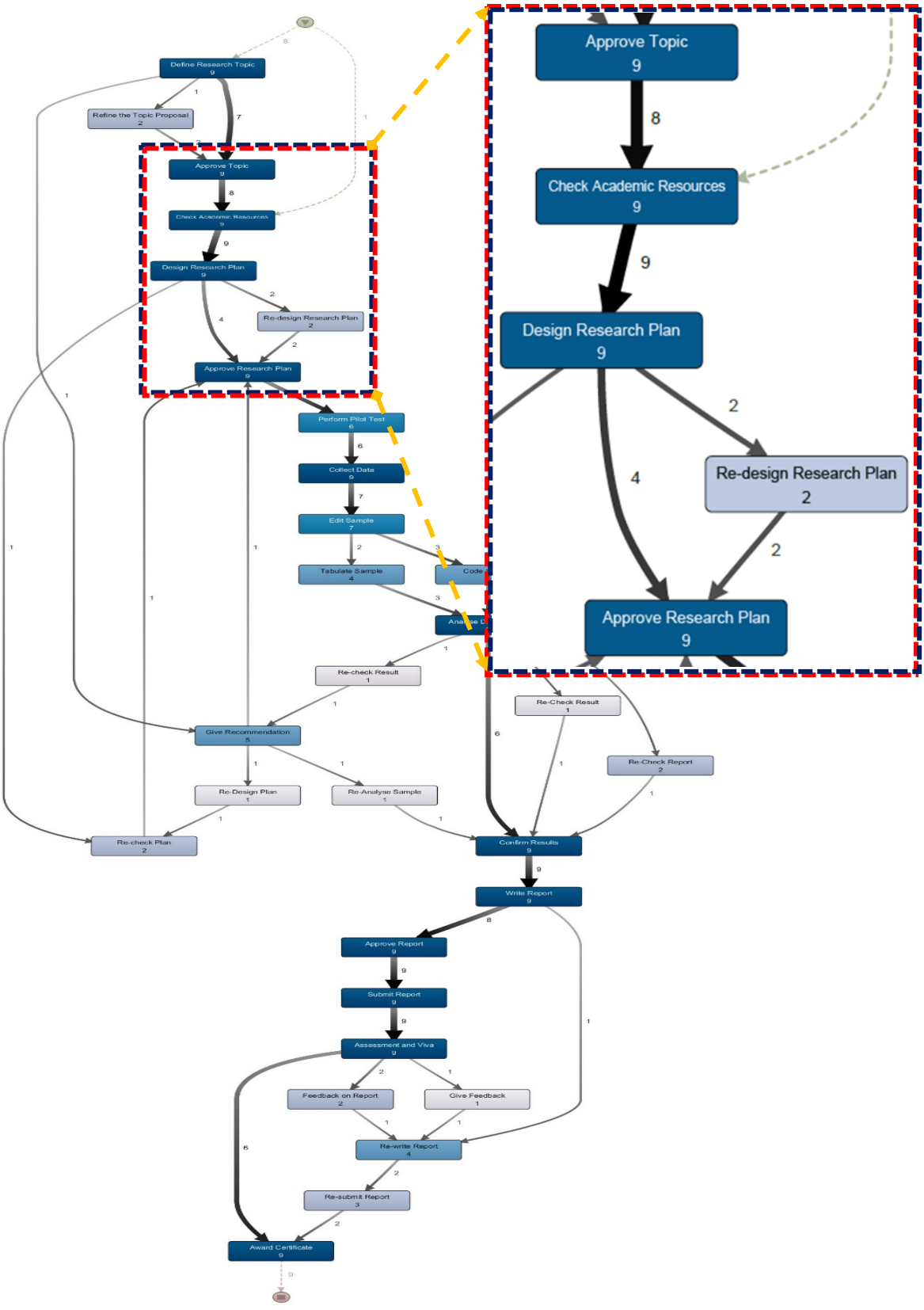


Figure 4.5 Fuzzy Model derived from mining the research process event data logs.

Consequently, suitable learning patterns were determined which enables the automatic creation of the learning process mapping in Figure 4.5. The logic is: by applying the fuzzy

miner algorithm, it allows us to see in details how the processes have been performed by revealing the underlying process mappings (i.e. the activities workflows as performed in reality), and also provides us with the opportunity to focus on the streams of learning patterns as well as visualize the paths they follow within the process.

Accordingly, the process mapping establishes a direct connection between the discovered models and the actual low-level event data about the learning process by allowing for visualising the process elements from various perspectives. Moreover, the process mapping step was necessary especially when our aim is to make the *semantics* information about the learning data readily available for further steps of mining and analysis at a much more conceptual level.

Perhaps, the next step for the semantic-based planning is to define the means for annotation of the fuzzy model and outcomes. The approach is based on a comparable representation between the attributes found within the logs and the derived model. Hence, in view of the resulting fuzzy model, the work designs and develop a BPMN model to help make available the metadata (semantics) for each element of the fuzzy model including the individual activity paths and/or sequence as performed during the process executions. To achieve this, it was necessary to construct a BPMN model as described in Figure 4.6 with notational elements capable of describing the nesting of individual learning activities (workflow) by using the event-based split and join gateways (i.e. AND – XOR – OR etc.). In fact, the purpose of the BPMN model is to help in further description of the fuzzy model, where each annotated label is now interrelated with a description of the paths, and even more, the sub-processes (milestones) they follow within the resulting model.

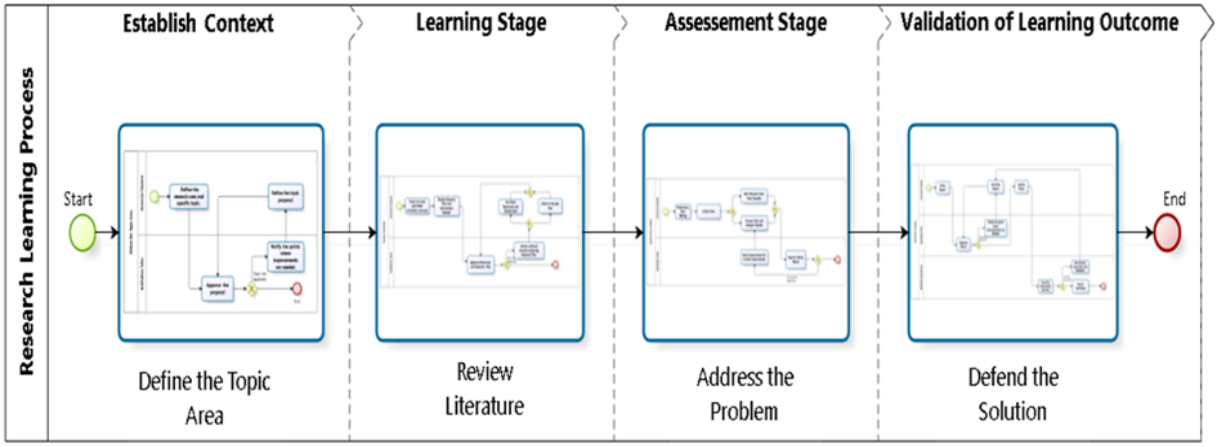


Figure 4.6 BPMN model for the Learning Process with the defined milestones

On one hand, to allow for semantic-based analysis of the resulting models, it was necessary to construct process transition information about the mapped processes, and the learning activities. On the other hand, the work based the BPMN learning model workflow, as shown in Figure 4.6 on four milestones (or sub-processes) namely: *Establish Context → Learning Stage → Assessment Stage → Validation of Learning Outcome;* with the primary aim to determine the classifications and/or grouping of the learning activities. Indeed, the classification tasks helps to explain the groups (i.e. milestones), and yet still, helps to provide a consistency check in design and computing of the semantic learning model and algorithms.

Accordingly, the work has to define and represent the BPMN models and its workflow with annotations that describes the links (relationships) between each one of the concepts in order to produce a workflow library (meta-model) that describes the concepts in the learning ontology. The purpose of performing such task is to allow for semantical representation of the Workflow Activity Patterns (WAPS) (Van der Aalst, et al., 2004) of the meta-model based on the sequences (control-flow) of the individual learning activities with their underlying classes.

To this end, Figure 4.7 to 4.10 shows example of how the work semantically represent the learning activities workflow and association of the concepts based on the deployed learning model.

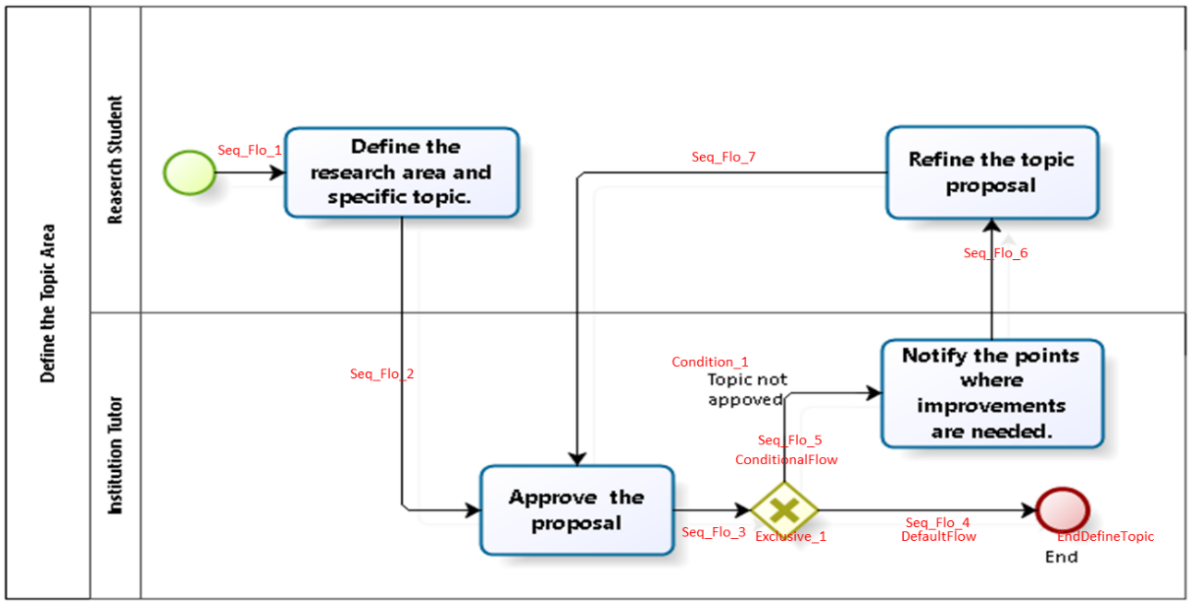


Figure 4.7 Meta description for Define Topic Area Milestone and Activities workflows

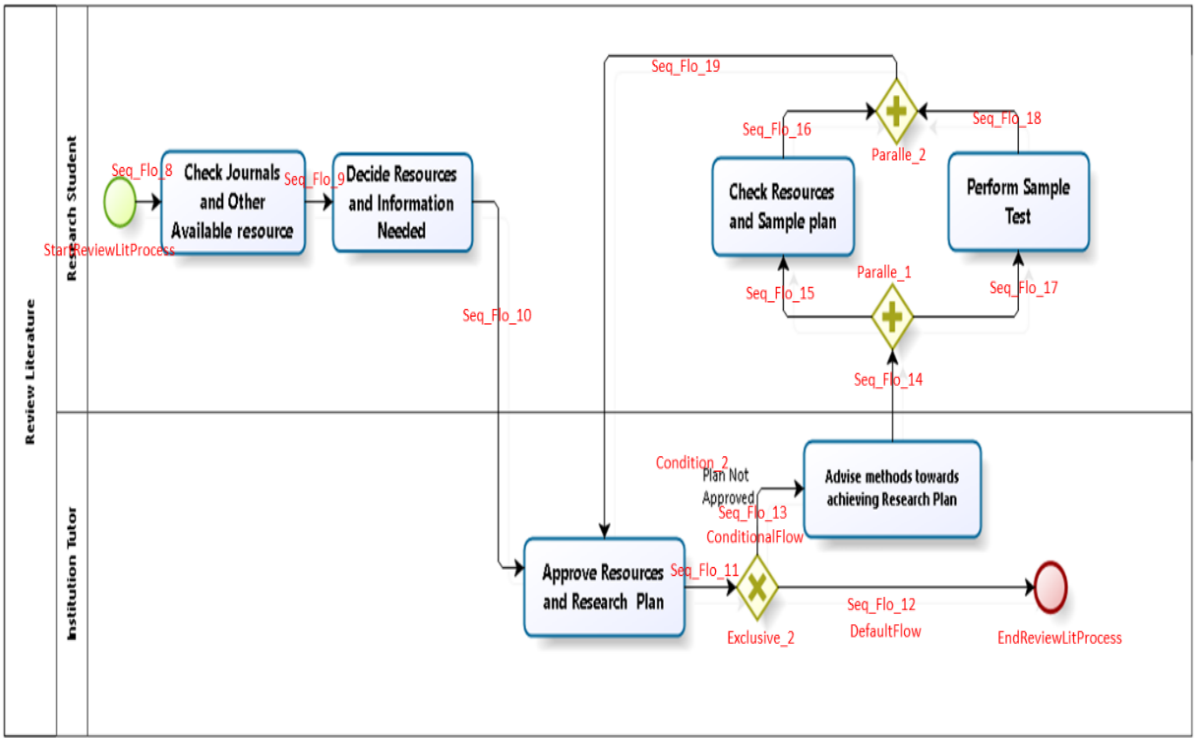


Figure 4.8 Meta description for Review Literature Milestone and Activities workflows

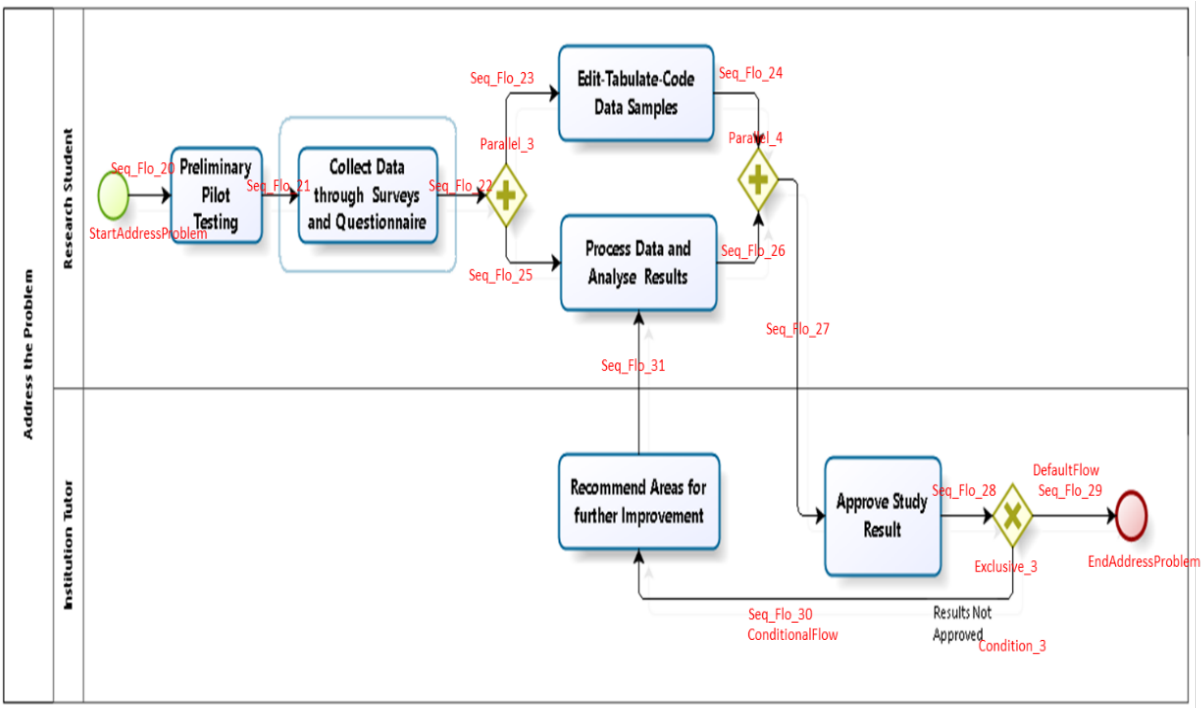


Figure 4.9 Meta description for Address the Problem Milestone and Activities workflow

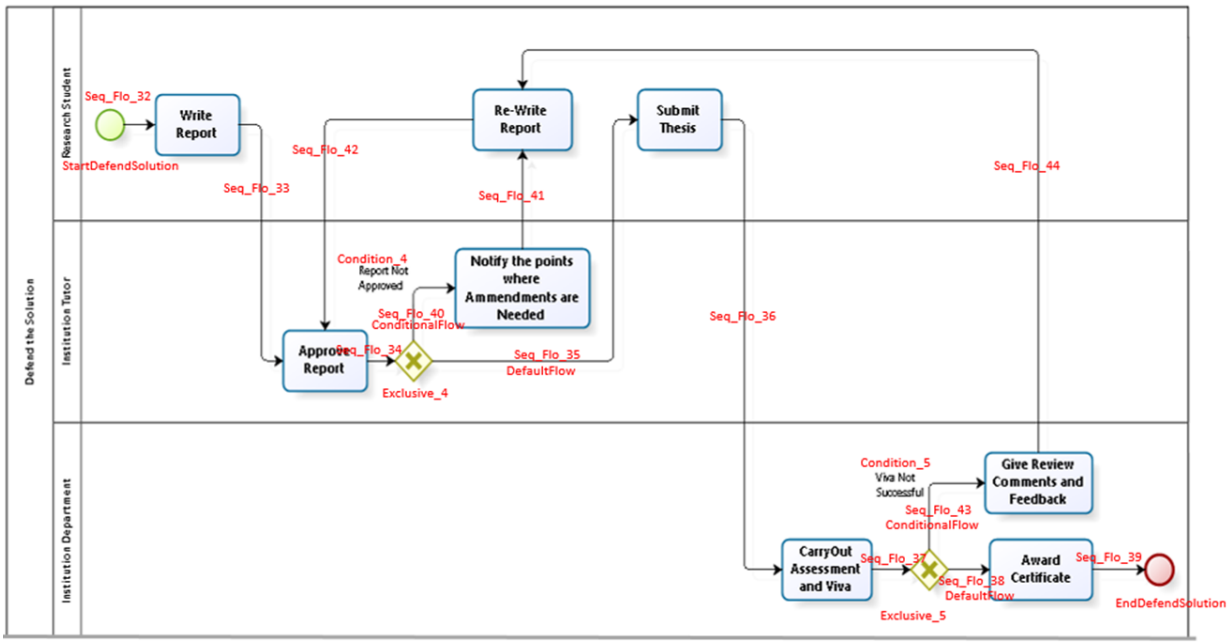


Figure 4.10 Meta description for Defend Solution Milestone and Activities workflows

Obviously, with access to such new information (*metadata*) or better still *workflow library*, it then becomes easy to generate a semantic model (i.e. learning knowledge base) that describes and/or integrates the original fuzzy model with concepts defined in an ontology.

To this end, the thesis describes in the next sections - how the work have utilised the ontological concepts and schema to integrate and support the semantically modelling and automated generation of the different process elements, individual learning milestones, relations or associations of the individual learning activities and workflow from the domain ontology standpoint.

4.5 Automated generation of Process Instances, Classes, and Learning Concepts

The following section describes the semantic learning process mining algorithm formalization and ordering for the semantic-based analysis and model in this thesis. The work shows how by constructing semantic process models and description of the process elements based on the learning activity concepts, it becomes possible and easy to determine the individual learning patterns/behaviours within the learning knowledge base. The semantic learning process mining algorithm (SLPM) formalization (*Algorithm 4*) explains the basis for the semantical analysis method.

Thus, to expound the strategies for constructing the learning activity concepts and classification of learning classes (or sub sets), the research propose the following steps:

Algorithm 4: Generating process instances, classes, and learning sub sets for defined ActivityConcepts ***AC***.

```
1: For all definite classes and process descriptions
2: Input: AC, learners prior activity list ACL_List
3: Output: AC's learning activity sequence set LS
4: Procedure: Generate Learning Activity Classes and Subsequence Sets
5: Begin
6: LS = Null
7: AC_ProcessInstance_List = Null
8: AC_LearningActivity = 0
9: LS ← LS + AC
10: For each Ci ∈ LS
11:   Ci_Precondition_List ← Get_Precondition (OWL_xml_Ci)
12:   For each Cj ∈ Ci_Precondition_List
13:     Cj_CorrespondingSubclassSet_List = Null
14:     Cj_ProcessInstance_List = Null
15:     If Cj ∉ ACL_List AND Cj ∉LS then
16:       LS ← LS + Cj
17:       Cj_ CorrespondingSubclassSet_List ← Cj_ CorrespondingSubclassSet_List + Ci
18:       Cj_ProcessInstance_List ← Cj_ProcessInstance_List + Ci + Ci_ProcessInstance_List
19:       Cj_LearningActivity = Ci_LearningActivity + 1
20:     Else If Cj ∉ACL_List AND Cj ∉ LS AND Cj ∉ Ci_ProcessInstance_List then
21:       Cj_ CorrespondingSubclassSet_List ← Cj_ CorrespondingSubclassSet_List + Ci
22:       Cj_ProcessInstance_List ← Cj_ProcessInstance_List + Ci + Ci_ProcessInstance_List
23:       If Cj_LearningActivity < Ci_LearningActivity + 1 then
24:         For each Ck ∈ LS_Subsequently_Cj
25:           Ck_LearningActivity = All (Ck_CorrespondingSubclassSet_LearningActivity) + 1
26: Return LS
27: End If
28: End For
```

Accordingly, it is important to note from the use case example of the Learning process in the previous section 4.4.1, the work refers 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 complete the research process. For that reason, it was necessary 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.

As a result, the learning activity concepts and class generation (*algorithm 4*) outlines the procedures that takes place during the process of generating the lists of process instances and the defined concepts within the learning knowledge-base. Henceforth, for each concept ***Ci*** within the current process base, first extract the precondition (prerequisite) list from its OWL file descriptions *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* we use the following standard annotation, $\boldsymbol{R}$ to refer to the research process, and $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{d}$ for the activity concepts as described in the procedure in Figure 4.11 which the work has developed to help support the semantical analysis of the resulting learning model as gathered in this thesis as follows:

SLPM Algorithm Formalization:

Let $\mathcal{L}$, be a process log for $Person$, $\boldsymbol{P}$, over $Research\ process$, $\boldsymbol{R}$, and $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, d \in \boldsymbol{R}$

where: $\boldsymbol{a}$ = DefineTopicArea Milestone

$\boldsymbol{b}$ = ReviewLiterature Milestone

$\boldsymbol{c}$ = AddressProblem Milestone

$\boldsymbol{d}$ = DefendSolution Milestone

IF $\boldsymbol{P} \ldots \boldsymbol{n}$ is a measure of the number of times $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}$ occurs in $\boldsymbol{R}$ for $Person$, $\boldsymbol{P}$

$$\boldsymbol{P} \ldots \boldsymbol{n} = |n \subseteq \mathcal{L} \in R|$$

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

THEN

SuccessfulLearner, $\boldsymbol{PSL} = |SL \subseteq \mathcal{L} \in R|$

$$\boldsymbol{PSL} = |SL \subseteq \mathcal{L}a| + |SL \subseteq \mathcal{L}b| + |SL \subseteq \mathcal{L}c| + |SL \subseteq \mathcal{L}d|$$

UncompleteLearner, $\boldsymbol{PUL} = |UL \subseteq \mathcal{L} \in R-1|$

$$\boldsymbol{PUL} = |UL \subseteq \mathcal{L} \in R-a| \ or\ |UL \subseteq \mathcal{L} \in R-b| \ or\ |UL \subseteq \mathcal{L} \in R-c| \ or\ |UL \subseteq \mathcal{L} \in R-d|$$

Figure 4.11 Semantic Learning Process Algorithm Formalization

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

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

Also, 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 activities 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, since the focus of this work as shown in Figure 4.11 is on computing the sets of individual process instances that has completed (*successful learners*) or not completed (*incomplete leaners*) the research process. We must note that to complete a research process, one must complete a set(s) of given 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 activity or milestone without completing a distinctive sequence set of learning activities that makes up the process or preceding learning concepts. So for this reason, the sum or difference in process logs for any named person, $\boldsymbol{P}$, is defined in a straightforward way:

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

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

For example, the work shows in Figure 4.11 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, $\boldsymbol{PSL}$, is the sum of the set of activities log, $\mathscr{L}$, that a learner has completed for the learning activity milestones $\boldsymbol{a}$, and $\boldsymbol{b}$, and $\boldsymbol{c}$, and $\boldsymbol{d}$. Hence

$$\textit{If } \boldsymbol{PSL} \text{ is the Class that consist of the set } |SL \subseteq \mathscr{L}a| + |SL \subseteq \mathscr{L}b\,| + |SL \subseteq \mathscr{L}c| + |SL \subseteq \mathscr{L}d\,|$$

$$\textit{Then } \boldsymbol{PSL} \text{ is the set } |SL \subseteq \mathscr{L} \in R\,|.$$

In the same way, the work also defines in Figure 4.11 that "Every Person that hasOnlyCompleteMilestone a DefineTopicArea or that hasOnlyCompleteMilestone a ReviewLiterature or that hasOnlyCompleteMilestone an AddressProblem is an UncompleteLearner".

Accordingly, the Uncomplete Learners, $\boldsymbol{PUL}$, 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 $|n \subseteq \mathscr{L} \in R|$. Hence,

If $\boldsymbol{PUL}$ is a Class that consist of the set $|UL \subseteq \mathscr{L} \in R{-}a \mid or \mid UL \subseteq \mathscr{L} \in R{-}b \mid or \mid UL \subseteq \mathscr{L} \in R{-}c \mid or \mid UL \subseteq \mathscr{L} \in R{-}d|$,

Then $\boldsymbol{PUL}$ is the set $|UL \subseteq \mathscr{L} \in R{-}1|$.

Over the next section, the work describes the tools which the research has used to practically apply and enable the implementation of the proposed algorithms and formalisations particularly the ontology schema and functions that allows the definition of the different classes and object/data properties, including the method used to classify and query the resulting model, and then demonstrate its applications and use case scenarios in chapter 5.

4.6 Main Components of the Learning Domain Ontologies

Owl ontologies makes use of the components listed earlier in section 3.7.1 to provide additional new functions that allows for formal descriptions and structuring of concepts for any process domain of interest (in this thesis – the learning process domain).

Henceforth, the main components of the OWL ontologies are namely: (i) Classes, (ii) Properties, (iii) Process Instances or individuals, and (iv) Reasoner.

To express the ontological schema in context of this thesis, the work introduces OWL as it concerns protégé (Musen, et al., 2015).

Currently, there are two main ways of modeling ontologies in protégé, namely: the (i) Frame-based, or (ii) OWL. Each one of this modelling techniques has its own user interface and features, as follows:

- *Protege Frames editor* – which allows users to build and populate ontologies that are *frame-based* in accordance with the Open Knowledge Base Connectivity Protocol (OKBC).
    - Classes
    - Slots for properties and relationships, and
    - Instances for class

- *Protege OWL editor* - which allows users to build ontology for the Semantic Web, specifically with OWL schema
    - Classes
    - Properties
    - Instances
    - Reasoning

Therefore, just like protégé, the OWL schema supports additional new facilities (*the reasoner*) that makes it possible to describe and infer the concepts for any domain of interest as we utilized in this thesis to describe the learning process.

Over the next sub-sections, the work describes in details - how the main functions and properties of those components of OWL ontologies is used for the method in this thesis.

4.6.1 Learning Model Classes

OWL *classes* are referred to as *set(s)* that contains the *individuals* by formally stating the actual (precise) requirement for any individual to become a member of that class. In other words, classes are explicit representation of concept(s), and every now and then in literature, the term *concept* is used in place of class. It is also important to note that *OWL classes* can be structured (ordered) into a superClass → subClass hierarchy which in turn are referred to as "taxonomy" (Semantic Web Primer, 2012).

A typical example of a class as utilized in context of this research is the "SuccesfulLearners" class (which are explained in details in the use case scenario of the learning process in section 5.1 of the thesis) – which contains all of the *individuals* that are classified as successful learners within the Learning process domain ontology. Superlatively, the subClasses are unified (i.e subsumed) by their superClasses. For instance, if we look at the following classes "LearnerCategory", and "SuccessfulLearners" as defined in Figure 4.12 which forms part of the semantic model that has been developed in this thesis. It's obvious that a *SuccesfulLearner* is a subClass of *LearnerCategory*. This also means that inversely the *LearnerCategory* class is a superClass of *SuccessfulLearner*. Hence, all *SuccesfulLearners* are classed as a kind of *Learners*. Therefore, every member of the class *SuccesfulLearners* are also members of the class *Leaners,* but not all kind of *Learners* are necessarily members of the class *SuccessfulLearners*. Moreover, being classified as a *SuccessfulLearner* implies that the referenced entity (i.e individual) is also a participant of the superClass *LearnerCategory*.

Besides "SuccessfulLearner" is subsumed by "LearnerCategory". Indeed, this is how taxonomy (i.e class-hierarchy) is defined in OWL ontologies as shown in Figure 4.12.

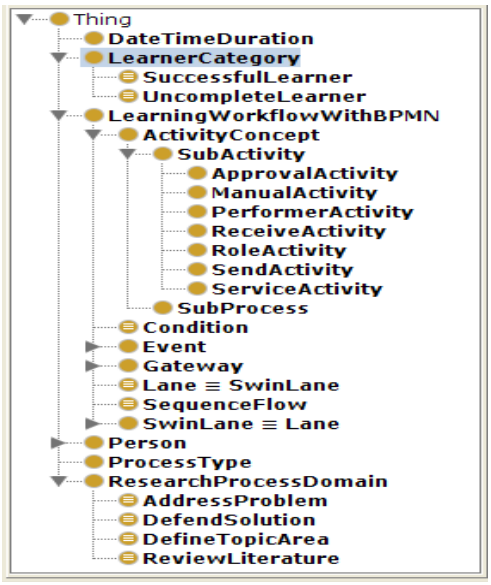


Figure 4.12 Example of class hierarchies (taxonomy) defined within the learning process domain ontology

In summary, a good practice when designing OWL Classes especially in protégé is to note that:

- ✓ *Necessary* and *sufficient* conditions are used to described *Defined Classes*
- ✓ Defined Classes are called *equivalent classes*
- ✓ Equivalent classes can be defined using the *universal restriction* properties
- ✓ Universal class restrictions can be denoted using the word "*only*" value
- ✓ Any *individual* must fulfil the universal class (object/datatype property) restriction to become a member of the specified class.

On the other hand,

- *Necessary* conditions are used to describe *Primitive Class*
- Primitive Classes are called *Superclasses* in OWL ontologies
- Primitive classes are defined using the *existential restrictions*
- Existential class restrictions are denoted using the word "*some*" value
- Any *individual* only need to fulfil the existential class (object/datatype property) restriction to become a member of a specified class.

4.6.2 Learning Model Individuals

Individuals (also referred to as *process instances*) represent objects that are found within the domain ontologies. OWL Individuals are considered to be *"instances of classes"* (Semantic Web Primer, 2012). Conceivably, an important practice when developing OWL ontologies is to explicitly indicate if a set(s) of individuals are the same as each other or dissimilar to each other. Else, the reasoner might consider them to be the same as each other, or on the other hand, different to each other.

For example, \DefineTopicArea" and \DefineTopic" might all refer to the same individual, except explicitly stated that these are two different individuals.

The Figure 4.13 shows an OWL ontology graph for the sets of individuals within the learning ontology. Notably, as described in the graph, individuals are represented as *diamonds*.

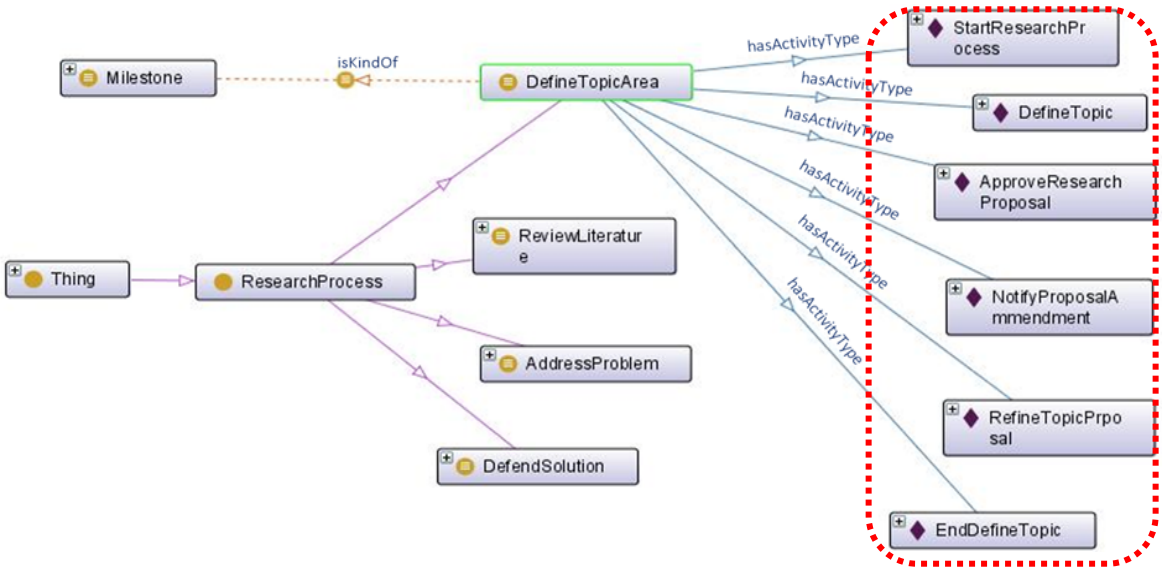


Figure 4.13 Example of individuals within the defined learning process domain ontology

4.6.3 Learning Model Properties

OWL *properties* are used to provide process descriptions (i.e. logical expressions or relations) for all *Object* and *Data* types found within the domain ontology that allows for logical reasoning and queries. Perhaps, OWL Object properties are used to create the Relations (*R*) between *subjects* and *objects* in the domain process (i.e. between classes and individuals) (Semantic Web Primer, 2012). Thus, *object properties* are binary relations on *individuals* and/or *classes* that provides the link between two individuals, or two classes, or class and individual together.

For example, in Figure 4.14 we realise that the "Individuals" from *StartResearchProcess* - to - *EndDefineTopic* are linked to the class *DefineTopicArea* through the "hasActivityType" object property.

Also, it is important to note that Owl properties may also have an inverse property for the defined property, e.g., inverse property of "hasActivityType" is "isActivityTypeOf" hence the following logical expressions can be described:

"DefineTopicArea *hasActivityType* StartResearchProcess to EndDefineTopic"

Inversely, the "StartResearchProcess to EndDefineTopic *isActivityTypeOf* DefineTopicArea"

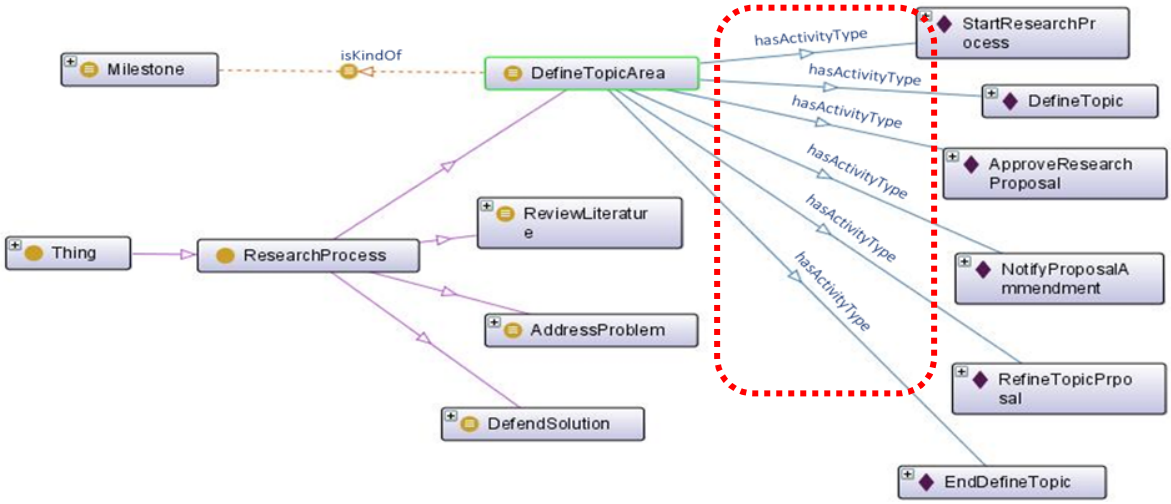


Figure 4.14 Example of object property description within the learning model ontology

Even more, OWL properties could also be restricted to have a single value (i.e. FunctionalProperty), or one or the other *symmetric* or *transitive* properties. So far, we have only looked at the "*Object properties"* (i.e. relationships between individuals and/or classes). On the other hand, the work looks at the OWL *"Datatype properties"*.

Owl Datatype properties links a set(s) of individuals to RDF literal values or XML Schema Datatypes. In principle, OWL datatype properties describes relations between an individual(s) and data-values. Hence, Datatype properties could be utilized to link an individual to an explicit data values which can be untyped or inputted. Thus, OWL Datatype properties can be used to measure numerical or literal values within the ontologies.

Indeed, *OWL Properties* (be it either object or datatype property) can be used to create restrictions which helps to define, restrict, and identify a particular set(s) of individuals that belongs to a class.

Predominantly, there are 3 main types of restrictions (Bechhofer, et al., 2004; Schreiber, 2005; Kumar, et al., 2011) that can be performed by using the OWL schema namely:

(i) Quantifier restrictions

- ✓ Existential quantifier
- ✓ Universal quantifier

(ii) Cardinality restrictions

(iii) hasValue restrictions

For the purpose of the work done in this thesis, our focus is more on the Quantifier Restrictions, especially for use in the quantitative analysis and validation of the proposed semantic approach. For example, the work defined in the semantic learning model that: a *SuccessfulLearner* is a subclass of, amongst other NamedLearnerCategory, a Person that performs some LearningActivityConcepts, who has a universal object property restriction (i.e. relationship) with the four milestones of the ResearchProcessClass.

Thus, as shown in Figure 4.15 - the *necessary condition* is: if something is a Successful Learner, it is *necessary* for it to be a participant of the Learning ActivityConcept class (*existential restriction*) and *necessary* for it to have a kind of sufficiently defined condition (*universal restriction*) and relationship with the four classes or milestones: DefineTopicArea, ReviewLiterature, AddressProblem and DefendSolution.

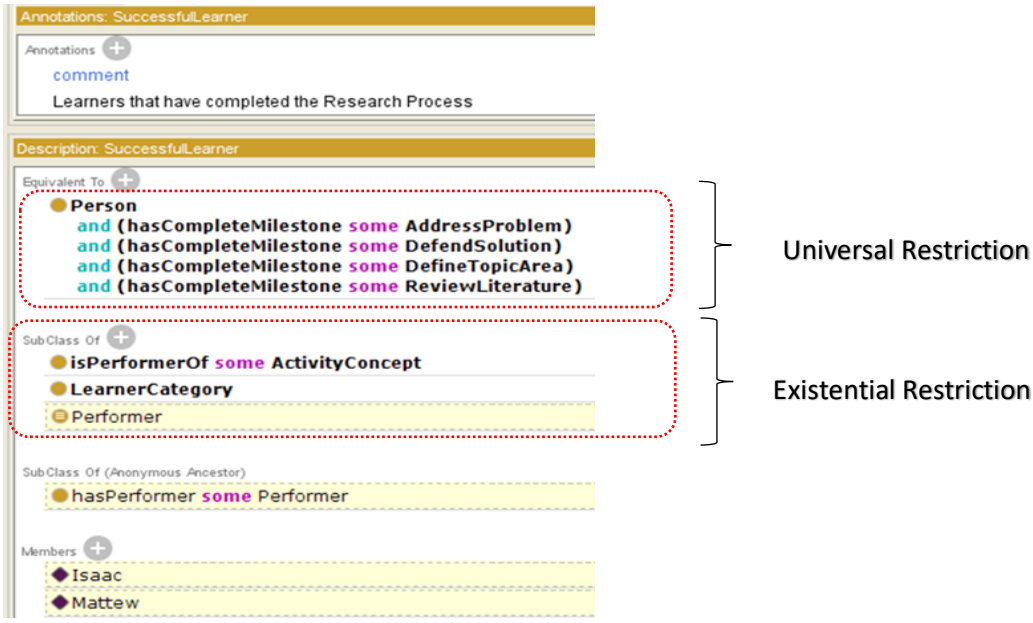


Figure 4.15 Example of OWL property restriction

Therefore, OWL restrictions are used mainly to describe *anonymous classes* (i.e. previously undefined classes). Whereas, as shown in the example Figure 4.15, *existential restrictions* are used to define the class of individual(s) that participates in at least one link (association) alongside an individual properties which is necessarily required for any individual to become a member of the class. On the other hand, *universal restrictions* are used to define the class of individual(s) which for a specifically defined property must only allow individuals that fulfils the stated condition to become part of the class. For instance, in Protégé (Musen, et al., 2015) the word `*some*' is used to signify existential restriction, whilst the words `*only*' or "*equivalent to*" are used to define universal restrictions etc.

4.7 Description Logic Queries and Reasoning

The Description Logic (DL) (Baader, et al., 2003) query is a process description language or syntax that can be used to check for consistency for all defined entities within the ontology model. DL queries makes use of the *Reasoner* as previously explained in the preceding section 3.8 to perform automatic classification of defined relationships (i.e. property assertions) that are described within the ontology.

Accordingly, the work in this thesis uses the DL query to compute and ascertain the inferred classes and individual within the learning domain ontology. 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, this thesis 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. What is the Start Activity of the second Milestone Review Literature?

DL Query: `ActivityConcept and isStartProcessOf some ReviewLiterature`

== computes and checks the start event of the second Milestone ReviewLiterature, and thus compares the activity of the second milestone with the result StartReviewLitProcess. Hence, Every Review Literature hasStartOfProcess StartReviewLitProcess.

DQ4. 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

DQ5. 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.

DQ6. 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.

DQ7. 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}

DQ8. Does Person P have Activity at least value of 3?

Example: Does Person (Danny) Activity at least three?

DL Query: `Person and hasActivity Type min 3`

== computes the Persons in the Model with a minimum of three Activities and compare if the result is equal to Person Danny

DQ9. Who performs Learning Task T?

Example: What are the different category of performers of a Learning Task in the Model?

DL Query: `Performer and isPerformerOf some ActivityConcept`

Or simply execute Role because of the SWRL description (as explained in the next section)

Role (?x) -> Lane (?x) which describes that if there exist a Role then it is also a Lane.

== the executed query and SWRL rule computes and checks for various category of Persons in the Model that performs a Learning task. Performer has been described also as a Person by the SWRL Rule: Performer (?x) -> Person (?x)

DQ10. Does Person P perform Learning Task T?

Example: Which Persons Performs a role as Institution Tutor?

DL Query: `Person and hasRole value InstitutionTutor`

== computes the persons in the model that has role as an Institution Tutor

DQ11. Does person P the first Milestone?

Example: Does person Clare the first Milestone (Define Topic Area)?

DL Query: `Person and hasActivityType some DefineTopicArea`

== compares the Persons of the First Milestone DefineTopicArea for the individual Clare i.e., checks if the persons of the first Milestone equals Clare thus, if an activity of the first Milestone is done by person Clare.

DQ12. Does Person P the second Milestone?

Example: Does person Ben the Second Milestone (Review Literature)?

DL Query: `Person and hasActivityType some ReviewLiterature`

== checks if the persons of the second Milestone equals Ben, i.e., compares the Persons of the second Milestone ReviewLiterature with Ben, thus if an activity of the second Milestone (Review Literature) is done by person Ben.

DQ13. Does person P the Third Milestone?

Example: Does Paul the Third Milestone (Address Problem)?

DL Query: `Person and hasActivityType some AddressProblem`

== compares the Persons of the Third Milestone with Paul i.e., Checks if the persons of the Third Milestone equals Paul, thus if an activity of the Third Milestone DefendSolution is done by person Paul?

DQ14. Does person P the Last Milestone?

Example: Does person Danny the Last Milestone (Defend Solution)?

DL Query: `Person and hasActivityType some DefendSolution`

== computes and check if the result of Persons in the Last Milestone DefendSolution is equal to person Danny i.e., compares the Persons of the Last Milestone with Danny, thus if an activity of the Last Milestone (Defend Solution) is performed by person Danny?

DQ15. For all Activities always Event E implies eventually Event F?

Example: For all Activities always Event (End) implies eventually Event (Start)

DL Query: `Event and hasSuccessor some Start`

== describes and computes that - Hold for all activities that if event End occurs, then eventually event Start occurs too. We use this to define the Start and End of each Milestone in the Model e.g., we define that = Every DefineTopicArea is a ResearchProcess that isKindOf a Milestone and hasEndOfProcess EndDefineTopic and hasStartOfProcess StartResearchProcess.

It is possible to then ask: Does the End of a Milestone eventually means the Start of the next Milestone too?

Question 16 below answers this query

DQ16. Eventually Event E and then F?

Example: Eventually EndDefineTopic and then StartReviewLitProcess?

DL Query: `Event and hasSuccessor value StartReviewLitProcess`

== checks and compares that the End of the DefineTopicArea during the research process execution process means the Start of the next milestone ReviewLiterature.

DQ17. Finally Person P?

Example: List all the Persons that performs an Activity in the Research Process?

DL Query: `Person and hasActivityType some ResearchProcess`

== computes any Person P that is a performer of a Learning Task within the Model. This can also be described using the SWRL rule as explained in the next section of the thesis:

Person (?Performer), hasActivityType (?Perfomer, ?ActivityConcept), isPerformerOf (?ActivityConcept, ?Role) -> isPartOfResearch Process (?Performer, ?Role)

== the SWRL rule describes that any person that performs a Learning Activity is then automatically part of the Research Process.

4.8 Semantic Web Rule Language

The Semantic Web Rule Language (SWRL) (Horrocks, et al., 2004; Bechhofer, et al., 2004) extends OWL Description Logics with "rules" while supporting the existing semantics or sentence structures (syntax) in OWL ontologies. According to (Horrocks, et al., 2004) SWRL combines OWL DL ontologies with Rule Markup Language (RML). Reference (Yarandi, 2013) also notes that:

- *Semantically*: SWRL rules derive formal meanings through extension of the OWL DL model-theoretic semantics, and
- *Syntactically*: SWRL is based on OWL XML presentation syntax (Horrocks et al. 2005).

Furthermore, the semantic web rule language format trails to add new sets of axiom to the OWL DL ontologies which includes *horn-like* rules (i.e. First order logics) (Wang & Kim, 2006) or better still *Association Rule Learning* (Han, et al., 2011; Okoye, et al., 2016; Okoye, et al., 2014) syntax format to broaden the semantics and formal structures of ontologies.

In principle, SWRL increases the expressiveness of OWL ontologies (Yarandi, 2013; Bechhofer, et al., 2004). According to (Vassileva & Bontchev, 2009) many selection of rule-engines could be utilized when applying the SWRL rules because they don't forcefully state (i.e restrictions) on how the reasoning have to be executed. For instance, the universalClass or equivalentClass used to determine if an expression within the ontology is true or false. This is where the SWRL function is paramount as it does not decide how such restrictions should be performed, but rather extends the axioms (facts) by adding rules to the already pre-defined assertions within the ontology.

To that effect, a SWRL enriched ontology comprises of a mixture of OWL concepts and rules as described below in the learning model:

```
1) ResearchProcess(?x) -> LearningWorkflowWithBPMN(?x)
2) Performer(?x) -> Person(?x)
3) Person(?Performer), hasActivityType(?Performer, ?ActivityConcept),
   isPerformerOf(?ActivityConcept, ?Role) ->
   isPartOfResearchProcess(?Performer, ?Role)
4) ProcessType(?x) -> PerformerActivity(?x)
```

```
5) Start(?start) -> DateTimeDuration(?start)
6) End(?end) -> DateTimeDuration(?end)
7) isActivityTypeOf(?x, ?y), isRoleOf(?y, ?z) -> isSubProcessOf(?x, ?y)
8) hasDefaultFlow(?Exclusive, ?SequenceFlow) ->
   hasSuccessor(?Exclusive, ?SequenceFlow)
9) hasConditionalFlow(?Exclusive, ?SequenceFlow) ->
   hasSuccessor(?Exclusive, ?SequenceFlow)
10)     Exclusive(?x) -> Condition(?x)
11)     Role(?x) -> Lane(?x)
12)     LearningWorkflowWithBPMN(?x) -> Pool(?x)
13)     hasActivityType(?x, ?y), hasRole(?Role, ?z) ->
   belongToPool(?Role, ?z)
```

As described in the example rules, the SWRL syntax are of the form:

Antecedent ! Consequent (i.e written as a1 ^ a2 ^ ::: ^ an)

or yet still *horn-like* rule, where the *Antecedent* represents the *body* and the *Consequent* represents the *head*.

Therefore, a typical SWRL Syntax is as follows:

$$atom \wedge atom .... \rightarrow atom \wedge atom$$

where the *Antecedent* and *Consequent* could consist of multiple atoms or still be empty.

Accordingly, atoms are syntactically expressed in the following form (Yarandi, 2013):

- C(x) where C is an OWL description and x is an OWL individual variable or a data value.
- P(x; y) where P is an OWL object property and x and y are OWL individual variables or data values.
- Q(x; y) where Q is an OWL data property and x and y are OWL individual variables or data values.
- B(x1; x2; :::) where B is a built-in relation and x1; x2; ::: are OWL individual variables or data values.
- sameAs(x, y), differentFrom(x, y) where x, y are OWL individual variables or data values.

Generally, the informal meaning of any rule (e.g. the SWRL) states that: *whenever* a condition defined in the *Antecedent* holds, *then* all condition(s) stated in the *Consequent* must also hold. Apparently, such conditions (or rule) is similar to the Association Rule Learning (Han, et al., 2011; Okoye, et al., 2014) also used for event logs or data mining to perform process analysis. Thus:

IF (X) THEN (Y)

where X = Antecedent (e.g. learning pattern) and Y = Consequent (e.g. the pattern extension)

Indeed, the rule expressions as shown in this section shows some of the example of SWRL rule and syntax which are described within the learning model ontology developed in this thesis. The work creates the SWRL rules to associate the existing domain classes with the right concepts in order to automatically infer the whole learning domain ontology.

Thus, from the definitions in the SWRL rules, the work explains some of the definition and functionality of the *Rules* as are implemented in the resulting semantic learning model ontology as follows:

SQ1. `Person (?Performer), hasActivityType (?Perfomer, ?ActivityConcept), isPerformerOf (?ActivityConcept, ?Role) -> isPartOfResearch Process (?Performer, ?Role)`

== this Rule describes that any person that performs a Learning Activity classified as a Role is then automatically part of the Research Process.

SQ2. `hasDefaultFlow (?Exclusive, ?SequenceFlow) -> hasSuccessor (?Exclusive, ?Sequence Flow)`

== describes that if there exists a Default flow for an Exclusive gateway then this flow is also a Successor i.e., If X hasDefaultFlows Y then Y is DefaultFlowOf s X.

SQ3. `Research Process (?x) -> Learning Workflow With BPMN (?x)`

== describes that if there exists a Research process then it is automatically a Learning Workflow

SQ4. `LearningWorkflowWithBPMN (?x) -> Pool (?x)`

== describes that if there exists a Learning Workflow then it is also a Pool

SQ5. `Role (?x) -> Lane (?x)`

== describes that if there exists a Role then it is also a Lane

SQ6. `hasActivityType (?x, ?ActivityConcept), hasRole (?ActivityConcept, ?Role) -> belongToPool (?x, ?Role)`

== describes that if there exists a learning activity which is performed under a particular Role, then this activity belong to the pool of that Role. Role has also been described as a Lane.

4.9 Summary

In this chapter of the thesis, the work describe the proposed SPMaAF framework, the main methods and algorithms used for integration and implementation of the approach in order to improve the analysis of the events log and discovered models in this thesis.

Typically, the works recognizes that much of the effort in developing sematic-based process mining approaches relies mainly on constructing an effective system that integrates the three main building blocks (i.e. annotated logs and models, ontology and semantic reasoning).

The following Table 4.1 shows a summary of the thematic focus and targeted goal for each of the different phases of implementation of the SPMaAF framework and proposed algorithms as described in this chapter.

	SPMaAF Framework	***Algorithm 1***	***Algorithm 2***	***Algorithm 3***	**Target Sections**
Phase 1	**X**	**X**			4.1 – 4.3; 4.3.1; 4.4
Phase 2	**X**		**X**	**X**	4.1 – 4.3; 4.3.2; 4.5 – 4.6
Phase 3	**X**			**X**	4.1 – 4.3; 4.3.3; 4.7 – 4.8

Table 4. 1 Different Phases of implementing the SPMaAF framework and thematic target of the proposed Algorithms.

Indeed, from this chapter and the thematic focus in Table 4.1, the work have presented the motivation behind the proposal of the SPMaAF framework and its main application in real-

time. Thus, whilst the process mining and semantic annotation process in Phase 1 of the thesis is focused on describing the meaning of the process models and its attributes, the ontology description process in Phase 2 is devoted to binding together the different concepts, classes and properties in ways that maximizes their influence and outcomes. Consequently, the semantic reasoning process and capabilities described in Phase 3 focus on providing a more conceptual analysis of the underlying ontologies and the process description or assertions that are closer to human understanding. The work also looks at the main components of the proposed semantic-based approach, including the available tools that enables its application. The work notes that the best way to create such systems is to make use of tools that supports the different components particularly ontology which every now and then are required to maintain consistency of the process elements. Without a doubt, the use of a reasoner to compute relations between various entities (process instances) in the ontology is practically possible, especially when building such 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 systems supports the application of rules such as the 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 and concepts within the ontologies.

Chapter 5. Implementation of the Semantic Fuzzy Mining Approach and Case Studies

This chapter of the thesis shows how the proposed semantic-based process mining and analysis framework (SPMaAF) is applied to answer real time questions about any given process domain as well as the classification of the individual process elements that can be found within the event logs and the discovered models. The chapter illustrates this through the use case scenario of the *learning process* and data about a *real time business process* used for the work in this thesis. The chapter finalizes with a practical description of the SPMaAF framework referred to as the Semantic fuzzy miner - including the integration of the different stages of its implementation in order to show how the various components fit and is capable of analysing process models and event logs at a more abstract level.

5.1 Case Study of the Learning Process and Use Case Scenario

The case study in this thesis is based on running example of the Research Process domain (as described in section 4 particularly sub section 4.4.1). The work makes use of the event log about the research process to prove how the proposed semantic-based approach is applied to answer real time questions about a learning process, as well as, in validation of the experimentations.

In the case study example, 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.

Indeed, 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 log and 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 this reason, the study shows in this thesis 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 (i.e from Defining the Topic Area –to- Review Literature –and- Addressing the Problem –then- Defending the Solution) as explained in Figure 4.6 – 4.10 in section 4.4.1 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 way, it becomes possible to logically ascertain which individuals has successfully completed the research process or not.

In short, the work in this thesis shows how it uses the case study of the Research Learning Process domain with focus on the use case scenario of the *successful* and *uncomplete* learners to demonstrate the capability of the SPMaAF framework and proposed algorithms by analysing the learning activity logs based on concepts rather than the event tags (i.e. labels) about the process. In turn, presenting the process mining results at a more conceptual level of analysis. Such method of conceptual analysis in explained in more details in the following section 5.2 of the thesis.

5.2 Semantic Representation and Modelling of Research Learning Process.

In this section of the thesis, 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). Particularly, the research focus is therefore on the use case scenario of the *Successful* and *Uncomplete* learners.

Accordingly, the work shows 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 section 4.4.1) 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.

Indeed, 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. Hence, as described in Figure 5.1 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.

Accordingly, Figure 5.2 to 5.5 shows the different milestones of the research process and the resulting activity concepts graph and relations mapping between the process instances (entities) that are defined in the semantic model that was developed for the purpose of the work in this thesis.

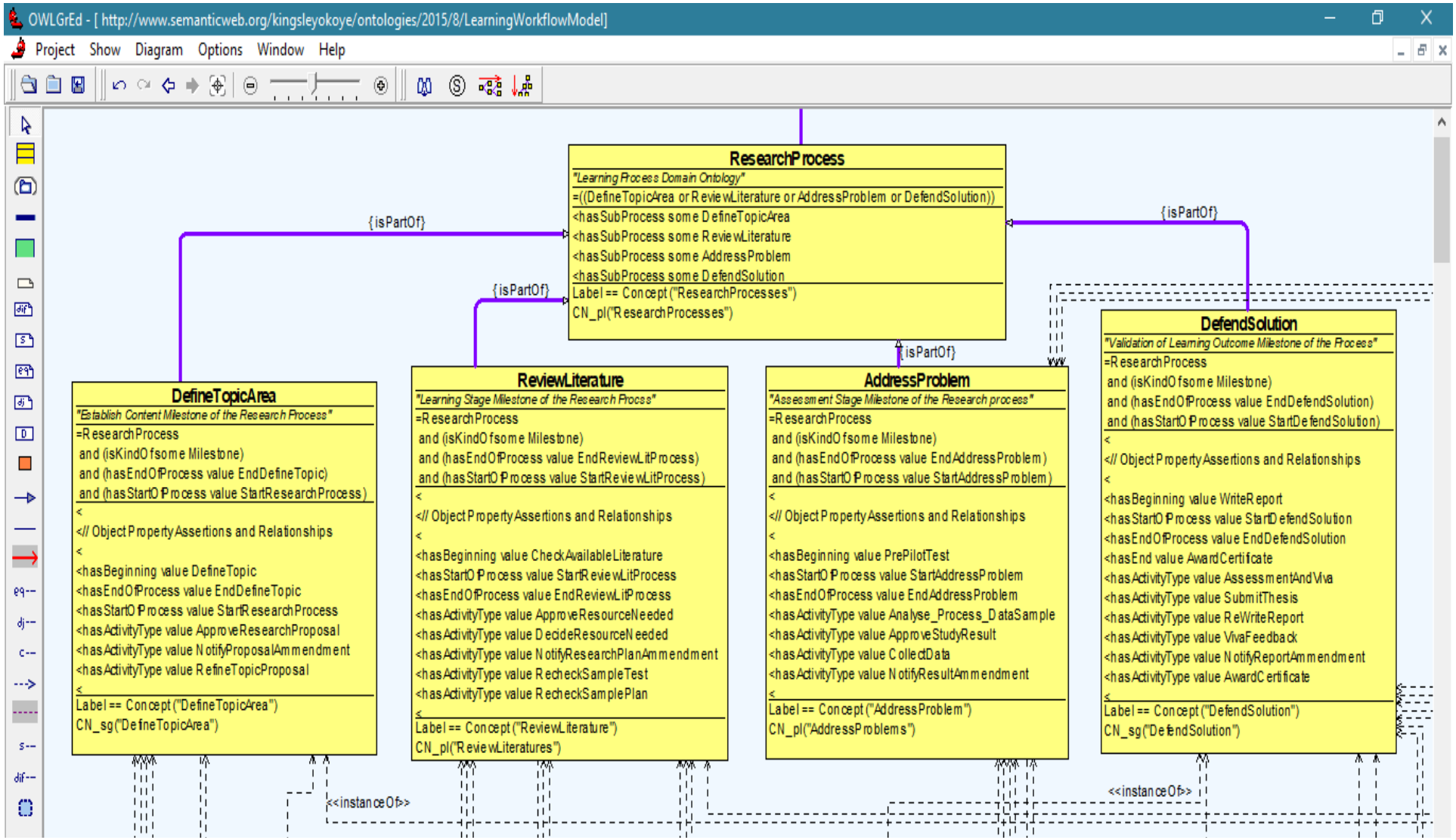


Figure 5.1 Research Process Domain with description of the Learning activity concepts and relationships

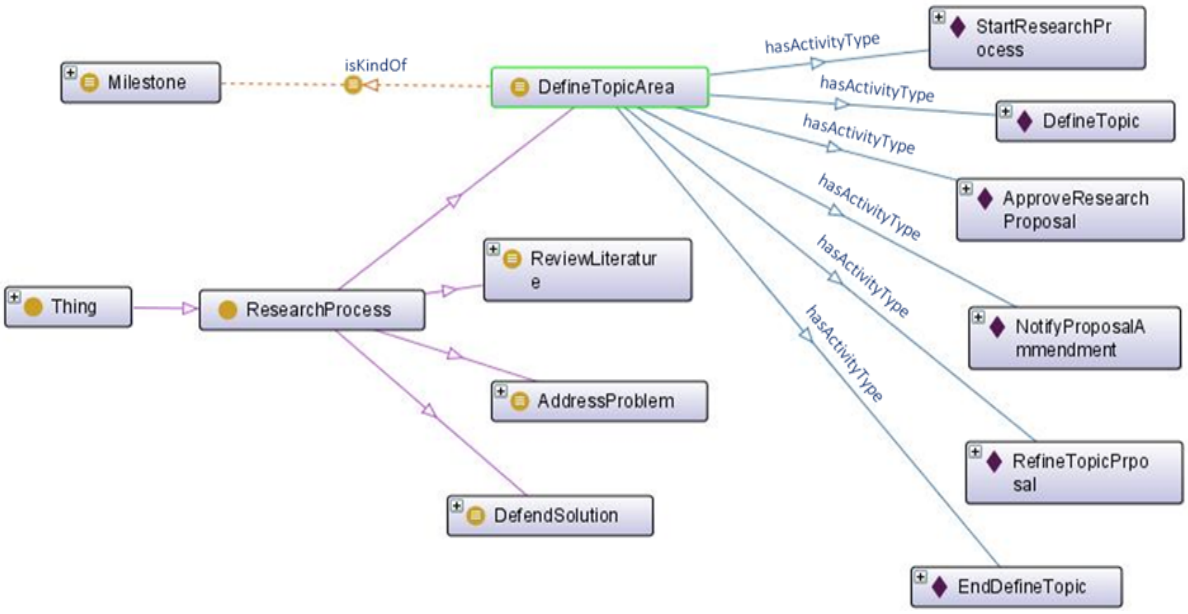


Figure 5.2 Ontology Graph and ActivityConcept mapping for the DefineTopicArea Milestone.

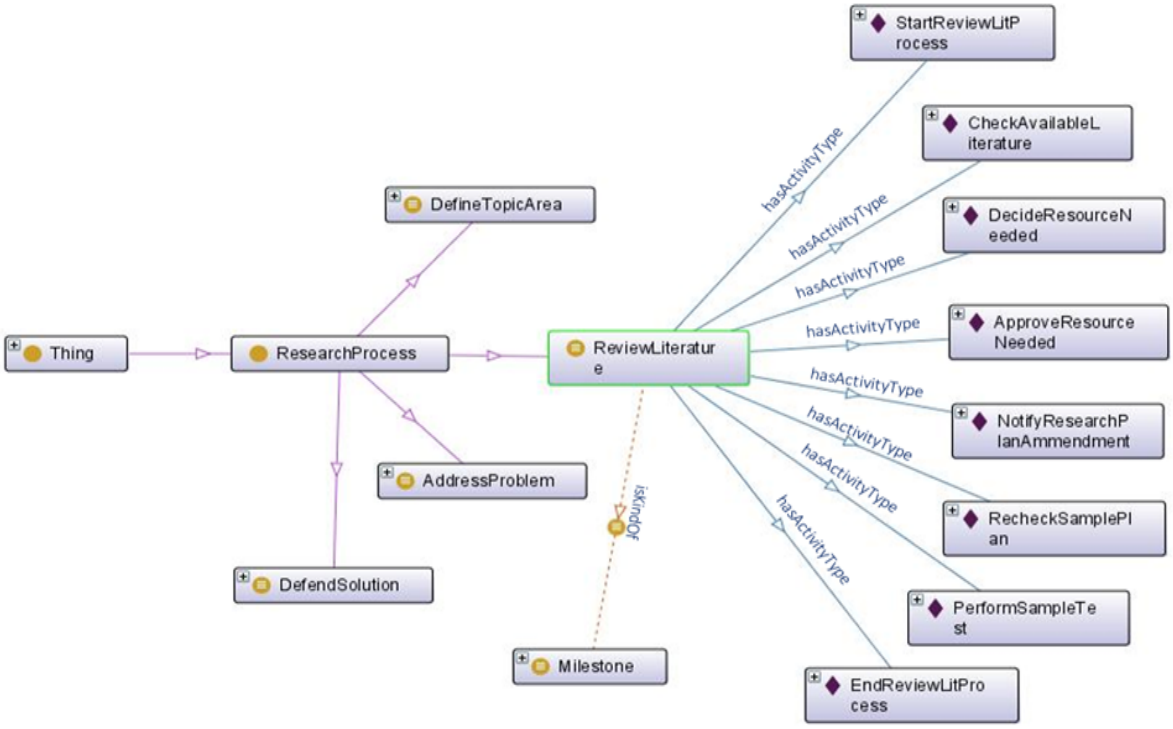


Figure 5.3 Ontology Graph and ActivityConcept mapping for the ReviewLiterature Milestone.

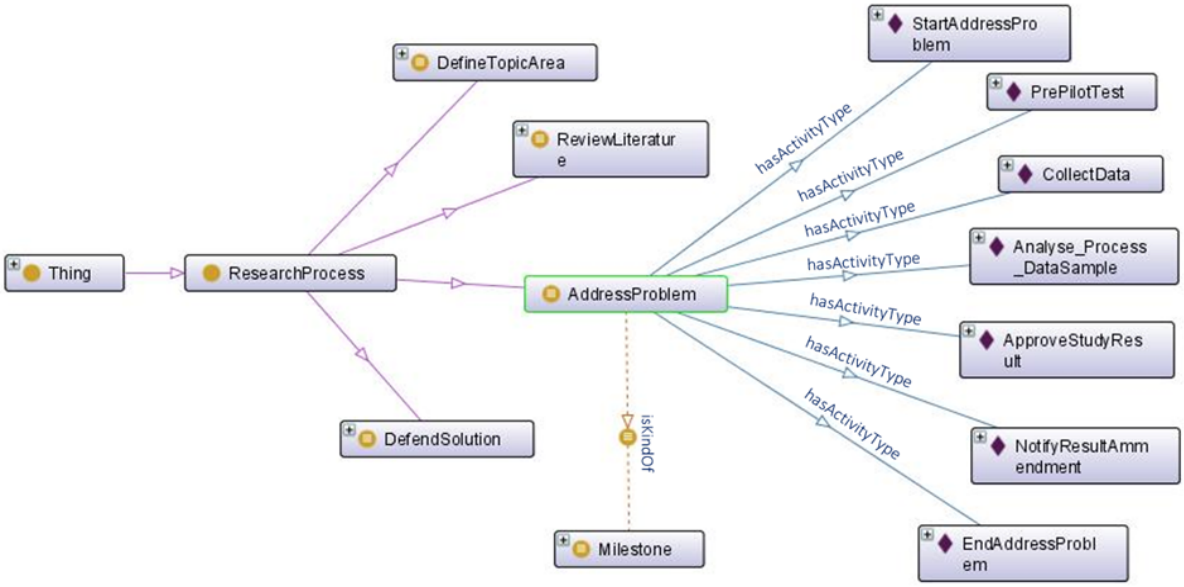


Figure 5.4 Ontology Graph and ActivityConcept mapping for the AddressProblem Milestone.

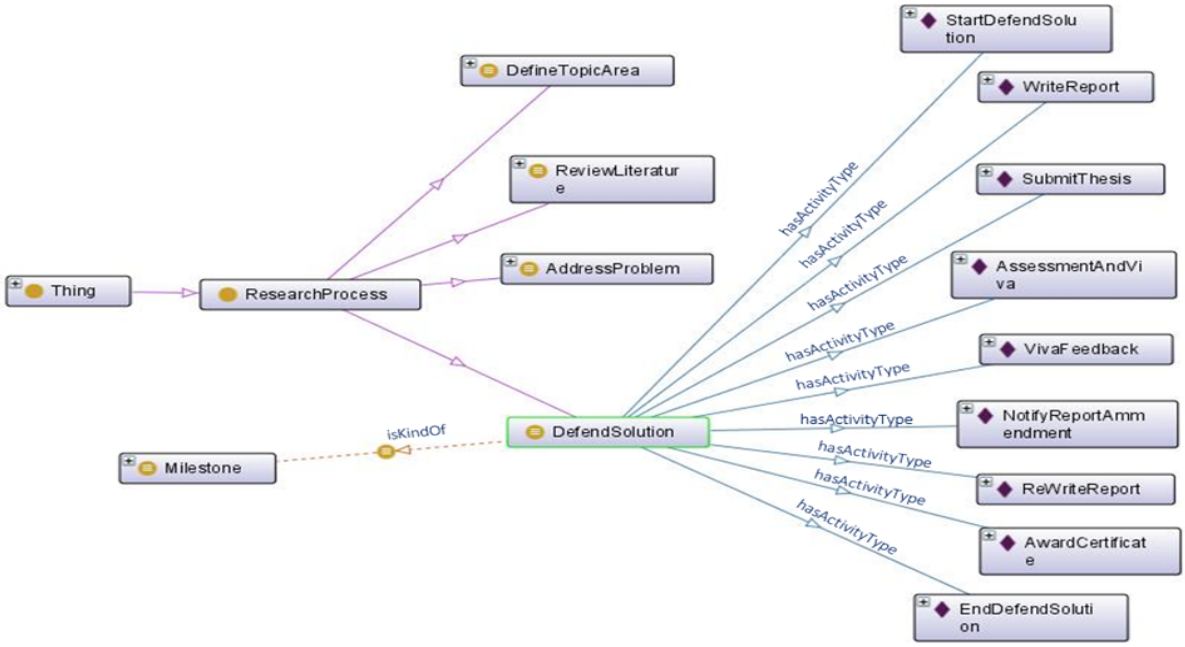


Figure 5.5 Ontology Graph and ActivityConcept mapping for the DefendSolution Milestone.

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 and classification of discoverable entities.

For instance, to address the real time learning questions the work have identified in section 5.1 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).

Moreover, as shown in Figure 5.6 - 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.

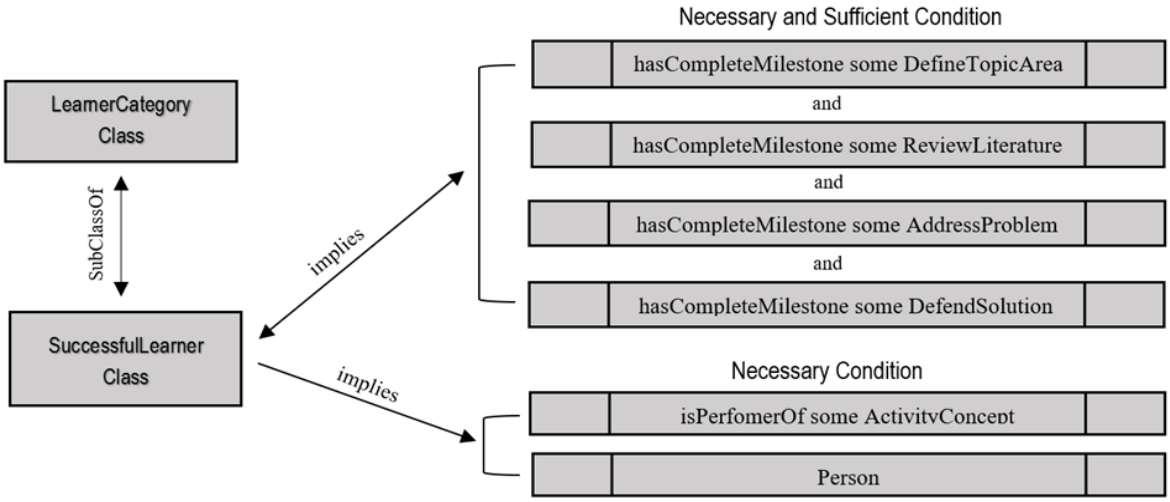


Figure 5.6 Attributes/Object Property Assertions for the SuccessfulLearner Class.

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.

As shown in Figure 5.7 - the *necessary condition* is: if something is a 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 e.g. DefineTopicArea, ReviewLiterature, AddressProblem but not all of the four classes.

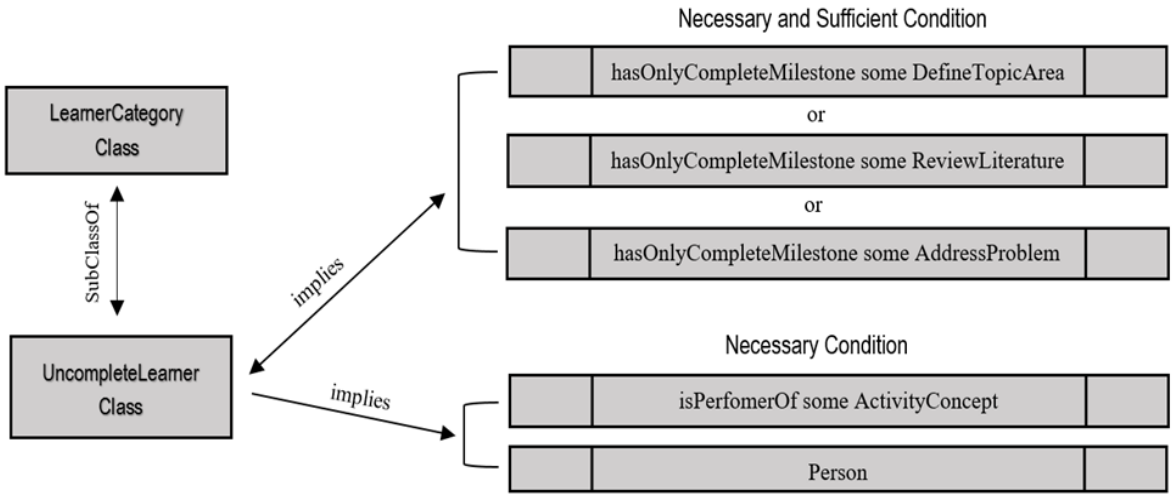


Figure 5.7 Attributes/Object Property Assertions for the UncompleteLearner Class

Ideally, we observe in Figures 5.1 to 5.7 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 specific class, for instance, the *successful* or *uncomplete* 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 identified in section 5.1.

In fact, the restrictions (i.e structured organisation) and the semantic labelling (annotation) serves as a good practice for representation of the learning process information by providing a formal way of determining the individual process instances within any kind of process knowledge base as shown in Figure 5.8 and 5.9.

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 5.8 including the OWL XML file syntax as follows:

SuccessfulLearner Class:

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

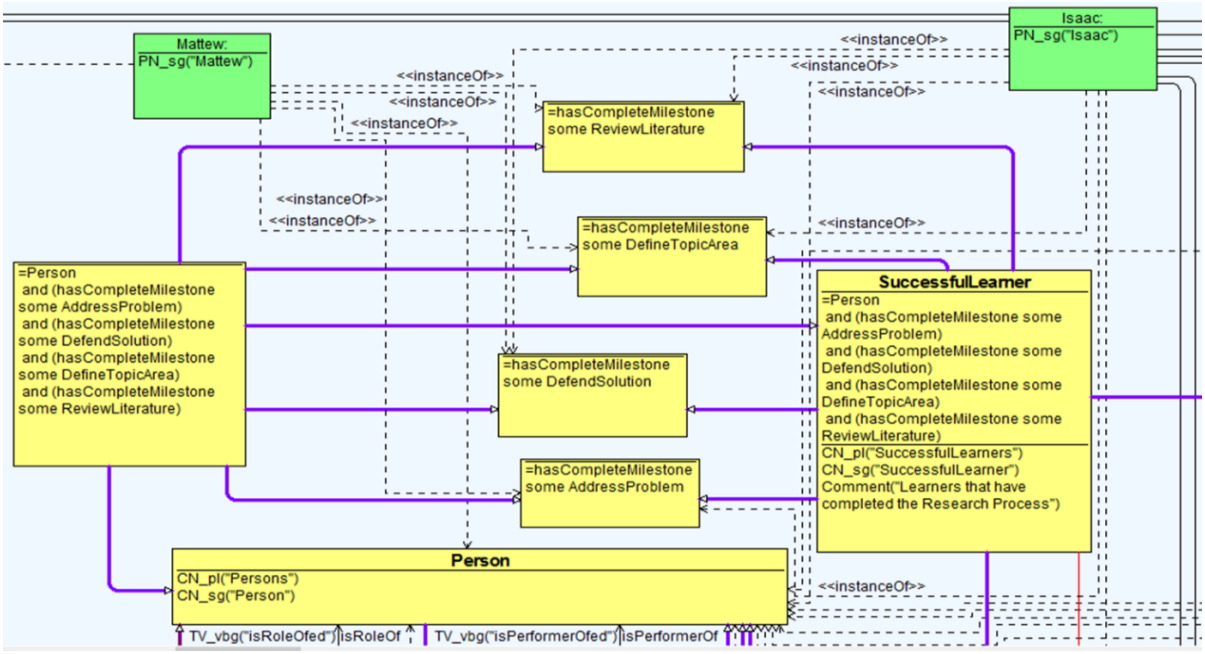


Figure 5.8 Concept assertions and the different formal relationships for the SuccessfulLearner Class

```
<EquivalentClasses>
        <Annotation>
            <AnnotationProperty
IRI="http://attempto.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>
```

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 5.9 including the OWL XML file syntax as follows:

Uncomplete Learner Class:

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

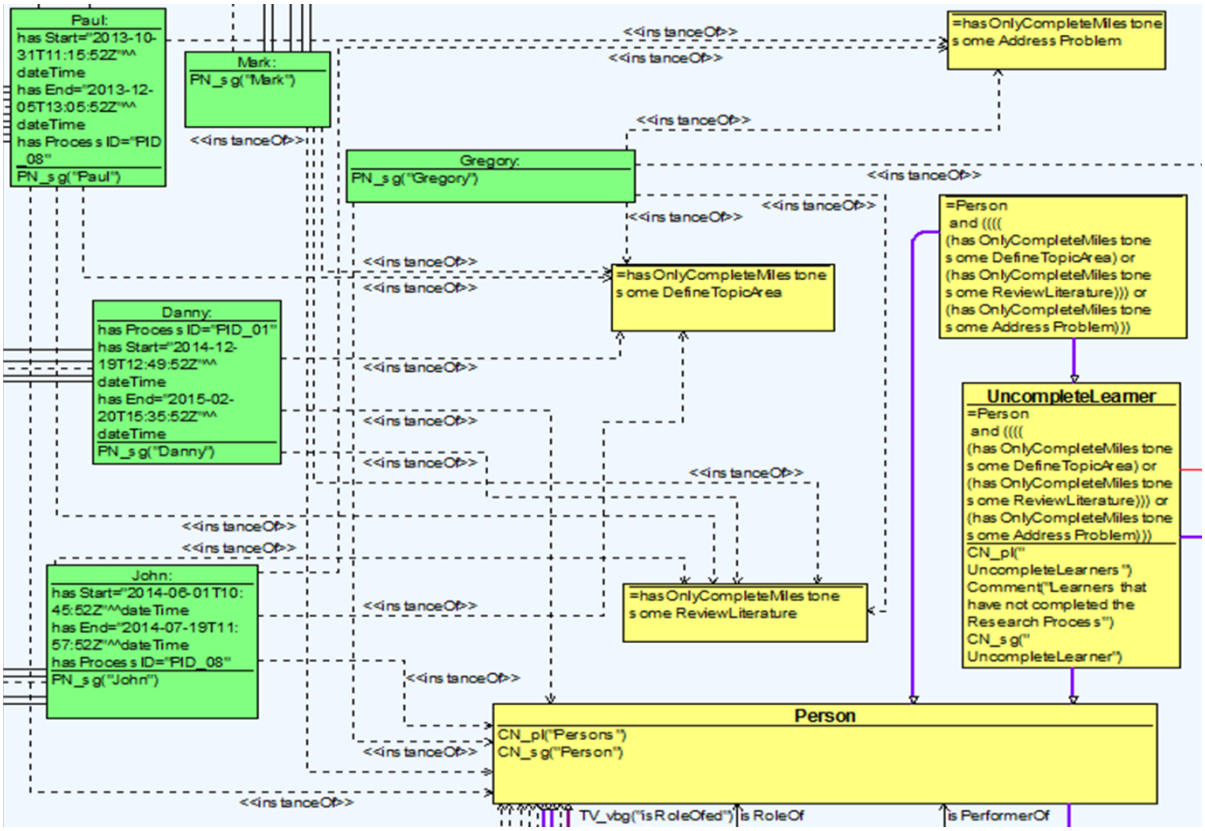


Figure 5.9 Concept assertions and the different formal relationships for the UncompleteLearner Class

```
<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 onlyHaveMilestones a
DefineTopicArea or that onlyHaveMilestones a ReviewLiterature is an
UncompleteLearner.</Literal>
        </Annotation>
```

```
    <Annotation>
        <AnnotationProperty IRI="http://purl.org/dc/elements/1.1/date"/>
        <Literal datatypeIRI="&xsd;string">2016-04-19 13:40:23</Literal>
    </Annotation>
    <Annotation>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UncompleteLearner</IRI>
    </Annotation>
    <Class IRI="#UncompleteLearner"/>
    <ObjectIntersectionOf>
        <Class IRI="#Person"/>
        <ObjectUnionOf>
            <ObjectUnionOf>
                <ObjectSomeValuesFrom>
                    <ObjectProperty IRI="#hasOnlyCompleteMilestone"/>
                    <Class IRI="#DefineTopicArea"/>
                </ObjectSomeValuesFrom>
                <ObjectSomeValuesFrom>
                    <ObjectProperty IRI="#hasOnlyCompleteMilestone"/>
                    <Class IRI="#ReviewLiterature"/>
                </ObjectSomeValuesFrom>
            </ObjectUnionOf>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="#hasOnlyCompleteMilestone"/>
                <Class IRI="#AddressProblem"/>
            </ObjectSomeValuesFrom>
        </ObjectUnionOf>
    </ObjectIntersectionOf>
</EquivalentClasses>
```

5.3 Experimentations and Main Results

5.3.1 Fuzzy-BPMN Mining approach: Experimentations and Implementation

In this section of the thesis, the research shows how it practically apply the current tools that supports process mining by participating in the First Process Discovery Contest (Carmona, et al., 2016) organised by IEEE CIS Task Force on Process Mining (IEEE CIS Task Force on Process Mining, 2016; Van der Aalst, et al., 2012). Indeed, the group has introduced the contest to foster research within the area of process mining (in particular, process discovery), with the primary aim of promoting the research field and its main applications in real world settings. According to (Carmona, et al., 2016) the contest is dedicated to the assessment of tools and techniques that discover business process models from event logs. A number of event logs were provided by the group (Carmona, et al., 2016). The event logs are generated from business process models that show different behavioral characteristics. The main objective is to compare the efficiency of techniques that discovers process models capable of providing a proper balance between “overfitting” and “underfitting” models. In other words, a discovered model is seen as *overfitting* (the event log) if it is too restrictive by disallowing behaviour which is part of the underlying process. On the other hand, the model is considered

as *underfitting* (the reality) if it is not restrictive enough by allowing behaviour which is not part of the underlying process. 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 (Carmona, et al., 2016)

Furthermore, each of the test event logs precisely ((*test_log_april_1* to *test_log_april_10*) and (*test_log_may_1* to *test_log_may_10*)) which can be found in (Carmona, et al., 2016) represents part of the original model that were not revealed. Also, the *test logs* with complete total of 20 traces for each log are considered to consist of 10 traces which are replayable (*allowed*) and another 10 traces which are not replayable (*disallowed*) by the model. Clearly, the total number of traces for the test logs (i.e. *April log* and *May log*) is therefore:

10 test logs x 20 traces which equals to a total of = 200 Traces for each of the *April log* and *May log* respectively

The aim of the research participation is to carry out a classification task to determine the individual traces that makes up the two test event logs.

To start with, the work discovers 10 process models from the *training sets* (Carmona, et al., 2016) using the Fuzzy miner (G¨unther, 2009; Günther & Van der Aalst, 2007; Rozinat & Gunther, 2012) and then makes use of the Business Process Modelling Notations (BPMN) (Van der Aalst, 2016) to analyse and provide the replaying semantics for the process models. Further details about the 10 different process models that are discovered using the method is described in (Okoye, et al., 2016) and are provided in the Appendix A section of this thesis.

Accordingly, the work performs a classification task for the *test set* (Carmona, et al., 2016), to generate the various cases (traces) that makes up each of the process executions. The work summarises in this thesis how it generated the 20 individual traces for each of the test log including the sequence of the activity executions for each of the individual traces. Further details about the classification results are also provided in (Okoye, et al., 2016).

Moreover, the *data set* that has been provided for the Process Discovery contest (Carmona, et al., 2016) 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. We assume that each of the logs consist of data sets that are related to a single process which refers to a single process trace (*Case*) and can be related to some *Activity*. Equally, the given event logs contains two attributes *case_id* and *act_name* which precisely specify the requirements that allows for implementing the process discovery technique following the definition 4.1 in (Van der Aalst, 2011). For that reason, the work assumes the following standard:

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

Indeed, the standard definitions were necessary because for the employed approach the activities play an important role for the discovered model and thus corresponds to the individual cases within the discovered fuzzy models. As there are multiple *events* that refers to similar *Activities*, the work supported the filtering of the 200 individual traces that makes up the test event logs with a *classifier* (Van der Aalst, 2016).

Therefore, if we utilise the notation $\underline{e}$ to refer to event names available within the logs, then the classifier for any event in the given log will be:

$$e \in \mathcal{E}, \quad \text{where} \quad \underline{e} \quad \text{is the name of the event.}$$

Accordingly, since the events are solely recognised by the corresponding activities name (act_name), we formerly assume that:

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

If we apply the classification conversion of the event logs provided in (Carmona, et al., 2016), i.e., Simple Event Log which are explained in details in Definition 4.4 of (Van der Aalst, 2011) to obtain the Log.

Then, the described event logs definition: Let A be a set of act_name.

Implies that a single trace σ is a sequence of activities, i.e., $\sigma \in A^*$. Where a simple event Log L is a multiset of traces over some set A.

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

Clearly, for the *training log* there are 1000 cases (traces) that defines the log. However, the research focus is to identify the sets of traces (i.e. 200 for *April* and 200 for *May* logs) that characterize the *test event log* for use in validating the discovered model (i.e. training log).

Perhaps, If we Let $L \subseteq C$ be the event logs for the test log, and assuming that the classifier $e \in \mathcal{E}$, is applied to the sets of activities, then from definition (4.5) in (Van der Aalst, 2011)

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

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

Likewise, all the Cases (traces) in the *test log* are converted into sequences of the activities (act_name) using the classifier. Thus

- A Case $c \in L$, is an identifier from the case C.
- $\hat{c} = \#trace(c) = \langle e1, e2, \dots, en \rangle \in \mathcal{E}*$ is the sequence of events executed for c
- $(\hat{c}) = \langle e1, e2, \dots, en \rangle$ maps these events onto the activity names(act_name) using the classifier.

From the described classification method ($\underline{e} = \#act\_name(e)$), we obtain from the logs containing the sequence sets of 200 traces for the test event log (*test_log_april_1* to *test_log_april_10*), i.e., 20 *traces* for each log as follows:

$\underline{L}$ (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 details of all the classified traces for the complete test logs can be found in (Okoye, et al., 2016).

Furthermore, having classified the test event logs using the method described above. The test logs were then imported into Disco (Rozinat & Gunther, 2012) to see in details how this processes has been performed (i.e. process mappings), and more importantly to determine the individual Cases (traces) that makes up the process so as to check if in reality it conforms (i.e. corresponds) with the classified traces. In turn, the results are fuzzy models that represents the various *cases* and *activities sequence* mappings as shown in example Figure 5.10.

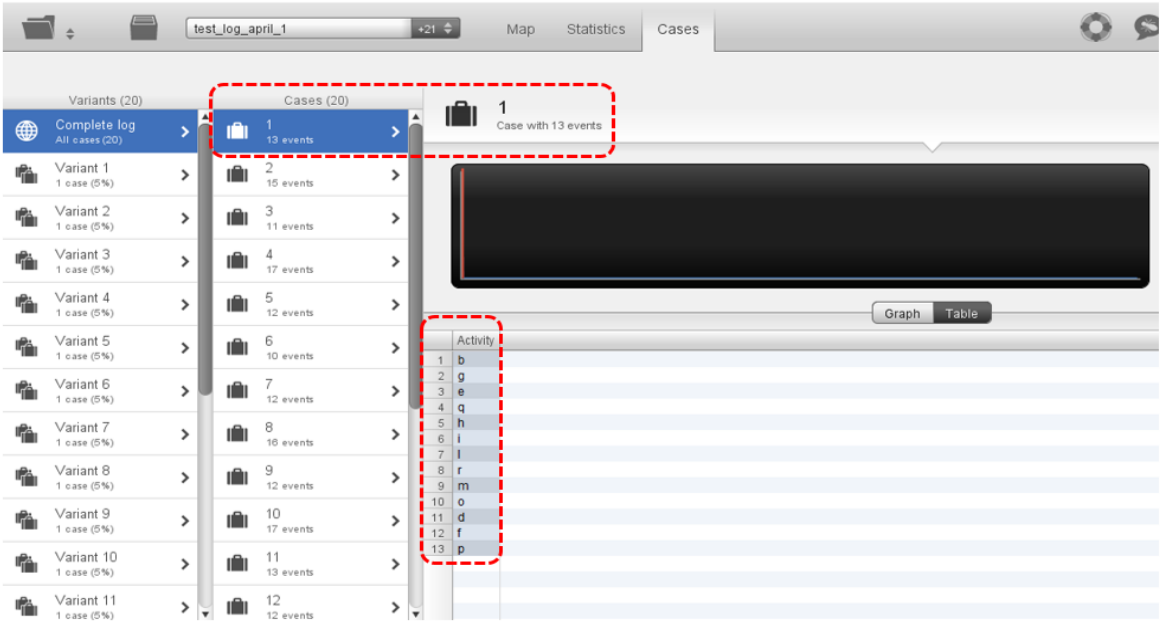


Figure 5.10 Case view for the test_log_april_1 showing the 20 cases with an example of case 1 (trace) with 13 events and table showing set of Activities for trace 1.

Indeed, the method described above is what the research used to check the results of the classification tasks to see if they conform to the given event logs. For example the activities for the first *case* 1 highlighted in Figure 5.10 indeed corresponds to the first trace discovered by the classifier, i.e.

$\underline{L}$ (test_log_april_1) =

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

In view of the trace classifications, the Fuzzy-BPMN approach determines the *fitness* (replaying semantics) of the individual traces for the test event logs by cross-validating the classified traces against the discovered process models from the *training logs*.

To achieve the set objective, it was necessary to construct BPMN models with notational elements as shown in Figure 5.11 capable of describing the nesting of individual activities (also referred to as *task*) by using the event-based split and join gateways - i.e. *AND, XOR,* and *OR* etc. 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, the work utilized the BPMN event-based gateways to replay the traces fitness alongside the derived model from the training event log, and in so doing, identify which traces that are fitting or not fitting the original model.

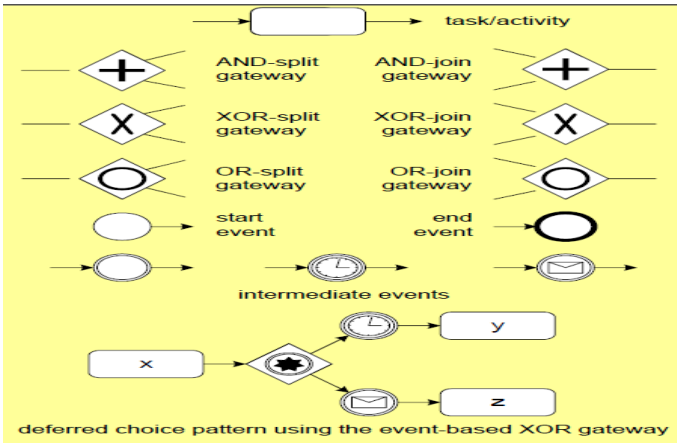


Figure 5.11 BPMN Gateway with Notational elements (Van der Aalst, 2011)

Indeed, an event in BPMN model could be compared to a place within a Petri-net, and just like Petri nets, are token based semantics which can be used to replay a particular trace within the discovered process model (Van der Aalst, 2011; Van der Aalst, 2016). Accordingly, the work makes use of the *Convert Petri net to BPMN* plugin in ProM (Verbeek, et al., 2011) to discover the BPMN models for the training logs. Figure 5.12 is an example of the discovered BPMN Diagram for the *training_log_1*. Further details about the other 10 different BPMN models that was discovered using the approach can be found in (Okoye, et al., 2016) and also included in the Appendix A.2 section of this thesis.

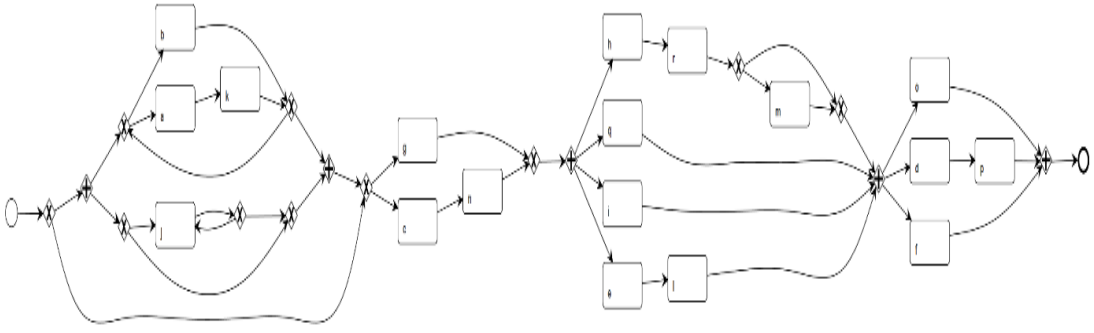


Figure 5.12 Example of BPMN model discovered for the training_log_1

Consequently, in Table 5.1 the work presents the classification results of the Fuzzy-BPMN miner approach for the *test event logs* cross-validated against the corresponding *training set* (model): where each individual cell indicates if the discovered model classifies the corresponding trace as fitting (allowed) or not fitting (disallowed). In other words, 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.

Table 5.1 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.

As shown in Table 5.1 the following performance metrics (Van der Aalst, 2011; Van der Aalst, 2016) 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 the classification results and models.

The IEEE CIS Task Force on Process Mining contest committee published on the website (Carmona, et al., 2016) (a) 10 test logs, each of which contains 20 traces that were used to score the submissions, and (b) 10 reference process models in BPMN generated from the original event logs which were not previously revealed. 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.

Presently, the only other contest related to process mining is the annual Business Process Intelligence Challenge (BPIC) (van Dongen, et al., 2016) which makes use of real life datasets, but without an objective evaluation criteria. The BPIC contest focuses more on the observed values of the process mining and analysis techniques, and as such does not limit its submissions to the process discovery methods (for instance, the contest also looks at some performance analysis techniques, conformance checking etc.). However, the submissions are also being assessed by a panel of judges. On the other hand, the BPM Process Discovery Contest (Carmona, et al., 2016) is quite different from the BPIC because it focuses more on process discovery techniques. In essence, datasets which are synthetic in nature are used to have an objectified "proper" answer to process mining problems. Thus, the process discovery is turned into a classification task with a training set and a test set, where a discovered process model needs to decide whether the classified 'traces' are fitting or not.

5.3.2 Semantic-Fuzzy Mining: Experimentations Outcomes and Results Analysis.

In this section, the work makes use of the event logs in (Carmona, et al., 2016) to describe how the work expounds the amalgamation of the two process mining techniques namely: Fuzzy miner and Business Process Modelling Notation (BPMN) approach in order to weigh up the performance of the proposed Semantic-based Fuzzy miner being able to perform a more accurate classification of the individual traces within the process base. This includes the capability to integrate ontological concepts and perform semantic reasoning capable of discovering worthwhile models given the datasets (with *training set* and a *test set*) for the cross-validation experiments. Henceforth, the sematic-based fuzzy mining and analysis allows the meaning of the process elements to be enhanced through the use of property characteristics and classification of discoverable entities. The method is introduced in order to generate inference knowledge that are used to determine useful patterns (traces) in an easy way, and predict accurately to a greater degree future outcomes. Indeed, this form of conceptualisation allows for analysis of the process elements at a much more conceptual level.

Perhaps, as explained earlier in the proposed algorithms in section 4.3, ontology is a method that trails to connect such set(s) of discoverable entities with either another class, or with a fixed literal and can also describe 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. 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 Figure 5.13 and 5.14 the work makes use of the "*hasTraceFitness*" object property to reference the class for the test logs that has a "*TrueTrace_Classification_(TP)*" or "*FalseTrace_Classification_(TN)*"

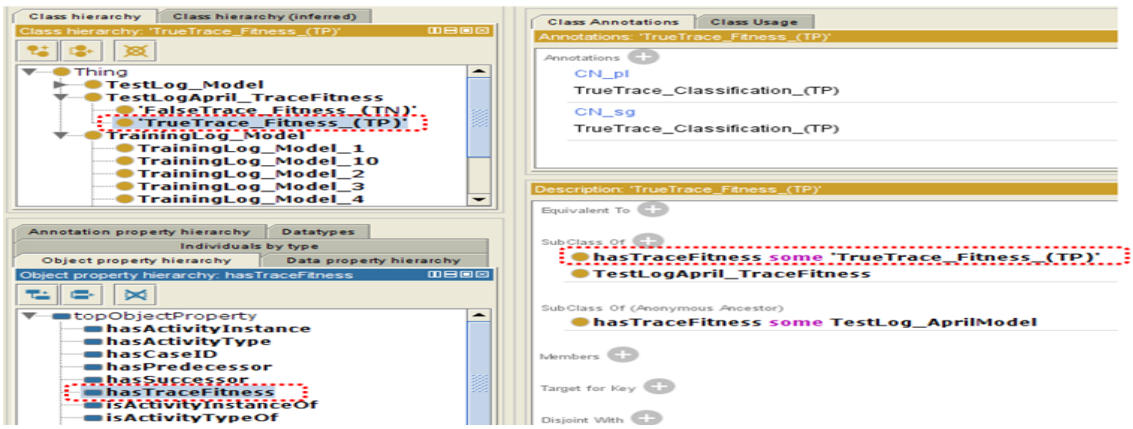


Figure 5.13 Object Property Assertion (annotation) for the True trace classifications.

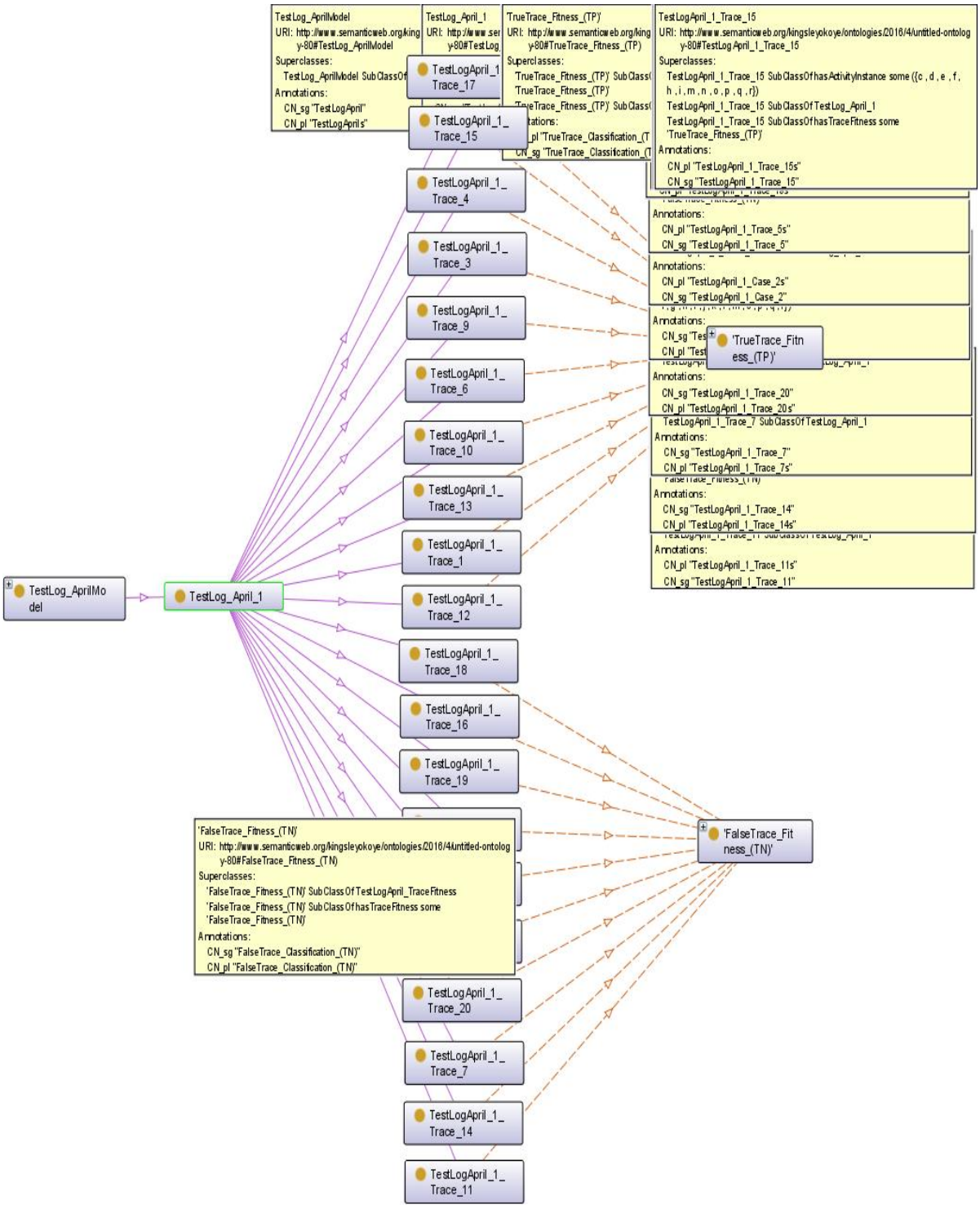


Figure 5.14 Example of OntoGraph for the TestLog_April_1 class with description of some of the semantic annotations.

More so, as explained in section 4.3 if we Let A be the set of all process executions or actions that can be performed within the semantic model. A process action $a \in A$ is characterized by a set of input parameters $In a \in P$, which is required for the execution of a and a set of output parameters $Out a \subseteq P$, which is produced by a after execution. For instance, the work executes

the DL (Baader, et al., 2003) queries below as a set of input parameters to output the set of traces for the example "TestLog_Apri_1" within the model that has 'TrueTrace_Fitness_(TP)' and 'FalseTrace_Fitness_(TN)' respectively. Thus:

"`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 Figure 5.15 and 5.16 respectively.

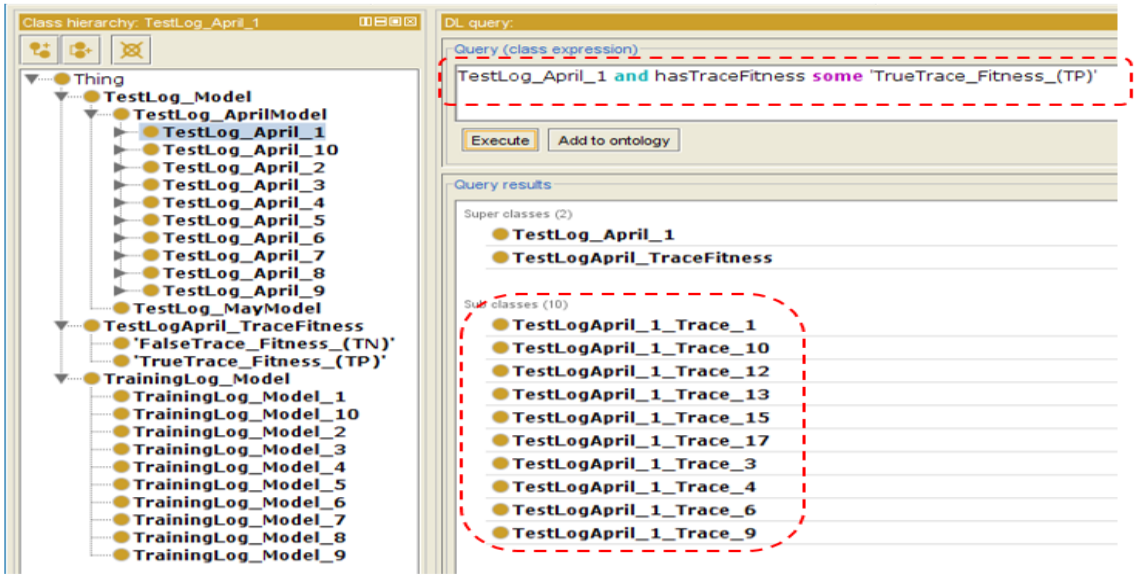


Figure 5.15 Example of the TrueTrace_Fitness_(TP) classification for the TestLog_April_1 with the correctly classified traces.

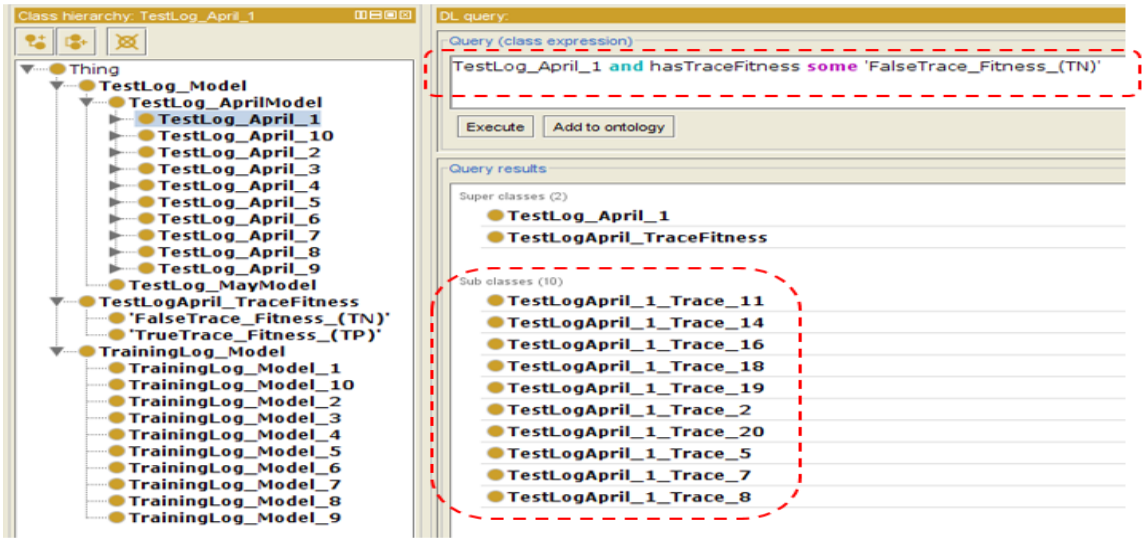


Figure 5.16 Example of the FalseTrace_Fitness_(TN) classification for the TestLog_April_1 with the correctly classified traces.

Accordingly, for the application phase of the approach in this thesis, the work implements 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 uses 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) (Clark & Parsia, et al., 2017) to extract and load the inferred concepts ascertained within the ontology (i.e. the semantic model). The purpose for designing the application 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 5.17 shows the application interface the work has developed for querying and retrieving the sets of data within the defined ontology model and the concepts as implemented in Java using the OWL API in Figure 5.18 as follows:

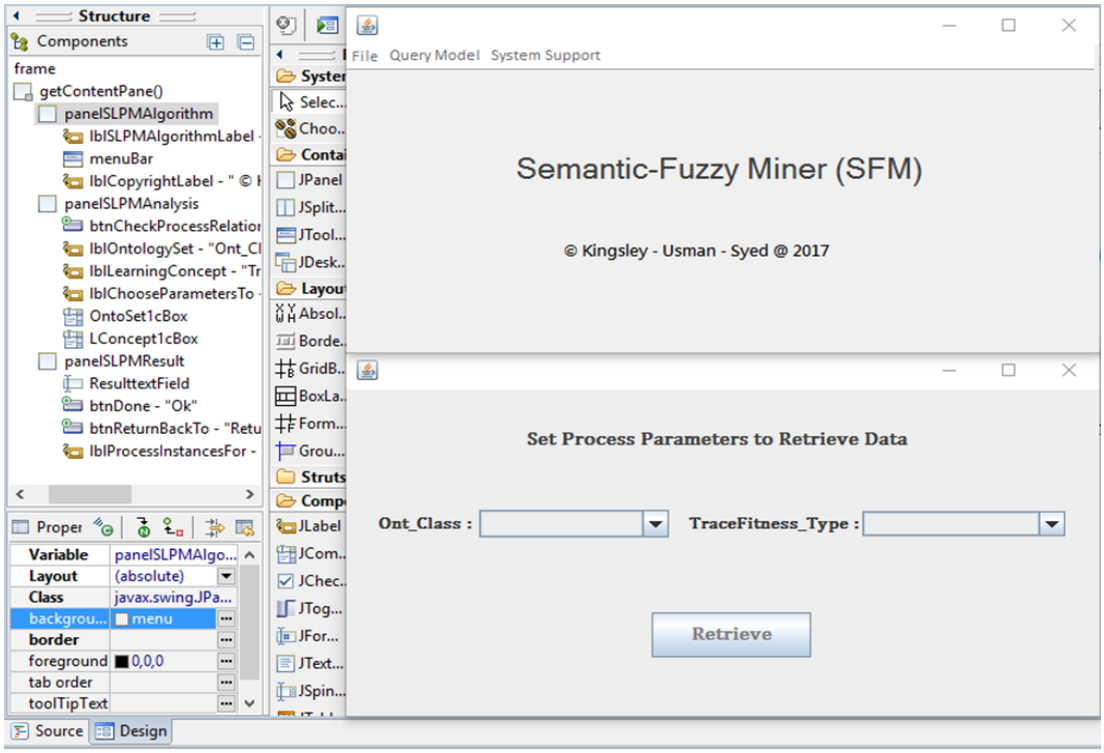


Figure 5.17 Application Interface for the semantic-fuzzy miner (SFM) in java runtime environment.

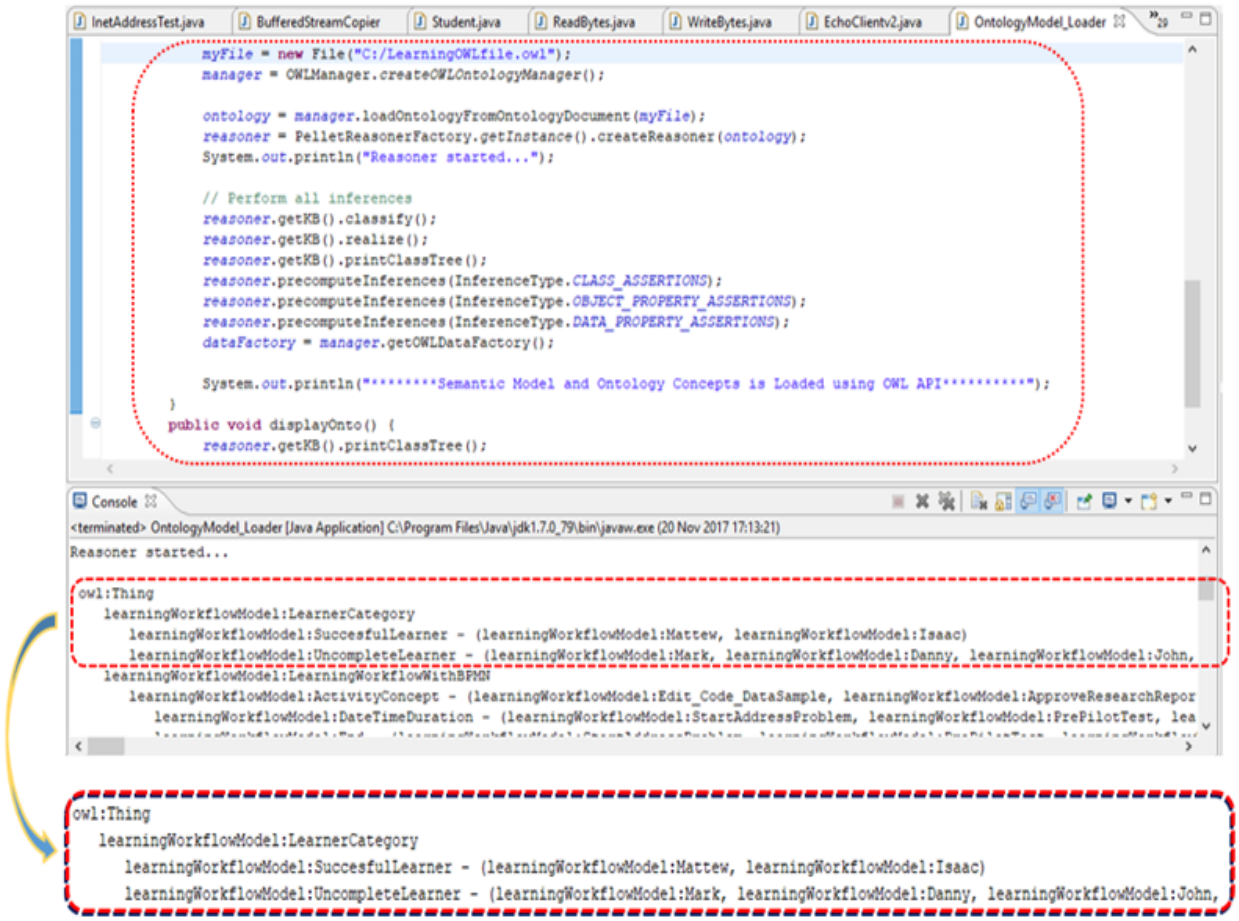


Figure 5. 18 Inferred Learning Concepts in the Java Application using OWL API

Indeed, the semantic fuzzy mining approach and its main application (e.g. as shown in Figure 5.17 and 5.18) references a number of different OWL ontologies (for instance, the training model ontology, test set ontology, traceFitness Classification ontology etc.) which were created for the experiment. For each ontology, all concepts in their turn were considered by the reasoner and are checked for consistency by referencing the process parameters. Based on behavioural characteristics of the provided datasets in (Carmona, et al., 2016), a cross validation method was adopted in order to overcome the variability in the composition of the training sets and test sets. The traces were computed and recorded according to the reasoner response, and the classifier was tested on the resulting individuals by assessing its performance with respect to correctly classified traces. For each result of the classification process, the replayable (true positives) and non-replayable (true negatives) traces were learned. Thus, 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 5.2.

Table 5. 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.

From the Table 5.2, 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, thus equal to 0. Indeed, 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. 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.

5.4 Summary

In this chapter of the thesis, the work has shown how the proposed SPMaAF framework and the resulting semantic fuzzy mining 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 in the event logs and models. This includes the integration of the main components and tools used for implementing the semantic process mining approach and proposed algorithms as explained in chapter 4 of this thesis. The process is illustrated through the use case study of the learning process and data about the real time business process used in this thesis.

The following Table 5.3 shows the thematic summary of all the main implementation components and tools as utilized in this chapter as follows:

	Business Process (IEEE CIS Task Force on Process Mining)	***Learning Process (Research Process Domain)***	**Main Tools**
Events Log	**X**	**X**	Learning Activity Log, Training Log, Test Log
Process Models	**X**	**X**	Fuzzy Models, BPMN Models, OntoGraph
Semantic Annotation		**X**	Process Description Languages, SWRL Rules
OWL Ontology		**X**	Protégé Editor, OWLGriD
Reasoner		**X**	Pellet
Fuzzy-BPMN Notation	**X**		PROM, Disco
Semantic Model and Conceptual Analysis		**X**	DL Queries, OWL API

Table 5. 3 Main tools and implementation components of the proposed semantic-based approach and case studies in the thesis.

Practically, such method of data classification and conceptual model analysis as summarised in Table 5.3 can be applied to any given process domain provided there is available event data log from the domain in view. Moreover the method could 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.

Significantly, it is shown that the chapter that the classification performance of the semantic-based fuzzy mining approach 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 using the case studies of the Business Process and Learning Process domain data. Indeed, an increase in the predictive accuracy was achieved by means of the semantic-based annotations and conceptual analysis. Besides, the technique can also be exploited in any form of data analysis procedures for prediction or suggestion of missing information (metadata) about the different process elements especially when completing large ontology-based systems. Additionally, the new knowledge and semantic assertions could be used by the process owners, process analysts or IT experts to address and answer real time questions about their processes in view.

In the next chapter, the work reveals how it measures in a qualitative and quantitative manner, the outcomes and impact of the implemented approach in this thesis compared to other benchmark algorithms and techniques used for process mining and analysis.

Chapter 6. Evaluation of Research Outcome and Results

This chapter of the thesis looks at the extent to which the Semantic Fuzzy mining approach permits the conceptual analysis of the events logs and discovered models. Qualitatively, the chapter looks at the case study of the *Research Learning Process domain* in order to determine how the proposed method in this thesis supports the discovery, monitoring and enhancement of real-time domain processes through further semantic analysis of the discovered models. In addition, the chapter quantitatively assess the level of accuracy of the classification results to predict behaviours of unobserved instances within the process knowledge-base by determining which traces are fitting or not fitting the discovered model using the *training set* and *test log* from the IEEE CIS Task Force on Process Mining for the cross-validation. 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.

6.1 Qualitative Evaluation of the Semantic Fuzzy mining Approach and Outcomes

Evidence from the research design framework, algorithms and experimentations shows that the semantic-based approach sparks methods that highly influence and support:

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

(ii) provision of real time semantic knowledge and understanding about domain processes (e.g. the case study of learning process used in this thesis) which are useful towards the development of process mining algorithms that are more intelligent with high level of effective conceptual reasoning capabilities.

In the experimentations, the research observes that ontologies help in harmonizing the various process elements that are found within the process models and datasets. In addition, the semantic annotations and reasoning helps to extract and add useful conceptual knowledge to the mining results.

Apparently, the work qualitatively use the case study of the learning process and addresses the series of typical real-time learning question as explained in section 5.1 of this thesis to show in details how the semantic-based approach is implemented and relevant in the context of process mining and abstract analysis. Therefore, the main tools and components realised as a result of implementing the semantic-based process mining approach particularly the SPMaAF framework and semantically motivated algorithms in chapter 4 is summarised as follows:

- *Event Logs* – used to show how process mining can be applied to improve the informative value of learning process data.
- *Process Model* – which describes 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 – which 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 process elements.
- Ontology – which describes how to make use of ontologies with effective semantic reasoning to lift process mining analysis from the syntactic level to a much more conceptual level.
- Semantic Learning Process Mining Algorithm – which reveals how references to ontologies and effective raising of process analysis from syntactic to semantic level enables real time viewpoints on the learning process model; 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.

In principle, the work utilized the case study of the learning process and use case scenario of the *successful* vs *uncomplete* learners to pilot the structure of event logs and process models in order to describe various semantic viewpoints (i.e metadata) in relation to how the process has been previously performed, as well as help in discovering the actual process workflows within the 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 models and readily available event logs through explicit specification of conceptualisation to identify appropriate domain semantics and/or relationships among the process elements.

Clearly, with use case example of the *successful* vs *uncomplete* learners, the thesis focus were to identify useful informations that describes the represented behaviours/patterns within the deployed model, and then respond by making decisions based on the process descriptions and semantic reasoning capabilities. The method is all aimed at improving the process analysis and system performance. Besides, the integration of the different ontologies, conceptual reference models, and a semantic reasoner enables the definition of a more universal analysis question, and then trails to find answers for those questions in an automated, thus, computerized manner.

Even more, due to the fact that the analysis are carried out at a more abstraction level (i.e. conceptualization). For instance, as shown in Figure 5.1 to 5.11, the results can be easily understood (i.e. closer to human comprehension) and the process of adding new concepts in the ontology or better still changes or modifications to the attributes (i.e. labels or tags) do not necessarily entails or requires updating the analysis questions or queries. For example, the process to determine the instances (learners) that have successfully completed the research process, one could easily include more activity concepts (or attributes) without requiring updating the actual question. The question remains the same and applicable to the class of individual that fulfils the universal or existential restrictions by way of the object property assertions and descriptions. Indeed, such characteristics proves to bring a much added tractability and flexibility to the entire process and the analysis of the derived process models.

From all evidence, such semantic-based approach proposed in this thesis is a significant contribution to the state of the art, where many existing process mining techniques requires some form of reconstruction to bring the process analysis to a much more 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 conceptualization has not been previously applied within the area of learning process domain.

In turn, the series of experimentations in this thesis proves that a system with formally encoded semantic labelling, ontology and semantic reasoning capabilities as signified in the design framework, algorithms, and semantic-based planning and 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 6.1 the research have carefully analysed the influence of the proposed semantic fuzzy mining approach compared to other existing benchmark algorithm for semantic process mining. Noticeably, as shown in the proposed approach and the resulting analysis in Table 6.1 - the use of ontologies, semantic reasoning/assertions, and references to labels within the event logs and process models - makes it possible to define a more easy and yet accurate way to analyse and automatically find answers to the real-time questions about the process elements and the relationships they share between themselves, as described earlier in Figure 5.13 to 5.18 in this thesis.

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) (deMedeiros, et al., 2008) currently in literature as presented in Table 6.1.

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

Table 6.1 The Semantic-Fuzzy miner and its application properties evaluated against existing benchmark algorithm.

Firstly, the semantic fuzzy mining approach based on these critical properties proves to be more robust and accurate than the traditional mining techniques. This is due to the fact that the approach also takes the semantics perspectives of event logs and process models into account. Moreover, as opposed to the existing semantic LTL checker which only considers and takes event logs concept as input to parameters of a Linear Temporal Logic (LTL) formula to analyse process, the semantic fuzzy mining approach also takes the process models as input. Besides, because those models are automatically created from the actual event logs of the process domains (e.g. in this thesis – events log about the research learning process domain), 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 the proposed Semantic-Fuzzy miner is the fact that ontologies are defined in Web Service Modelling Language (WSML) (de Bruijn, et al., 2006) format with the semantic LTL checker, whereas in the this work ontologies are defined using OWL (W3C, 2012; Horrocks, et al., 2007) and SWRL (Horrocks, et al., 2004) 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 (DL) to the semantic web. Moreover, it's the state of the art *logical layer* upon which semantic architectures are currently built in literature (Lisi, 2008). In fact, the OWL ontologies (for instance, as described in section 4.6 and implemented in section 5.2 and 5.3.2 of this thesis) allows one to specify far more about the *properties* and *classes* which are defined within the 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 thesis is developed not just for representing information in formats that can be easily understood by humans, but also for building applications that trails to inclusively process the informations that they contain or supports. In other words, supports *machine-understandable* systems rather than just *machine-readable* systems.

Thirdly, from a *reasoning* or *classifier* point of view, whilst the semantic LTL checker makes use of the WSML2Reasoner (Bishop, et al., 1999; de Bruijn, et al., 2006) to perform a more complex inferences that are beyond subsumption reasoning by only

benefiting from the inclusion of semantic annotations, the semantic fuzzy mining approach is integrated with Pellet reasoner (Sirin & Parsia, 2004) which typically in addition to semantic annotations has been proven to incorporate optimizations for nominals, conjunctive rules and query answering, and incremental reasoning capabilities that supports process descriptions and logic, i.e, class assertions and object/data property assertions, and are indeed shown to be very effective in reasoning particularly at a more conceptual level. For instance as described in section 4.7 of this thesis.

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.

6.2 Quantitative Evaluation and Analysis of the Semantic Fuzzy Mining Approach

In this section of the thesis, the work presents how it have quantitatively assess and validate the accuracy and performance of the classification results for the semantic-based approach.

Fore mostly, it's imperative to bear in mind that to quantitatively measure the quality of process mining algorithms or techniques, it is essential that we must first focus on the accuracy of the classification results (i.e. the outcomes of the classifier over the given data sets) rather than focusing on the *seen* (observed) process instances. Moreover, the quality of analysis of the classification result is perhaps useful to further predict good classification for *unseen* (unobserved) instances.

Henceforth, given a dataset that consists of *N* instances one can presumably identify that for each of the instances: what the actual class is, and what the predicted class is (often expressed as *confusion matrix*) (Van der Aalst, 2016; Van der Aalst, 2011).

According to the author in (Van der Aalst, 2011) the confusion matrix considers a given set of data with only two sets of classes; Positive (+) and Negative (-) values, and are measured using some performance formula for the classifiers as described in Table 6.2.

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 x *Precision* x *Recall*) / (*Precision* + *Recall*)

Table 6.2 Performance measures formula for the Classifiers

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 x *precision* x *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* $N = tp + fn + fp + tn$ equals the total number of instances within the dataset, *Then* base on the definitive expression, it becomes easy for one to determine the values of the class Positive (+) and Negative (-) classified by making use of the classifier. For instance, the total number of instances that are actually positive, i.e., $p = tp + fn$ can perhaps be realized.

On the other hand, the total number of instances which are actually negative, $n = tn + fp$ can also be determined.

In addition, if $p' = fp + tp$ refers to the total number of instances that are classified as positive by the classifier, then $n' = fn + tn$ likewise refers to the number of instances that are classified as negative by the classifier.

To this end, the formulas in Table 6.2 are construed. According to (Van der Aalst, 2011) 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 dataset is split into a *training set* and a *test set*.

Cross-validation (Van der Aalst, 2016) is one of the performance indicator method that could be utilized to evaluate process mining techniques. In such settings, the event log is split into a *training log* and a set of *test logs* in which the proposed mining approach tends to learn process models from a major part of the logs (i.e. the training log) as well as the individual cases that forms the event log (i.e. the test logs). Hence, the *training log* is utilized to discover the models, while the *test logs are* utilised to assess the fitness of the discovered models based on the unobserved traces.

The main idea of cross-validation method is to quantitatively assess the quality of the learned models in relation to the test logs that contains the *actual behaviours* (i.e fitting traces), as well as the quality of the learned models in relation to the test logs that contains *random behaviours* (i.e. artificially generated negative events). Superlatively, it is expected that the models scores way better for the logs that contains

the *actual behaviours* than the logs that contains the *random behaviours*. Therefore, the experimentations carried out in this thesis and analysed in this section measures to what extent the scoring of the discovered model when encoded with real semantics (formal domain knowledge) about the process elements (instances) helps to enhance the analysis of the process mining techniques from the syntactic levels to a much more conceptual level of analysis. 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.

Henceforth, 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 sets in (Carmona, et al., 2016). The available datasets stands for the same one the research used in this thesis, and for testing of the subject knowledge and practical application of the process mining in reality. Characteristics of those datasets are explained in the objectives of the process discovery contest (Carmona, et al., 2016) that focus on discovering worthwhile process models from a set of *training log* representing the 10 different real time business process executions, and sets of *test event logs* provided for evaluation of the employed process mining approach. Each of the test event logs represents part of the original model with complete total of 20 traces for each of the individual test logs, and are considered to have 10 traces which are indeed capable of being replayed (*allowed*) and 10 traces which perhaps cannot be replayed (*disallowed*) by the model. Therefore, a wide variety of problems is being represented in such settings.

In this thesis, the work have used the test event logs with complete total of 200 traces to validate the proposed 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 (Ghawi, 2016), DrFurby Classifier (Verbeek & Mannhardt, 2016), Heuristic Alpha+ Miner (Shteiner, et al., 2016) Fuzzy-BPMN miner (Okoye, et al., 2016) etc., that uses the same event logs in (Carmona, et al., 2016) to discover process models and provides replaying semantics for the individual traces that makes up the test log.

Accordingly, the work makes use of the standard Percent of Correct Classification (PCC) (Baati, et al., 2017) to assess the performance of the classifiers. Henceforth, the standard Percent of Correct Classification (Baati, et al., 2017) for the *test logs* is defined as follows:

Log_PCC = (*number of correctly classified traces*) / (*total number of traces*) x 100

For example, for the *training_model_7* as earlier shown in Table 5.1, the standard Percent of Correct Classification (PCC) for the *test log* for the initial results in section 5.3.1, i.e. the Fuzzy-BPMN mining approach is determined as follows:

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

On the other hand, the standard Percent of Correct Classification (PCC) for the *training_model_7* as shown in Table 5.2 for the Semantic-Fuzzy miner approach (section 5.3.2) is as follows:

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

Thus, using the logical formula i.e. standard Percent of Correct Classification (PCC) (Baati, et al., 2017) the research measures and analyse in Table 6.3 the sophistication of the other existing benchmark algorithms (Ghawi, 2016; Verbeek & Mannhardt, 2016; Shteiner, et al., 2016) and the initial result of the Fuzzy-BPMN miner (Okoye, et al., 2016) as described in section 5.3.1, to weigh up the proposed Semantic-Fuzzy mining approach and experimental results.

The outcome from the approach and the different benchmark techniques and classification results are as shown in Table 6.3.

	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
Sum of traces correctly classified	183	192	199	171	200

Table 6.3 Evaluation results of the Semantic-Fuzzy miner and other benchmark process mining techniques.

Clearly, from the evaluation results in Table 6.3, and the plots in the charts: Figure 6.1 to 6.3, 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 outcomes.

Additionally, the semantic-based approach has shown an error free performance indicator when measured using the classifier formulas, i.e. Error = (fp + fn)/N) where fp = 0 and fn = 0, thus, Error = (0 + 0) / 200 = 0.

Also, the semantic fuzzy mining approach has shown a high level of accuracy through the formula, Accuracy = (tp + tn)/N) where tp = 100 and tn = 100, thus, Accuracy = (100 + 100) / 200 = 1. Obviously, going by the F1 Score = 1, and the error-rate =0, the *Precision* and *Recall* of the Semantic-Fuzzy miner classifications are indeed efficient.

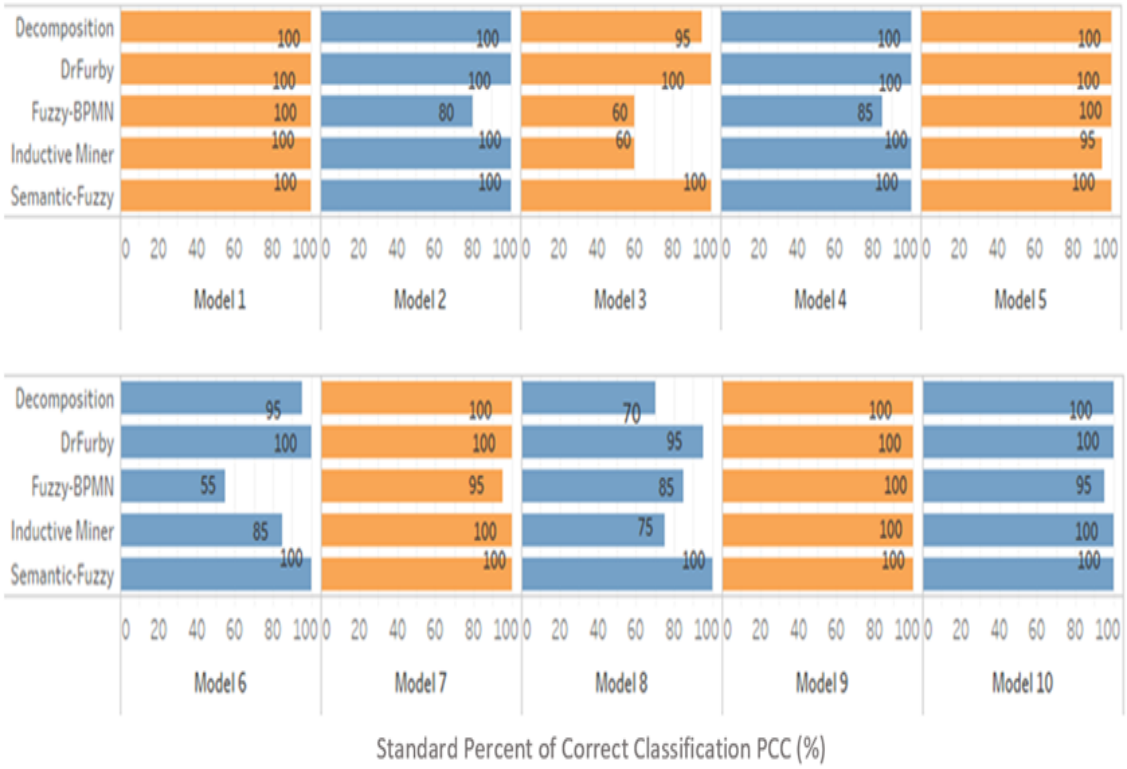


Figure 6.1 Chart showing the sum of correctly classified traces by the various algorithms for each Model 1 to 10 - using the standard Percent of Correct Classification PCC (%).

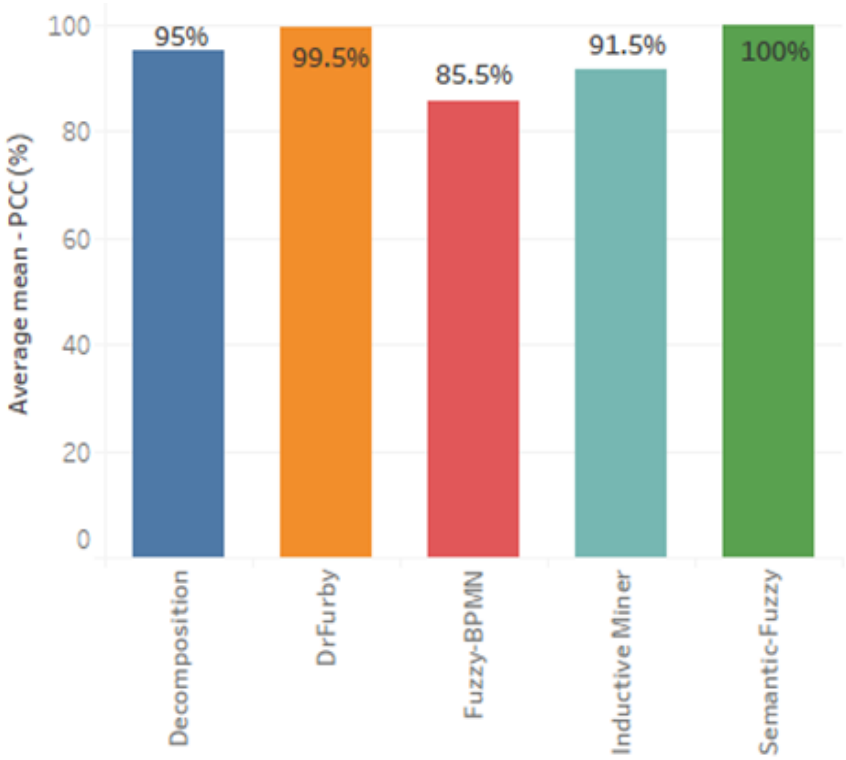


Figure 6.2 Sum of Average mean PCC (%) for the various Algorithms

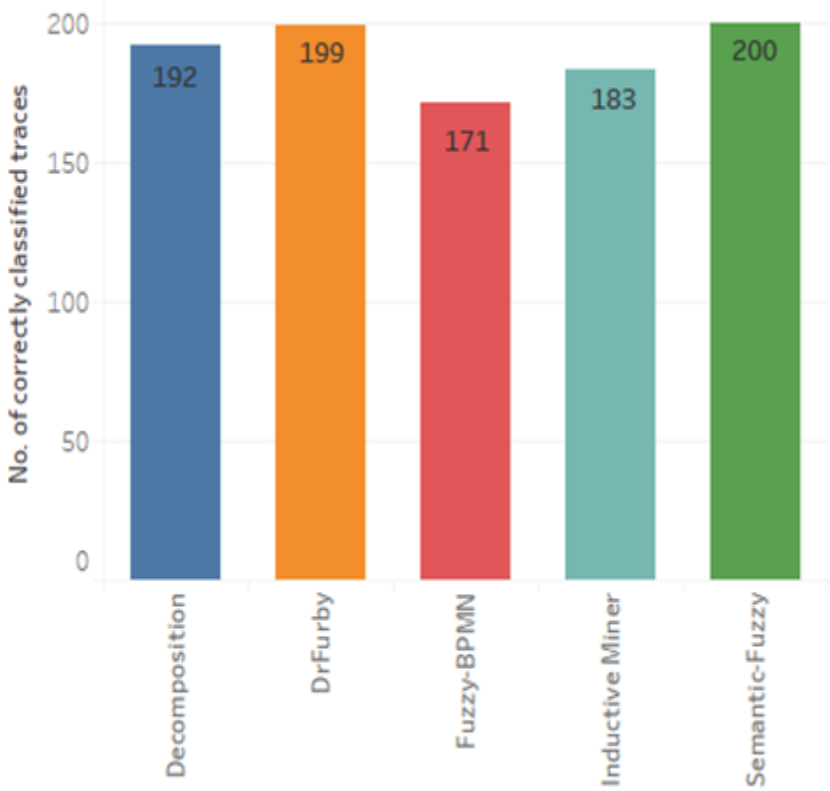


Figure 6.3 Total number of traces correctly classified by each algorithm

6.3 Evaluation Summary and Discussion

Indeed, the use of ontologies (*Ont* $\in$ *Onts*) and the relations (*R*) between the concepts (*COnts*) defined in the ontologies were beneficial to aggregate tasks and compute formally the structure of the process models including the several abstraction levels (Gruber, 1995; Okoye, et al., 2016; Okoye, et al., 2017). The main idea is that for any semantic-based process mining approach, these aspects of aggregating the task (d'Amato, et al., 2008) or computing the hierarchy of the process models (Lehmann & Hitzler, 2010) should not only be designed to be *machine-readable*, but must focus on providing a system which is also *machine-understandable.*

This implies that an ample design and development of such systems (such as the semantic fuzzy miner application in this thesis) that not just retrieve informations from the various process knowledge-bases they are used in, but also trails to process (analyse) the informations in those systems they support; has to be feed with process models which are already in a form that allows the computer to infer new facts (e.g.

through semantically annotated logs or the use of process description languages) in order to understand the underlying ontology. For example as presented in this thesis using the case study of the research process and datasets from the IEEE CIS Task Force on Process Mining.

Essentially, the purpose of the semantic annotation 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*. Moreover, the fuzzy logic (Zadeh, 1999; Zadeh, 1965) has since been introduced as an extension of the Boolean logic which allows a proposal to be in another state as true or false (Dammak, et al., 2014) by enabling the modelling of uncertainty and imprecision (as previously explained in section 3.5.5 and the summary Table 3.2 in section 3.9) that often characterize the human representations of knowledge and/or captured datasets. Perhaps, we observe that by semantically integrating (De Giacomo, et al., 2018) the fuzzy models with concepts within a defined ontology, the resulting systems can make decisions like humans do. For instance, the learning questions in section 5.1 and 5.2 that allows us to determine which entities within the learning model are classified as successful learners or not. Consequently, such systems proves to offer solutions that bears the characteristics of "intelligence" which is usually attributed to humans only. Besides, this has been considered broadly as a specific feature of *Computational Intelligence* rather than literally an area of just the *Artificial intelligence* notion.

Currently, the fuzzy logic has become mature and is being used in different areas of application. For instance, as applied in the process carried out in this thesis to support the semantic-fuzzy mining approach. The intention of the technique is particularly focused on using the fuzzy logic to represent imprecise and uncertain (complex) data, for instance, the datasets provided in (Carmona, et al., 2016) used for the purpose of the work in this thesis for semantic labelling (annotation), representation (ontology), and reasoning (reasoner). In fact, the thesis have provided the semantic-fuzzy miner as a tool which can be utilised to construct process models that are easy to understand, and yet still provides implicit as well as explicit information on the extensible sets of parameters (concepts) used to determine and analyse the process models at a much more conceptual level as presented in chapters 4 and 5 of this thesis. Indeed. the proposed SPMaAF framework, algorithms formalizations and the resulting semantic-

fuzzy mining approach 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 readily available datasets 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 techniques 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 fuzzy 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 proposed in this thesis can be summarised in two forms;

(ii) encoding knowledge about specific process domains, and
(iii) advanced analysis and reasoning of processes at a much more conceptual level.

Indeed, the semantic-based fuzzy mining approach as described in this thesis 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.

On the other hand, the thesis has also shown that it is possible to integrate the fuzzy model with other tools. For example, as presented in section 5.3.1 of the thesis, the research uses the integration of the Fuzzy-BPMN approach to construct process models with notational elements that are capable of describing the nesting of individual activities (i.e process instances) by using the event-based split and join gateways - i.e. AND, XOR, and OR etc. This is because of the limitations that are generally related to the fuzzy models as noted earlier in Table 3.2 in section 3.9: where most often the fuzzy models appears to be relaxed in nature especially when compared with the semantics of other process modelling languages such as the Petri nets or BPMN. Hence, there are no explicit distinction possible between simple choice (i.e. OR split), parallel choice (i.e. AND split), or multiple choice (i.e. XOR split).

Moreover, as specified in section 5.3.1 of the thesis, the events gateways in BPMN model are token based semantics which can be used to replay a particular trace within the discovered process models (Van der Aalst, 2011; Van der Aalst, 2016) and as such overcomes the identified limitation with the fuzzy models. Hence, the amalgamation and proposal of the Fuzzy-BPMN Miner in this thesis. Besides, the work has shown through the Semantic Fuzzy Miner and the series of validation experiments, and evaluation outcomes that it is possible to improve the information values of such type of models (i.e from the syntactic) to greater extent (i.e a more conceptual level of analysis) by carefully integrating and tuning the semantics metrics that those models lack.

Chapter 7. Conclusion

7.1 Research Achievements

Fore mostly, the main target of this thesis is to identify challenges with existing process mining techniques and their influence towards the provision of worthwhile process models. In turn, the proposed method in this work allows for abstraction levels of process analysis through the semantic-based process mining approach. In view of that, the work carried out in this thesis pursues to support such semantically motivated information processing and conceptual analysis by introducing a semantic-fuzzy mining approach that targets the semantic issues as it concerns the process mining field. This entails the initial phase of collecting and transformation of the readily available datasets (i.e. event logs) to the process models discovery, and then 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 by improving the quality of the system performance as well as accuracy of the classification results.

Practically, this thesis uses the case study of the learning process domain (i.e a Research Process) and data about a real-time business process (by the IEEE CIS Task Force on Process Mining) to do the following:

- extract data from process domains to show how the work 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 learning process activities and executions.
- transform the data into mining executable formats to support the discovery of valuable process models through the employed method for annotating unlabelled learning activity sequences using ontology schema/vocabularies.
- monitor and enhance real-time processes through further semantic analysis of the discovered models.
- provide techniques for accurate classification of unseen process instances (traces) within the process models, and useful strategies towards the development of

process mining algorithms that are more intelligent, predictive and robotically adaptive with high level of semantic reasoning capabilities.

- ❖ Importance of semantics process mining to augment information values of data about domain processes: case study of learning process.

Moreover, the investigations carried out in this thesis focus on ascertaining by a series of validation experiments - how the outcome of the process mining technique and individual trace classifications can be enriched through further semantic analysis and representations of the derived models. As a result, a semantic-based fuzzy miner was developed, in addition to the various algorithms and semantically design framework proposed in this thesis.

For all intents and purposes, the work in this thesis makes use of effective semantic reasoning to enhance the process mining results and analysis from the syntactic level to a much more conceptual level by semantically representing and analysing the resulting process models. In other words, the semantic analysis uses the metadata (semantics) described in the event logs about the domain processes, and links them to concepts in an ontology in order to extract and perform a more abstract analysis of the data sets through semantic reasoning. Semantic reasoning was supported due to the formal definition of ontological concepts and expression of relationships that exist between the event logs and process models. Thus, the method makes use of the semantics of the sets of activities within the process to generate rules and events relating to task in order to automatically discover hidden traces (i.e. unobserved behaviours) and enhance the process models and ontologies through the technique for semantic annotation of the elements found within the process base.

The work also presents the semantic-based fuzzy mining approach as means towards discovering and enrichment of the sets of recurrent behaviours or patterns that can be found within any process domain with the aim to determine attributes the process elements share amongst themselves, or that distinguishes a particular set of entities (process instance) from another within the knowledge-base. Hence, the technique was developed in order to address the problem of determining the presence of different patterns (traces) within the domain process and the derived models. In fact, the drive for such an approach is: by pointing to references (i.e. the concepts - classes,

individuals, objects or datatypes property assertions/annotations) in an ontology and application of semantic reasoning, it becomes easy to classify and refer to particular cases or events within the available datasets and discovered process models. The method explains how the semantic fuzzy mining approach is able to address the presence of different patterns within the model. In other words, The intention of the semantic fuzzy mining approach is particularly focused on using the fuzzy logic to represent the imprecise and uncertain (complex) data about the domain processes, and then presents the resulting models in a format that allows one to analyse the available datasets based on concepts rather than the tags or labels within events logs about the process in view. For instance, in terms of determining which traces within the process knowledge base that are fitting (i.e true positives) or not fitting (i.e true negatives) the model (as explained in the experimentations in section 5.3), the thesis have leveraged the fuzzy logic which permits a proposal to be in another state as true or false. Thus, allowing us to determine through the classification process (i.e use of a classifier or reasoner) the presence of different patterns within the discovered models, as well as determine the total number of traces that can be replayed (fitting) or cannot be replayed (non-fitting) within the model.

Therefore, the unabridged notion of the proposed semantic fuzzy mining approach, design framework and experimental results proves that semantic concepts (i.e. annotation, ontology, and reasoning) can be layered on top of existing information asset (i.e. process models, event data logs etc.) to provide a much more easy and accurate way of analysing real time processes capable of providing real world insights and answers that can be more easily grasp by the process analysts. The work qualitatively validate this notion using a case study of the learning process, and accordingly, assess quantitatively the reliability and accuracy of the classification results of the approach using real time business data from the IEEE CIS Task Force on Process Mining. In other words, the series of experimentations and semantically motivated algorithms shows that the SPMaAF and its main application in real-time has the capability to enhance process mining results and analysis from the syntactic level to a much more conceptual level that can be easily understood by the process owners, process analysts, IT experts and Software developers as opposed to the traditional process mining techniques which do not consider the semantic aspects or informations that are contained in the event logs. Besides, the proposed method could

be used to mine and analyse events data log about any domain processes (e.g. the business process, learning process etc.) independent of any application development. Thus, the SPMaAF framework and the resulting sets of algorithms can be implemented or reused in any domain of interest be it within the same organisation that uses the model or similar operational process applied to a different sector.

7.2 Research Findings and Impact

Specifically, the findings and main impact of the approach proposed in this thesis are summarized as follows:

- Firstly, the work makes use of 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 in event data logs and process models. In principle, the work provides a method towards finding useful structures for process elements and an easy way to determine the relationships they share within a process knowledge-base.

- Secondly, the work provides a process mining technique that is able to induce new knowledge based on previously unobserved behaviours, which can be used 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 method's level and ability to accurately classify the individual traces (i.e. classification of the process elements) to predict behaviours of unobserved instances within the process knowledge-base. Principally, the work in this thesis introduce 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 is capable of being exploited to predict and/or suggest missing information in relation to process elements especially when completing large ontology-based systems. This is in fact as a result of the increase in predictive accuracy of the classifications and error-free process analysis method.

- Thirdly, this thesis propose a semantic-based fuzzy mining application (Semantic Fuzzy Miner) to realise its contribution. In addition, the work also proposed 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.

Indeed, the main purpose for designing the SPMaAF framework, semantically motivated process mining algorithms, and the semantic-based application - *Semantic-Fuzzy miner*: 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 and models through the conceptualization method.

The work also applies the case studies in this research (as implemented in chapter 5) to illustrate the technical functionalities and usefulness, level of impact and real-time application of the semantic-based process mining approach. The purpose for designing such an intelligent system is to perform a semantic process analysis of the available datasets and the discovered process models capable of providing real world answers that are closer to human understanding. For instance, the learning questions in section 5.1 to 5.2 the research addressed in order to determine the *successful* and *uncomplete* learners within the learning knowledge base.

Lastly, in addition to the research findings and claim as summarized in this section of the thesis - the research have presented the main components of a semantic-based information retrieval, extraction and processing system such the SPMaAF framework introduced in chapter 4 of this thesis. Besides, not only did the work illustrates and explain how it have integrated the main building blocks behind such type of system (i.e. the use of semantic annotation, ontology, and reasoner) to support the development and implementation of the proposed SPMaAF framework, sets of semantically motivated algorithms and the resulting Semantic Fuzzy mining approach in chapter 5. But also, the thesis empirically looked at the level of implications of the semantic-based approach and the discovered process models, validation of the classification results and its influence compared to other existing benchmark algorithms within the field of process mining in chapter 6.

7.3 Limitations and Future Work

The research investigations carried out in this thesis have identified and analysed the problems with current tools used for process mining, particularly in relation to semantics. The research has proposed a set of process mining approach that proves to be more suitable for semantic analysis of the event logs and process models. However, whilst the research believe that such methods are practically suitable for mining processes at a more conceptual level, there could also exist a number of limitations and threats to validity. In turn, whilst this thesis have introduced a set of descriptive framework and method to resolve the sets of identified problems and question that motivates the research investigations, there could be potentially many ways to address those problem, or even, bigger areas that have not been yet addressed. Owing to the fact that the semantic process mining is a new area within the process mining field, and there are not too many tools or algorithms that support such an approach currently in literature. Therefore, we assume that this work is only an incentive of more robust and intensive research within the context of the semantic-based process mining.

Clearly, the technique for semantically analysing domain process which this work introduces is one of the main important contributions of the research. Moreover, the approach is capable of extracting conceptual information from the event logs and process models. However, the correlation and integration of the building blocks (i.e. annotated logs or models, ontology, and reasoner) that underlies the proposed method assumes that the work has presented an approach which can possibly be re-introduced or extended in a more resourceful way.

An additional limitation of the approach in this thesis is that it appears to be a fusion theory which integrates the fuzzy model with other tools. In many settings, fuzzy models have proven to be ambiguous and characteristically contains vast number of arc nodes which are disjointed via impounded nodes that are primitive in nature. Therefore, with such process model, it may not be probable to extract meaningful informations about the process elements. Even though this work has shown that it is possible to improve the information values of such type of models to some greater extent by carefully integrating and tuning the semantics metrics that those models lack, the process seems to be a cumbersome task and does not guarantee and/or carry some threats to the validity of the outcomes.

Moreover, another threat to validity of the research is that there are no currently tools capable of directly converting the fuzzy models into some other modelling formats or notation. As a consequence, the work leverages a varied range of event log conversion in order to achieve different viewpoint about the domain processes. Indeed, future works could focus on extending the approach through provision of tools capable of automatically integrating such metrics with the models in order to support their analysis at a more abstract level and better still guarantee the accuracy of the results. Besides, this work has shown that a way to resolve those problems is to provide the option for specifying semantics which in turn is capable of allowing for a more well-structured and precise format for such models.

Finally, the research believes that there is a lot of opportunities for future works in extending the proposed approach in this thesis. Besides, a worthwhile extension will be to complement the approach with a platform for completely automatic discovering and/or integration of the semantic informations.

Nonetheless, the semantic-based approach represented in this thesis is one of the many methods for process management and information processing techniques which can be utilized to analyse event logs and derive meaningful process models about any process domain. Obviously, there are several directions towards which the outcomes and proposals of this research investigations can be improved or further extended in the future. Future works can adopt the proposed approach in this thesis to analyse data extracts from any domain area of interest or settings, including refinement of the realized semantic learning process mining algorithms and techniques that has already been developed in this thesis. Such extensions may include more sophisticated increment of the ontology schemas and reasoning capabilities that has been defined in this thesis.

In addition to the aforementioned areas that could be considered for future works, another potentially worthwhile area to pursue in the future is to expound the current system to include and spread out to diverse organisations or business owners in their current process settings. This may include the development of authoring tools capable of augmenting the stated achievements of this thesis, or yet, the mentioned process domains as well as their operational processes in real world settings.

References

Adriansyah, A., Van Dongen, B. & Van der Aalst, W. M. P., 2011. *Conformance Checking using Cost-Based Fitness Analysis..* Helsinki, Finland., IEEE International Enterprise Computing Conference (EDOC 2011) IEEE Computer Society.

Adriansyah, A., Van Dongen, B. & Van der Aalst, W. M. P., 2011. Towards Robust Conformance Checking. In: J. S. M. zur Muehlen, ed. *Lecture Notes in Business Information Processing.* Berlin: BPM 2010 Workshops, Proceedings of the 6th Workshop on Business Process Intelligence (BPI2010), pp. 122-133.

Aggarwal, C., 2007. *Data Streams: Models and Algorithms.* Volume 31 of Advances in Database Systems ed. Berlin: Springer, Berlin.

Alkharouf, N. W., Jamison, D. C. & Matthews, B. F., 2005. Online Analytical Processing (OLAP): A Fast and Effective Data Mining Tool for Gene Expression Databases. *Journal of Biomedicine and Biotechnology,* 2005(2), p. 181–188.

Allahyari, M., Kochut, K. J. & Janik, M., 2014. *Ontology-based text classification into dynamically defined topics.* Newport Beach, California, USA, IEEE International Conference on Semantic Computing (ICSC) 2014, pp. 273-278.

Baader, F. et al., 2003. *Description Logic Handbook: theory, implementation, and applications.* 1 ed. New York, NY, USA: Cambridge University Press.

Baati, K., Hamdani, T. M., Alimi, A. M. & Abraham, A., 2016. A Modified Na¨ıve Possibilistic Classifier for Numerical Data. In: A. Madureira, A. Abraham, D. Gamboa & P. Novais, eds. *Intelligent Systems Design and Applications. ISDA 2016. Advances in Intelligent Systems and Computing, vol 557..* Cham, Switzerland: Proceedings of the 16th International Conference on Intelligent Systems Design and Applications, Springer, Cham, pp. 417-426.

Baati, K., Hamdani, T. M., Alimi, A. M. & Abraham, A., 2016. A New Possibilistic Classifier for Heart Disease Detection From Heterogeneous Medical Data. *International Journal of Computer Science and Information Security,* 14(7), p. 443–450.

Baati, K., Hamdani, T. M., Alimi, A. M. & Abraham, A., 2017. Decision quality enhancement in minimum-based possibilistic classification for numerical data. In: A. Abraham, A. K. Cherukuri, A. M. Madureira & A. K. Muda, eds. *Advances in Intelligent Systems and Computing.* Springer International Publishing AG: Springer International Publishing, Proceedings of the 8th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2016).

Baker, R. & Yacef, K., 2009. The state of educational data mining in 2009: A review and future visions.. *Journal of Educational Data Mining,* 1(1), pp. 3-17.

Balcan, N., Blum, A. & Mansour, Y., 2013. *Exploiting ontology structures and unlabeled data for learning.* Atlanta Georgia, USA, Proceedings of the 30th Int. Conference on Machine Learning 2013, pp. 1112-1120.

Bechhofer, S., 2003. *OWL Reasoning Examples,* Manchester, UK: University of Manchester.

Bechhofer, S. et al., 2004. *OWL Web Ontology Language Reference,* Manchester, UK: Technical report W3C Proposed Recommendation.

Bemers-Lee, T., Hendler, J. & Lassila, O., 2001. The Semantic Web. *Scientific American,* 284(5), pp. 28-37.

Bishop, B. et al., 1999. *WSML Reasoner,* Boston, MA: IRIS Reasoner - SOA4All.

Bogarín, A., Cerezo, R. & Romero, C., 2018. A Survey on Educational Process Mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery (WIRES), Wiley Periodicals, Inc. publisher*, 1 January/February, 8(1), p. e1230.

Bogarín, A., Romero, C., Cerezo, R. & Sánchez-Santillán, M., 2014. Clustering for improving educational process mining. *ACM, New York, NY, USA,* pp. 11 - 15.

Bontcheva, K., 2004. *Open-source Tools for Creation, Maintenance, and Storage of Lexical Resources for Language Generation from Ontologies..* Centro Cultural de Belem, Lisbon, Portugal, Proceedings of 4th Language Resources and Evaluation Conference (LREC'04).

Born, M., Dörr, F. & Weber, I., 2007. User-Friendly Semantic Annotation in Business Process Modeling. In: M. Weske, M. Hacid & C. Godart, eds. *Web Information Systems Engineering – WISE 2007 Workshops. WISE 2007. Lecture Notes in Computer Science, vol 4832..* Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 260-271.

Brewster, C. & O'Hara, K., 2007. Knowledge representation with ontologies: Present challenges future possibilities. *International Journal of Human-Computer Studies,* 65(7), p. 563 – 568.

Buijs, J. C. A. M., 2014. *Flexible evolutionary algorithms for mining structured process models,* Eindhoven, the Netherlands: PHD Thesis, Eindhoven, Technische Universiteit Eindhoven.

Cairns, A. H. et al., 2014. *Towards Custom-Designed Professional Training Contents and Curriculums through Educational Process Mining.* Paris, France, IMMM 2014 : The Fourth International Conference on Advances in Information Mining and Management, pp. 53 -58.

Cairns, A. H. et al., 2015. Process Mining in the Education Domain. *International Journal on Advances in Intelligent Systems,* vol 8(1 & 2).

Cairns, A. H. et al., 2014. *Using Semantic Lifting for Improving Educational Process Models Discovery and Analysis.* s.l., SIMPDA, volume 1293 of CEUR Workshop Proceedings, CEUR-WS.org, pp. 150-161.

Calvanese, D. et al., 2005. *DL-Lite: Tractable description logics for ontologies.* Pittsburgh, Pennsylvania , Proc. of AAAI.

Calvanese, D. et al., 2009. *Ontologies and databases: The DL-Lite approach..* Berlin Heidelberg, In Proc. of RW. Springer-Verlag Berlin Heidelberg, pp. 255-356.

Calvanese, D., Kalayci, T. E., Montali, M. & Tinella, S., 2017. Ontology-based Data Access for Extracting Event Logs from Legacy Data: The onprom Tool and Methodology. In: W. Abramowicz, ed. *Business Information Systems (BIS 2017). Volume 288 of Lecture Notes in Business Information Processing.* 1 ed. Springer International Publishing: Proceedings of 20th International Conference on Business Information Systems 2017 Poznan, Poland, p. In Press.

Calvanese, D., Montali, M., Syamsiyah, A. & van der Aalst, W. M. P., 2016. Ontology-Driven Extraction of Event Logs from Relational Databases. In: M. Reichert & H. Reijers, eds. *Lecture Notes in Business Information Processing.* Cham: Business Process Management Workshops. BPM 2015. Springer, Cham, pp. 140-153.

Carmona, J., de Leoni, M., Depair, B. & Jouck, T., 2016. *Process Discovery Contest @ BPM 2016,* Rio de Janeiro: IEEE CIS Task Force on Process Mining.

Cesarini, M., Monga, M. & Tedesco, R., 2004. *Carrying on the e-learning process with a workflow management engine.* Nicosia, Cyprus, proceedings of ACM Symposium on Applied Computing, pp. 940-945.

Chen, C., 2008. Intelligent web-based learning systems with personalized learning path guidance. *Computers and Education,* Volume 51, pp. 779-784.

Chesani, F., Ciampolini, A., Loreti, D. & Mello, P., 2016. Process Mining Monitoring for Map Reduce Applications in the Cloud. *ACM Digital Library - Proceedings of the 6th International Conference on Cloud Computing and Services Science ,* 1 and 2(1), pp. 95-105 .

Clark & Parsia, L., University of Manchester, U. & University of Ulm, G., 2017. *The OWL API,* Manchester, UK: Sourceforge.net - original version API for OWL 1.0 developed as part of the WonderWeb Project.

Cunningham, H., 2005. *Information Extraction, Automatic,* Sheffield, UK: University of Sheffield.

Cunningham, H. et al., 1995. *GATE: General Architecture for Text Engineering.* [Online] Available at: https://gate.ac.uk/ [Accessed 20 June 2017].

d'Amato, C., Fanizzi, N. & Esposito, F., 2008. Query answering and ontology population: An inductive approach, . In: S. Bechhofer, M. Hauswirth, J. Hoffmann & M. Koubarakis, eds. *Proceedings of the 5th Euro. Semantic Web Conference, ESWC2008. Vol. 5021 of LNCS.* Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 288-302.

Dammak, S. M., Jedidi, A. & Bouaziz, R., 2014. *Fuzzy semantic annotation of Web resources.* Sousse, 2014 World Symposium on Computer Applications & Research (WSCAR), pp. 1-6.

Davies, J., Fensel, D. & van Harmelen, F., 2002. *Towards the Semantic Web: Ontology-driven Knowledge Management.* 1 ed. Chichester, UK: John Wiley & Sons, Ltd.

de Beer, H. T., 2005. *The LTL Checker Plugins: a (reference) manual,* Eindhoven, the Netherlands: processmining.org.

de Bruijn, J., Lausen, H., Polleres, A. & D., F., 2006. The Web Service Modeling Language WSML: An Overview. In: Y. Sure & J. Domingue, eds. *The Semantic Web: Research and Applications. ESWC 2006. Lecture Notes in Computer Science, vol 4011.* Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 590-604.

De Giacomo, G. et al., 2018. Using Ontologies for Semantic Data Integration. In: S. Flesca, S. Greco, E. Masciari & D. Saccà, eds. *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years. Studies in Big Data.* Cham: Springer, Cham, pp. 187-202.

De Leoni, M. & Van der Aalst, W. M. P., 2013. Data-Aware Process Mining: Discovering Decisions in Processes Using Alignments. In: S. Y. Shin & J. C. Maldonado, eds. *ACM Symposium on Applied Computing (SAC 2013).* Coimbra, Portugal: ACM Press, New York, NY, pp. 1454-1461.

de Leoni, M., Van der Aalst, W. M. P. & Dees, M., 2016. A General Process Mining Framework for Correlating, Predicting and Clustering Dynamic Behaviour Based on Event Logs. *Information Systems,* 56(1), pp. 235-257.

de Leoni, M., Van der Aalst, W. M. P. & ter Hofstede, A. H. M., 2012. Visual Support for Work Assignment in Process-Aware information Systems: Framework Formalisation and Implementation. *Decision Support Systems,* 54(1), pp. 345-361.

de Leoni, M., Van der Aalst, W. & ter Hofstede, A., 2008. Visual Support for Work Assignment in Process-Aware Information Systems. In: M. Dumas, M. Reichert & M. Shan, eds. *Business Process Management. BPM 2008. Lecture Notes in Computer Science, vol 5240..* Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 67-83.

de Medeiros, A. K. A., 2006. *Genetic Process Mining,* Eindhoven, the Netherlands: PHD Thesis Technische Universiteit Eindhoven.

de Medeiros, A. K. A. & Van der Aalst, W. M. P., 2009. Process Mining towards Semantics. In: T. S. Dillon, E. Chang, R. Meersman & K. Sycara, eds. *Advances in Web Semantics, Lecture Notes in Computer Science.* Berlin: Springer Berlin Heidelberg, pp. 35-80.

de Medeiros, A. K. A. & Van der Aalst, W. M. P., 2009. Process Mining towards Semantics. In: T. Dillon, E. Chang, R. Meersman & K. Sycara, eds. *Advances in Web Semantics I. Lecture Notes in Computer Science, vol 4891.* Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 35-80.

de Medeiros, A. K. A., Weijters, A. J. & van der Aalst, W. M. P., 2007. Genetic Process Mining: An Experimental Evaluation. *Data Mining and Knowledge Discovery ,* 14(2), p. 245–304 .

deMedeiros, A., van der Aalst, W. M. P. & Pedrinaci, C., 2008. *Semantic Process Mining Tools: Core Building Blocks.* Galway, Ireland,, ECIS, June 2008, pp. 1953-1964.

Dill, S. et al., 2003. *SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation.* Budapest, Hungary, Proceedings of WWW'03.

Dolog, P. &. N. W. (., 2007. Semantic web technologies for the adaptive web. In: P. Brusilovsky, A. Kobsa & W. Nejdl, eds. *The adaptive web Volume 4321 of the series Lecture Notes in Computer Science* . Berlin, Heidelberg,: Springer-Verlag, Berlin, Heidelberg, pp. 697-719.

Domingue, J., Dzbor, M. & Motta, E., 2004. *Magpie: Supporting Browsing and Navigation on the Semantic Web..* Funchal, Portugal, In: Nunes, N., Rich, C. (Eds.), Proceedings ACM Conference on Intelligent User Interfaces (IUI).

Dou, D., Wang, H. & Liu, H., 2015. *Semantic Data Mining: A Survey of Ontology-based Approaches.* California, USA, 9th IEEE Int. Conference on Semantic Computing, p. 244 – 251.

Dubois, D. et al., 1988. *Possibility theory: an approach to computerized processing of uncertainty 2.* 2 ed. New York:: Plenum press.

Dumas, M., van der Aalst, W. M. P. & ter Hofstede, A. H. M., 2005. *Process-Aware Information Systems: Bridging People and Software through Process Technology.* 1 ed. New York, NY: Wiley.

Dżega, D. & Pietruszkiewicz, W., 2013. Intelligent Decision-Making Support within the E-Learning Process. In: A. Peña-Ayala, ed. *Intelligent and Adaptive Educational-Learning Systems: Achievements and Trends.* Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 497-521.

Elhebir, M. H. A. & Abraham, A., 2015. A Novel Ensemble Approach to Enhance the Performance of Web Server Logs Classification. *International Journal of Computer Information Systems and Industrial Management Applications ,* 7(2015), pp. 189-195.

Erdmann, M., Maedche, A., Schnurr, H. p. & Staab, S., 2000. *From manual to semi-automatic semantic annotation: about ontology-based text annotation tools.* Luxembourg, Proceedings of the COLING-2000 Workshop on Semantic Annotation and Intelligent Content, pp. 79-85 .

Fahland, D. & van der Aalst, W. M. P., 2012. Repairing Process Models to Reflect Reality. In: B. A., G. A. & K. E., eds. *Business Process Management. BPM 2012. Lecture Notes in Computer Science, vol 7481.* Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 229-245.

Fensel, D., Hendler, J., Wahlster, W. & Lieberman, H., 2002. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential.* 1 ed. Cambridge: MIT Press.

Fensel, D. et al., 2000. *OIL in a Nutshell.* Juan-les-Pins, France, EKAW '00 Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management, pp. 1-16 .

Ferreira, C., Gama, J. & Santos Costa, V., 2012. Predictive Sequence Miner in ILP Learning. In: M. S.H., T. A. & L. F.A., eds. *Inductive Logic Programming. ILP 2011. vol 7207 of Lecture Notes in Computer Science.* Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 130-144.

Frias-Martinez, E., Chen, S. & Liu, X., 2006. Survey of Data Mining approaches to user modeling for adaptive hypermedia. *IEEE TSMC-Part C: Applications and Reviews,* 36(6), pp. 734-749.

Frosch-Wilke, D., Sanchez-Alonso, S. & Dodero, J. M., 2008. *Evolutionary Design of Collaborative Learning Processes through Reflective Petri Nets.* Santander, Cantabria, Spain, IEEE ICALT, pp. 423-427.

G¨unther, C., 2009. *Process Mining in Flexible Environments.,* Eindhoven, the Netherlands: PhD thesis, Department of Technology Management, Technical University Eindhoven.

Gatner, 2010. *Magic Quadrant for Business Intelligence Platforms.* [Online] Available at: www.gartner.com [Accessed 20 June 2017].

Gavetti, G. M., Henderson, R. & Giorgi, S., 2004. *Kodak and The Digital Revolution.* Case 705-448 ed. USA: Havard Business School .

Ghawi, R., 2016. *Process Discovery using Inductive Miner and Decomposition,* Rio de Janeiro: In CoRR abs/1610.07989 (2016) Technical Report Submission for the Process Discovery Contest @ BPM 2016, [1st Edition], IEEE Task Force on Process Mining.

Goedertier, S., Martens, D., Vanthienen, J. & Baesens, B., 2009. Robust Process Discovery with Artificial Negative Events. *Journal of Machine Learning Research,* 10(1), pp. 1305-1340.

Gold, E. M., 1967. Language Identification in the Limit. *Information and Control,* 10(5), pp. 447-474.

Gomez, G., Manser, M., Sanchez, J. & al, e., 2002. *Bizagi - Time to Digital.* [Online] Available at: https://www.bizagi.com/en/products/bpm-suite/modeler [Accessed 2 May 2015].

Greco, G., Guzzo, A., Pontieri, L. & Sacca, D., 2006. Discovering Expressive Process Models by ClusteringLog Traces. *IEEE Transaction on Knowledge and Data Engineering,* 18(8), pp. 1010-1027.

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

Gruber, T. R., 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition,* 5(2), pp. 199-220.

Gruber, T. R., 1995. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies,* 43(5), pp. 907-928.

Gu¨nther, C. W. & Van der Aalst, W. M. P., 2006. A generic import framework for process event logs. In: J. Eder & S. Dustdar, eds. *Business Process Management Workshops.* Berlin: Springer, Berlin,, pp. 81-92.

Gunther, C. et al., 2008. Using process mining to learn from process changes in evolutionary systems. *International Journal of Business Process Integration and Management,* 3(1), pp. 61-78.

Gunther, C. W., 2009. *OpenXES - Developer Guide,* Netherlands: IEEE 1849-2016 XES.

Günther, C. W. & Van der Aalst, W. M. P., 2007. Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics. In: A. G., D. P. & R. M., eds. *Business Process Management. BPM 2007. Lecture Notes in Computer Science, vol 4714.* Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 328-343.

Haarslev, V. & Möller, R., 2001. RACER System Description. In: R. Goré, A. Leitsch & T. Nipkow, eds. *Automated Reasoning. IJCAR 2001. Lecture Notes in Computer Science, vol 2083.* . Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 701-705.

Han, J. & Kamber, M., 2004. *Data Mining: Concepts and Techniques.* 1 ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Han, J., Kamber, M. & Pei, J., 2011. *Data Mining: Concepts and Techniques.* 3 ed. Burlington, Massachusetts, United States: The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publishers.

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

Heflin, J. & Hendler, J., 2000. *Searching the web with SHOE.* Austin, Texas, AAAI-2000 Workshop on AI for Web Search, pp. 35-40.

Helic, D., 2006. Technology-Supported management of Collaborative Learning Processes. *International Journal of Learning and Change,* 1(3), pp. 285-298.

Herbst, J., 2001. *Ein induktiver Ansatz zur Akquisition und Adaption von Workflow-Modellen,* Baden-Württemberg, Germany: PhD thesis, Universit□at Ulm.

Hiroshi, S., 1997. *What is Occam's Razor?,* California: Original by Phil Gibbs 1996 - university of California Riverside.

Holzhüter, M., Frosch-Wilke, D. & Klein, U., 2013. Exploiting Learner Models Using Data Mining for E-Learning: A Rule Based Approach. In: A. Peña-Ayala, ed. *Intelligent and Adaptive Educational-Learning Systems: Achievements and Trends.* Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 77-105.

Holzhüter, M., Frosch-Wilke, D. & Sanchez-Alonso, S., 2010. *Discussion of the benefit potentials of process mining for e-learning processes.* Valencia, Spain, Proceedings of CSEDU, pp. 407-411.

Horrocks, I., 2002. Daml+oil: a description logic for the semantic web. *IEEE Data Engineering Bulletin,* 25(1), pp. 4-9.

Horrocks, I., 2008. Ontologies and the semantic web. *Communications of the ACM,* 51(12), pp. 58-67.

Horrocks, I. et al., 2004. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML,* Network Inference, Canada and Stanford University: W3C Member Submission - 2004 National Research Council of Canada, Network Inference, and Stanford University.

Horrocks, I., Patel-Schneider, P. F., McGuinness, D. L. & Welty, C. A., 2007. Owl: a description logic based ontology language for the semantic web. In: F. Baader, et al. eds. *The Description*

Logic Handbook: Theory, Implementation, and Applications. 2 ed. New York, NY: Cambridge University Press, p. 458–486.

Horrocks, I., Patel-Schneider, P. F. & van Harmelen, F., 2003. From shiq and rdf to owl: The making of a web ontology language. *Journal of Web Semantics,* 1(1), pp. 7-26.

Hosseini, S. A., Tawil, A.-R. H., Jahankhani, H. & Yarandi, M., 2013. Towards an ontological learners modelling approach for personalised e-learning. *International Journal of Emerging Technologies in Learning (iJET),* 8(2), pp. 4-10.

IEEE 1849-2016, X., 2016. *OpenXES - reference implementation of the First XES standard.* [Online] Available at: http://www.xes-standard.org/openxes/start [Accessed 20 May 2017].

IEEE BigData, 2017. *IEEE International Conference on Big Data(Big Data 2017).* [Online] Available at: http://cci.drexel.edu/bigdata/bigdata2017/CallPapers.html [Accessed 23 June 2017].

IEEE CIS Task Force on Process Mining, X., 2016. *1849-2016 - IEEE Standard for eXtensible Event Stream definition.* [Online] Available at: http://www.xes-standard.org/, [Accessed 22 June 2017].

IEEE Standards, 1.-2., 2016. *IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams.* [Online] Available at: http://ieeexplore.ieee.org/document/7740858/ [Accessed 26 June 2017].

Ingvaldsen, J. E., 2011. *Semantic process mining of enterprise transaction data,* Norway: PhD Thesis - Norwegian University of Science and Technology.

Ingvaldsen, J. E., Gulla, J. A., Hegle, O. A. & Prange, A., 2005. *Revealing the real business flows from enterprise systems transactions.* Miami Beach, Florida , 7th International Conference on Enterprise Information Systems.

Jablonski, S. & Bussler, C., 1996. *Workflow Management: Modeling Concepts, Architecture, and Implementation.* 1 ed. London: International Thomson Computer Press.

Jans, M. J., 2011. Process Mining in Auditing: From Current Limitations to Future Challenges. In: D. F., B. K. & D. S., eds. *Lecture Notes in Business Information Processing.* Berlin, Heidelberg: International Conference on Business Process Management Workshops. BPM 2011. Springer,, pp. 394-397.

Jareevongpiboon, W. & Janecek, P., 2013. Ontological approach to enhance results of business process mining and analysis. *Journal of Business Process Management,* 19(3), p. 459 – 476.

Jensen, K. & Kristensen, L., 2009. *Coloured Petri Nets.* 1 ed. Berlin: Springer, Berlin.

Kacalak, W., Majewski, M. & Zurada, J., 2010. Intelligent E-Learning Systems for Evaluation of User's Knowledge and Skills with Efficient Information Processing.. In: L. S. R. T. R. Z. L. Z. J. Rutkowski, ed. *ICAISC 2010 Lecture Notes in Artificial Intelligence.* Heidelberg: Springer Heidelberg, pp. 508-515.

Karel, R., 2011. *Stop Trying To Put A Monetary Value On Data – It's The Wrong Path ,* Cambridge, MA 02140 : Forrester Research, Inc..

Karp, P. D., Chaudhri, V. K. & Thomere, J., 1999. *Xol: An xml-based ontology exchange language, Technical report, SRI International.,* Pangea Systems Inc.: Technical report, SRI International , Pangea Systems Inc.

Kay, J. & Lum, A., 2004. Building User Models from Observations of Users Accessing Multimedia Learning Objects. In: A. D. M. Nurnberger, ed. *AMR 2003 LNCS.* Heidelberg: Springer Heidelberg, pp. 36-57.

Khaleghi, B., Khamis, A., Karray, F. O. & Razavi, S. N., 2013. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion,* 14(1), pp. 8-44.

Khasawneh, N. & Chan, C. -C., 2006. *Active user-based and ontology-based web log data preprocessing for web usage mining..* Washington, DC, USA, Proceedings of the 2006 IEEEIWIClACM International Conference on Web Intelligence, pp. 325-328.

Kifer, M., Lausen, G. & Wu, J., 1995. Logical foundations of object-oriented and frame-based languages. *Journal of ACM,* 42(4), p. 741–843.

King, M., 2003. *Living up to standards.* Budapest, Hungary., In Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing.

Kiryakov, A. et al., 2004. Semantic annotation, indexing, and retrieval.. *Web Semantics: Science, Services and Agents on the World Wide Web,* 2(1), pp. 49-79.

Kumar, A. P., Abhishek, K. & Vipin Kumar, N., 2011. Architecting and Designing of Semantic Web Based Application using the JENA and PROTÉGÉ – A Comprehensive Study. *International Journal of Computer Science and Information Technologies (IJCSIT) ,* 2(3), pp. 1279-1282.

Lausen, H., de Bruijn, J., Polleres, A. & Fensel, D., 2005. *WSML - A Language Framework for Semantic Web Services.* Washington DC, USA, Position Paper for the W3C rules workshop.

Lautenbacher, F., Bauer, B. & Forg, S., 2009. *Process Mining for Semantic Business Process Modeling.* Auckland, 13th Enterprise Distributed Object Computing Conference Workshops, pp. 45-53.

Lautenbacher, F., Bauer, B. & Seitz, C., 2008. *Semantic Business Process Modeling - Benefits and Capability.* California, USA, AAAI Spring Symposium: AI Meets Business Rules and Process Management, Stanford University.

Leemans, S. J. J., D., F. & Van der Aalst, W. M. P., 2013. Discovering Block-Structured Process Models from Event Logs: A Constructive Approach. In: J. M. Colom & J. Desel, eds. *Applications and Theory of Petri Nets 2013, Volume 7927 of LNCS.* Berlin: Springer, Berlin, pp. 311-329.

Leemans, S. J. J., D., F. & Van der Aalst, W. M. P., 2014a. Discovering Block-Structured Process Models from Incomplete Event Logs.. In: G. Ciardo & E. Kindler, eds. *Applications and Theory of Petri Nets 2014, volume 8489 of Lecture Notes in Computer Science.* Berlin: Springer, Berlin, pp. 91-110.

Leemans, S. J. J., D., F. & Van der Aalst, W. M. P., 2014. Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour.. In: N. Lohmann, M. Song & P. Wohed, eds. *Business Process Management Workshops, International Workshop on Business Process Intelligence (BPI), volume 171 of Lecture Notes in Business Information Processing.* Berlin: Springer, Berlin, pp. 66-78.

Leemans, S. J. J., D., F. & Van der Aalst, W. M. P., 2015. Scalable Process Discovery with Guarantees. In: K. Gaaloul, et al. eds. *Enterprise, Business-Process and Information Systems*

Modeling (BPMDS 2015), volume 214 of Lecture Notes in Business Information Processing. Berlin: Springer, Berlin, pp. 85-101.

Leemans, S. J. J., Fahland, D. & P., v. d. A. W. M., 2015. Exploring Processes and Deviations. In: F. Fournier & J. Mendling, eds. *Business Process Management Workshops. BPM 2014. Lecture Notes in Business Information Processing, vol 202.* Cham: Springer, Cham, pp. 304-316.

Leemans, S. J. J., Fahland, D. & van der Aalst, W. M. P., 2014. *Process and Deviation Exploration with Inductive visual Miner.* Eindhoven, The Netherlands, Proceedings of the BPM Demo Sessions 2014.

Lehmann, J. & Hitzler, P., 2010. Concept learning in description logics using refinement operators. *Machine Learning, vol. 78 (2010) 203-250.,* 78(1-2), pp. 203-250.

Leymann, F. & Roller, D., 1999. *Production Workflow: Concepts and Techniques.* 1 ed. Upper Saddle River, NJ: Prentice-Hall.

Lisi, F., 2008. Building Rules on Top of Ontologies for the Semantic Web with Inductive Logic Programming. *Theory and Practice of Logic Programming,* 8(3), p. 271–300.

Lisi, F. & Esposito, F., 2007. *Building Rules on top of Ontologies? Inductive Logic Programming can help!,* Bari, Italy: SWAP 2007.

M¨uller, H.-M., Kenny, E. E. & Sternberg, P. W., 2004. Textpresso: An ontology-based information retrieval and extraction system for biological literature. *PLoS Biology,* 2(11), p. e309.

Maita, A. et al., 2017. A systematic mapping study of process mining. *Enterprise Information Systems,* 0(0), pp. 1-45.

Manna, Z. & Pnueli, A., 1992. *The Temporal Logic of Reactive and Concurrent Systems: Specification.* 1 ed. New York: Springer-Verlag New York.

Mannila, H., Toivonen, H. & Verkamo, A. I., 1997. Discovery of Frequent Episodes in Event Sequences. *Data Mining and Knowledge Discovery,* 1(3), pp. 259-289.

Manning, C. D., Raghavan, P. & Schütze, H., 2008. *Introduction to Information Retrieval,* Cambridge: Cambridge University Press.

Mans, R., Van der Aalst, W. M. P. & Vanwersch, R., 2015. *Process Mining in Healthcare: Evaluating and Exploiting Operational Healthcare Processes.* 1 ed. Berlin: Springer Briefs in Business Process Management, Springer Berlin.

Maynard, D., Peters, W. & Li, Y., 2006. *Metrics for evaluation of ontology-based information extraction..* Edinburgh, Scotland, In WWW 2006 Workshop on "Evaluation of Ontologies for the Web" (EON) .

Maynard, D., Peters, W. & Li, Y., 2008. *Evaluating Evaluation Metrics for Ontology-Based Applications: Infinite Reflection.* Marrakech, Morocco, Conference: Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June.

Maynard, D. et al., 2007. Natural Language Technology for Information Integration in Business Intelligence.. In: W. Abramowicz, ed. *Business Information Systems. BIS 2007. Lecture Notes in Computer Science, vol 4439.* Berlin, Heidelberg : Springer, Berlin, Heidelberg , pp. 366-380.

Miani, R. G. L. & Hruschka Junior, E. R., 2015. Exploring Association Rules in a Large Growing Knowledge Base. *International Journal of Computer Information System and Industrial Management,* 7(2015), p. 106–114.

Millan, E., Agosta, J. & Perez de la Cruz, J., 2001. Bayesian student modeling and the problem of parameter specification.. *British Journal of Education Technology,* 32(2), pp. 171-181.

Montani, S. et al., 2017. Knowledge-Based Trace Abstraction for Semantic Process Mining. In: A. ten Teije, C. Popow, J. Holmes & L. Sacchi, eds. *Artificial Intelligence in Medicine. AIME 2017. Lecture Notes in Computer Science, vol 10259.* Australia: Springer, Cham, pp. 267-271.

Munoz-Gama, J. & Carmona, J., 2011. *Enhancing Precision in Process Conformance: Stability, Confidence and Severity.* Paris, France, Proceedings of CIDM 2011. IEEE.

Musen, M. A. et al., 2015. The Protégé project: A look back and a look forward. AI Matters.. *AI Matters,* 1(4), p. 4–12.

Naur, P., 1974. *Concise Survey of Computer Methods.* 1 ed. Kobenhaven: Studentlitteratur Lund, Akademisk Forlag.

Nguyen, L. & Phung, D., 2008. Learner model in adaptive learning. *World Academy of Science, Engineering and Technology,* Volume 45, pp. 395-400.

Obitko, M., 2007. *Knowledge Interchange Format,* Prague 6, Czech Republic: Ontologies and Semantic Web.

Obitko, M., 2007. *Web Ontology Language OWL,* Prague 6, Czech Republic: Ontologies and Semantic Web.

Okoye, K. et al., 2016. *Using semantic-based approach to manage perspectives of process mining: Application on improving learning process domain data.* Washington, D.C, Proceedings of 2016 IEEE International Conference on Big Data (Big Data), pp. 3529-3538.

Okoye, K. et al., 2017. Semantic-based Model Analysis towards Enhancing Information Values of Process Mining: Case Study of Learning Process Domain. In: A. Abraham, A. K. Cherukuri, A. M. Madureira & A. K. Muda, eds. *Advances in Intelligent Systems and Computing.* Springer International Publishing AG: Springer International Publishing.

Okoye, K., Tawil, A. R. H., Naeem, U. & Lamine, E., 2014. A Semantic Rule-based Approach Supported by Process Mining for Personalised Adaptive Learning. *Procedia Computer Science,* 37(C), pp. 203-210.

Okoye, K., Tawil, A. R. H., Naeem, U. & Lamine, E., 2016. A Semantic Reasoning Method Towards Ontological Model for Automated Learning Analysis. In: N. Pillay, et al. eds. *Advances in Intelligent Systems and Computing.* Springer Cham Switzerland: Advances in Nature and Biologically Inspired Computing, Proceedings of NaBIC Conference, 2015, Springer International Publishing, pp. 49-60.

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

Okoye, K., Tawil, A. R. H., Naeem, U. & Lamine, E., 2016. *Fuzzy-BPMN miner approach - Process Discovery Contest @ BPM 2016,* Rio de Janeiro: Technical Report Submission, IEEE CIS Task Force on Process Mining discovery contest [1st Edition] in BPI workshop at BPM 2016 Conference.

Ouksel, A. M. & Sheth, A., 1999. Semantic interoperability in global information systems. *SIGMOD Rec. ,* 28(1), pp. 5-12.

Park, S., Byun, D., Park, D. & Lee, H., 2005. *Evaluation System in e-Learning Through the Knowledge State,* Saudi Arabia: Recent Research Developments in Learning Technologies.

Patel-Schneider, P. F., Hayes, P. & Horrocks, I., 2004. *Owl web ontology language semantics and abstract syntax,* Bell Labs Research, Lucent Technologies, New Jersy, USA: W3C Recommendation.

Pechenizkiy, M. et al., 2009. *Process Mining Online Assessment Data.* Cordoba, Spain, Proceedings of EDM, pp. 279-288.

Peña-Ayala, A., 2013. *Intelligent and Adaptive Educational-Learning Systems: Achievements and Trends.* 1 ed. Heidelberg: Springer-Verlag Berlin .

Peña-Ayala, A. & Sossa, H., 2013. Proactive Sequencing Based on a Causal and Fuzzy Student Model. In: A. Peña-Ayala, ed. *Intelligent and Adaptive Educational-Learning Systems: Achievements and Trends.* Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 49 - 76.

Perez-Rey, D., Anguita, A. & J., C., 2006. OntoDataClean: Ontology-Based Integration and Preprocessing of Distributed Data. In: N. Maglaveras, I. Chouvarda, V. Koutkias & B. R., eds. *Biological and Medical Data Analysis, ISBMDA 2006. Lecture Notes in Computer Science, vol 4345.*. Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 262-272.

Perez-Rodriguez, R., Caeiro-Rodriguez, M. & Anido-Rifon, L., 2008. *Supporting PoEML educational processes in Moodle: A middleware approach.* Universidad Pontificia de Salamanca, Proceedings of SPDECER.

Poggi, A. et al., 2008. Linking Data to Ontologies. *In: Spaccapietra S. (eds) Journal on Data Semantics,* vol 4900(1), pp. 133-173.

Polyvyanyy, A. et al., 2015. *Process Querying in Apromore..* Innsbruck, Austria,, Proceedings of the BPM Demo Session 2015 Co-located with the 13th International Conference on Business Process Management (BPM Demos 2015), pp. 105-109.

Polyvyanyy, A. & et al, 2016. *Process Querying.* [Online] Available at: http://processquerying.com/ [Accessed 04 July 2017].

Polyvyanyy, A., Ouyang, C., Barros, A. & van der Aalst, W. M. P., 2017. Process querying: Enabling business intelligence through query-based process analytics. *Decision Support Systems,* In Press(-), pp. -.

Popov, B. et al., 2004. KIM – Semantic Annotation Platform. *Journal of Natural Language Engineering ,* 10(3-4), pp. 375--392 .

Rojas, E., Munoz-Gama, J., Sepúlveda, M. & Capurro, D., 2016. Process mining in healthcare: A literature review. *Journal of Biomedical Informatics,* 61(1), pp. 224-236.

Romero, C. & Ventura, S., 2013. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and,* Volume 3, p. 12–27.

Rouse, M., 2015. *Microsoft FAST Search,* New York: TechTarget.com.

Rozinat, A., 2010. *Process Mining: Conformance and Extension,* Eindhoven, the Netherlands: PhD Thesis. Technische Universiteit Eindhoven.

Rozinat, A., 2016. *Data Quality Problems In Process Mining And What To Do About Them,* Eidhoven, Netherlands: Fluxicon.com - Process Mining for Professionals.

Rozinat, A., 2016. *Top 5 Data Quality Problems for Process Mining,* Eindhoven, The Netherlands: Fluxicon.

Rozinat, A. & Gunther, C., 2012. *Disco User Guide - Process Mining for Professionals,* Eindhoven, The Netherlands: Fluxicon.com.

Rozinat, A. & Van der Aalst, W. M. P., 2008. Conformance Checking of Processes based on Monitoring Real Behaviour. *Journal of Information Systems,* 33(1), p. 64–95.

Santoro, F. M., Rosa, M. L., Loos, P. & Pastor, O., 2016. *14th International Conference on Business Process Management..* [Online] Available at: http://bpm2016.uniriotec.br/ [Accessed 20 June 2017].

Sapna, C., Pridhi, A. & Pawan, B., 2013. Algorithm for Semantic Based Similarity Measure. *International Journal of Engineering Science Invention ,* 2(6), pp. 75-78 .

Schreiber, G., 2005. *OWL restrictions,* Amsterdam: department of Computer Science, VU University Amsterdam.

Semantic Web Primer, 2., 2012. *Introducing RDFS & OWL,* Cambridge, UK: Linked Data Tools.

Seng, J. L. & Kong, I., 2009. A schema and ontology-aided intelligent information integration. *Expert Systems with Applications,* 36(7), p. 10538 – 10550.

Sevarac, Z., 2006. *Neuro fuzzy reasoner for student modeling.* Los Alamitos, Poceedings of ICALT - IEEE, pp. 740- 744.

Sheth, A. et al., 2002. Managing semantic content for the web. *IEEE Internet Computing,* 6(4), pp. 80-87.

Shteiner, M., Bodaker, L. & Senderovich, A., 2016. *Heuristic Alpha+ Miner (HAM): Process Discovery Contest 2016,* Rio de Janeiro: Technical Report Submission for the Process Discovery Contest @ BPM 2016, [1st Edition], IEEE Task Force on Process Mining.

Sirin, E. & Parsia, B., 2004. *Pellet: An owl dl reasoner.* Whistler, British Columbia, Canada, Proceedings of the 2004 International Workshop on Description Logics (DL2004)', Vol. 104, CEUR-WS.org.

Smith, H. & Fingar, P., 2006. *Business Process Management: The Third Wave.* 1 ed. Tampa, Florida: Meghan-Kiffer Press.

Snae, C. & Br¨ueckner, M., 2007. Ontology-driven e-learning system based on roles and activities for thai learning environment. *Interdisciplinary Journal of E-Learning and Learning Objects 3, 1–17.,* 3(1), pp. 1-17.

Song, M., Günther, C. W. & van der Aalst, W. M. P., 2009. Trace Clustering in Process Mining. In: D. Ardagna, M. Mecella & J. (. Yang, eds. *BPM 2008. LNBIP.* Heidelberg,: Springer, p. 109–120.

Souhei, I., Shigeki, H. & Naoki, Y., 2008. *A Formal Ontology for Business Process Model TAP: Tasks-Agents-Products.* Amsterdam, The Netherlands, Proceedings of the 2008 conference on Information Modelling and Knowledge Bases XIXHannu Jaakkola, Yasushi Kiyoki, and Takahiro Tokuda (Eds.). IOS Press, pp. 290-297.

Stahlknecht, P. & Hasenkamp, U., 2005. *Einfuhrung in die wirtschaftsinformatic,* Heidelberg: Springer Heidelberg.

Tadlaoui, M., Chikh, A. & Bouamrane, K., 2013. ALEM: A Reference Model for Educational Adaptive Web Applications. In: A. Peña-Ayala, ed. *Intelligent and Adaptive Educational-Learning Systems: Achievements and Trends.* Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 25 - 48.

Tanasa, D. & Trousse, B., 2004. Advanced data preprocessing for intersites web usage mining.. *Intelligent Systems, IEEE,,* 19(2), pp. 59-65.

Thorburn, W. M., 1918. The Myth of Occam's razor. *Mind,* 27(1), p. 345–353.

Timms, M., 2001. *Predicting Students' Need For Help Using Pre-test Data.* San Antonio, Texas, proceeding of AIED, pp. 1-6.

Trcka, N. & Pechenizkiy, M., 2009. *From Local Patterns to Global Models: Towards Domain Driven Educational Process Mining.* s.l., IEEE Computer Society, pp. 1114-1119.

Trčka, N., Pechenizkiy, M. & van der Aalst, W. M. P., 2010. Process Mining from Educational Data. In: C. Romero, S. Ventura, M. Pechenizkiy & R. S. J. D. Baker, eds. *Handbook of Educational Data Mining.* Boca Raton, Florida: Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, CRC Press, pp. 123-142.

Tsarkov, D. & Horrocks, I., 2006. FaCT++ Description Logic Reasoner: System Description. In: F. U. & S. N., eds. *Automated Reasoning. IJCAR 2006. Lecture Notes in Computer Science, vol 4130.* Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 292-297.

Tukey, J. W., 1962. The Future of Data Analysis. *Annals of Mathematical Statistics,* 33(1), pp. 1-67.

Tynjala, P. & Hakkinen, P., 2005. Theoretical underpinnings and pedagogical challenges.. *Journal of Workplace Learning,* 17(5-6), pp. 318-336.

Van der Aalst, W. M. P., 2004. Business Process Management Demystified: A Tutorial on Models, Systems and Standards for workflow Management. In: J. R. W. a. R. G. Desel, ed. *Lectures on Concurrency and Petri Nets, volume 3098 of Lecture Notes in Computer Science.* Berlin: Springer, Berlin, pp. 1-65.

Van der Aalst, W. M. P., 2011. *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* 1 ed. Berlin: Springer.

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

Van der Aalst, W. M. P. et al., 2003. Workflow Mining: A Survey of Issues and Approaches. *Data and Knowledge Engineering,* 47(2), pp. 237-267.

Van der Aalst, W. M. P. & Van Hee, K. M., 2004. *Worflow Management: Models, Methods, and Systems.* 1 ed. Cambridge, MA: MIT Press.

Van der Aalst, W. M. P., van Hee, K., van Werf, J. M. & Verdonk, M., 2010. Auditing 2.0: Using Process Mining to Support Tomorrow's Auditor. *IEEE Computer,* 43(3), pp. 90-93.

Van der Aalst, W. M. P., Weijters, A. J. M. M. & Maruster, L., 2004. Workflow Mining: Discovering Process Models from Event Logs.. *International Journal of IEEE transactions on Knowledge and Data engineering,* 16(9), pp. 1128-1142.

Van der Aalst, W. M. P., Adriansyah, A., de Medeiros, A. K. A. & al, e., 2012. Process Mining Manifesto. In: D. F., B. K. & D. S., eds. *Business Process Management Workshops. BPM 2011. Lecture Notes in Business Information Processing.* Berlin: BPM Workshops LNBIP, vol. 99, pp. 169-194. Springer, 2012., pp. 169-194.

van Dongen, B., Claes, J., Burattin, A. & De Weerdt, J., 2016. *12th International Workshop on Business Process Intelligence 2016.* [Online] Available at: http://www.win.tue.nl/bpi/doku.php?id=2016:start#organizers [Accessed 20 June 2017].

Vassileva, D. & Bontchev, B. (., 2009. *Adaptation engine construction based on formal rules.* Lisboa, Portugal , Proceedings of the First International Conference on Computer Supported Education - Volume 1: CSEDU,, pp. 326-331.

Veiga, G. M. & Ferreira, D. R., 2010. Understanding Spaghetti Models with Sequence Clustering for ProM. In: S. Rinderle-Ma, S. Sadiq & F. (. Leymann, eds. *BPM 2009. LNBIP.* Heidelberg: Springer, p. 92–103.

Verbeek, E. & Mannhardt, F., 2016. *DrFurby Classifier: Process Discovery Contest @ BPM 2016,* Rio de Janeiro: Technical Report Submission for the Process Discovery Contest @ BPM 2016, [1st Edition], IEEE Task Force on Process Mining.

Verbeek, H., 2014. *Process Mining research tools and application.* [Online] Available at: http://www.processmining.org/promimport/start [Accessed 14 June 2016].

Verbeek, H., Buijs, J., van Dongen, B. & van der Aalst, W. M. P., 2011. XES, XESame, and ProM 6. In: P. E. (. Soffer P., ed. *Information Systems Evolution.* Springer, Berlin, Heidelberg: CAiSE Forum 2010. Lecture Notes in Business Information Processing, pp. 60-75.

W3C, 2004. *OWL Web Ontology Language.* [Online] Available at: http://www.w3.org/TR/owl-ref/ [Accessed 20 May 2017].

W3C, 2., 2004. *RDF Vocabulary Description Language 1.0: RDF Schema,* MIT: W3C Recommendation.

W3C, S. W., 2012. *Web Ontology Language (OWL),* Oxford, UK: OWL Working Group.

Wang, E. & Kim, Y. S., 2006. A Teaching Strategies Engine Using Translation from SWRL to Jess. In: E. Wang & Y. S. Kim, eds. *Intelligent Tutoring Systems. ITS 2006. Lecture Notes in Computer Science, vol 4053..* Berlin, Heidelberg: Proceedings of the 8th international conference on Intelligent Tutoring Systems, ITS'06, Springer-Verlag, Berlin, Heidelberg, pp. 51-60.

Weerdt, J. D., Backer, M. D., Vanthienen, J. & B., B., 2011. *A Robust F-measure for Evaluating Discovered Process Models.* Paris, France, In Proceedings of CIDM, 2011. IEEE, p. 148–155.

Weijters, A. J. M. M. & Ribeiro, J. T. S., 2010. *Flexible Heuristics Miner (FHM),* Eindhoven, the Netherlands: BETA Working Paper Series, WP 334, Eindhoven University of Technology, Eindhoven .

Weijters, A. J. M. M. & Van der Aalst, W. M. P., 2003. Rediscovering Workflow Models from Event-Based Data using Little Thimb. *Integrated Computer-Aided Engineering,* 10(2), pp. 151-162.

Weijters, A. J. M. M., Van der Aalst, W. M. P. & de Medeiros, A. K. A., 2006. *Process Mining with the Heuristics Miner-algorithm,* Eindhoven, the Netherlands: Technical report, EUT, Eindhoven, BETA working Paper Series, WP 166.

Weske, M., 2007. *Business Process Management: Concepts, Languages, Architectures.* 1 ed. Berlin: Springer Berlin.

Wimalasuriya, D. C. & Dou, D., 2010. Ontology-based information extraction: An introduction and a survey of current approaches. *Journal of Information Science,* 36(3), pp. 306-323.

Wimalasuriya, D. C. & Dou, D., 2010. Ontology-based information extraction: An introduction and a survey of current approaches. *Journal of Information Science,* 36(3), pp. 306-323.

Xiao, G., Calvanese, D., Cogrel, B. & al, e., 2017. *Ontop supported by Optique.* [Online] Available at: http://ontop.inf.unibz.it/ [Accessed 22 June 2017].

Yankova, M., Saggion, H. & Cunningham, H., 2008. *Semantic-based Identity Resolution and Merging for Business Intelligence,* Sheffield: University of Sheffield, UK.

Yarandi, M., 2013. *Semantic Rule-based Approach for Supporting Personalised Adaptive E-Learning,* London, UK: PHD Thesis, University of East London.

Zadeh, L., 1965. Fuzzy sets. , Information Science. *Information and Control,* 8(3), pp. 338-353.

Zadeh, L. A., 1971. Similarity relations and fuzzy orderings. *Information sciences,* 3(2), pp. 177-200.

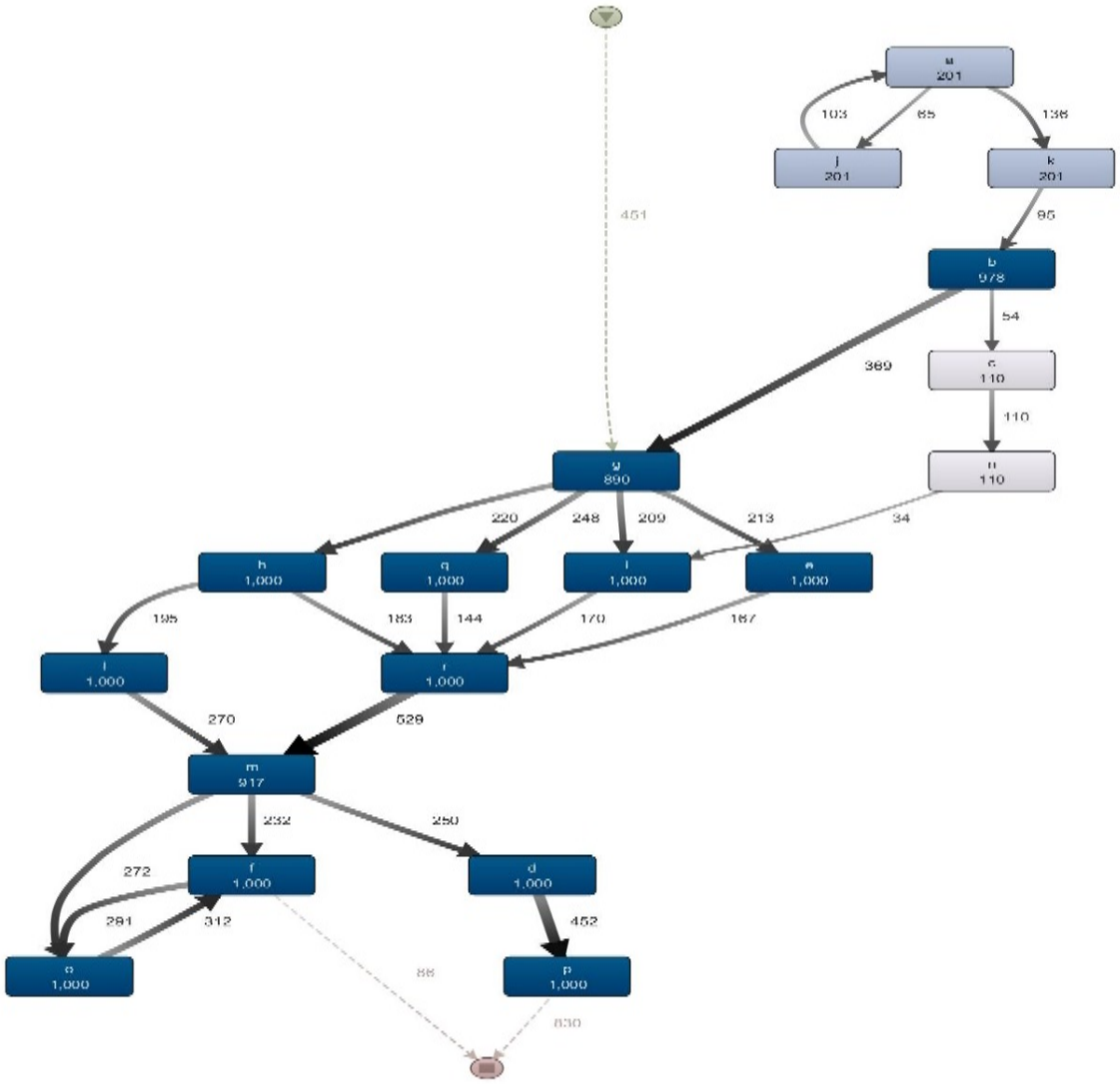
Zadeh, L. A., 1999. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems,* 100(1), pp. 9-34.

Appendix

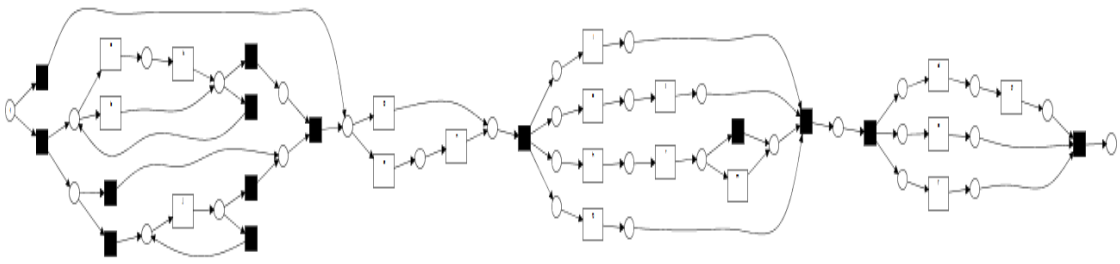
In this section of the thesis, the work includes other documents particularly examples of the discovered process models used for the work in this thesis.

A. Discovered Process Models for the Event Logs

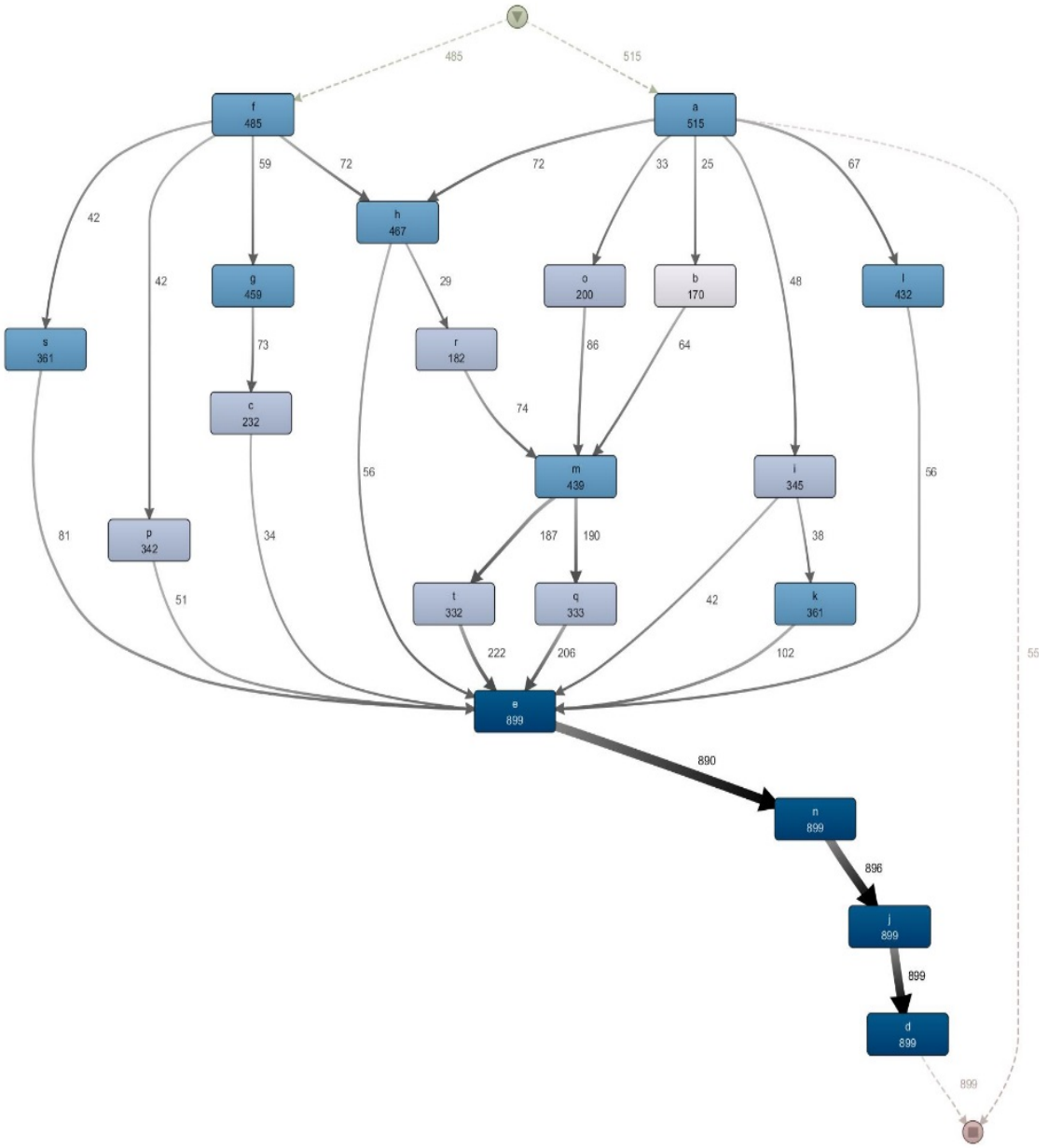
A.1. Fuzzy Models and Petri nets



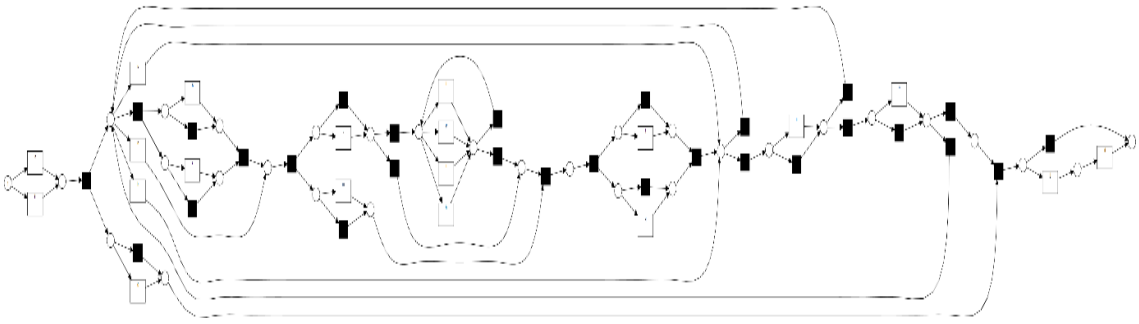
Appendix A.1.1 Fuzzy Model for training_log_1



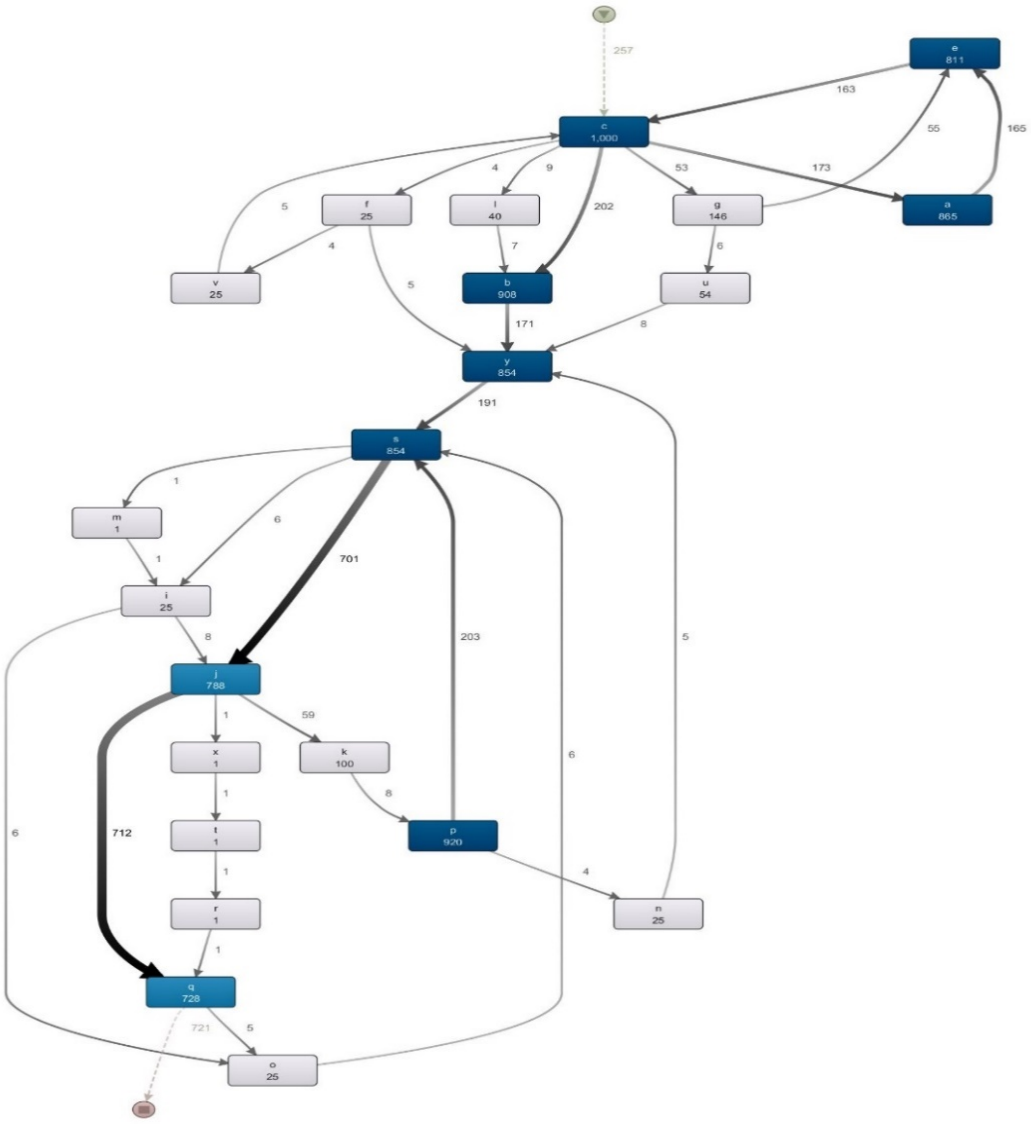
Appendix A.1.2 Petri net Model for training_log_1



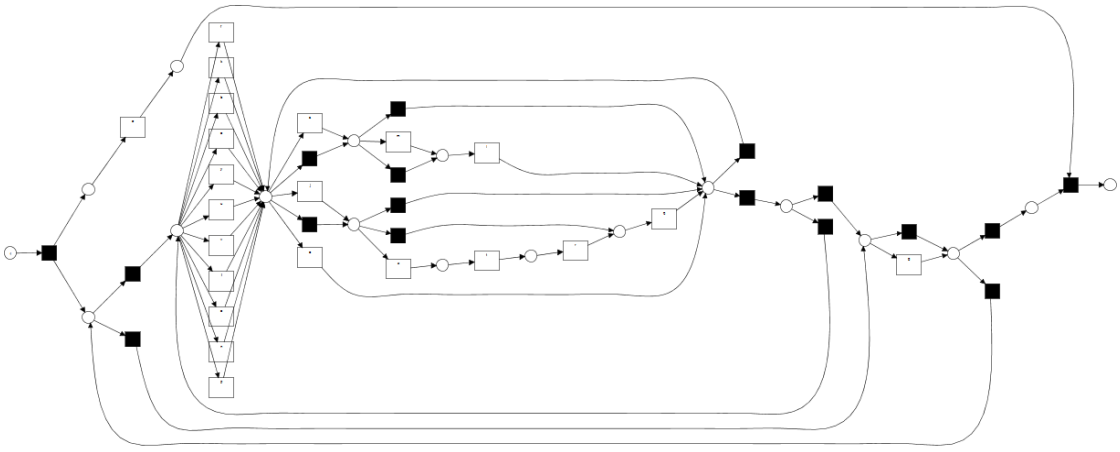
Appendix A.1.3 Fuzzy Model for training_log_2



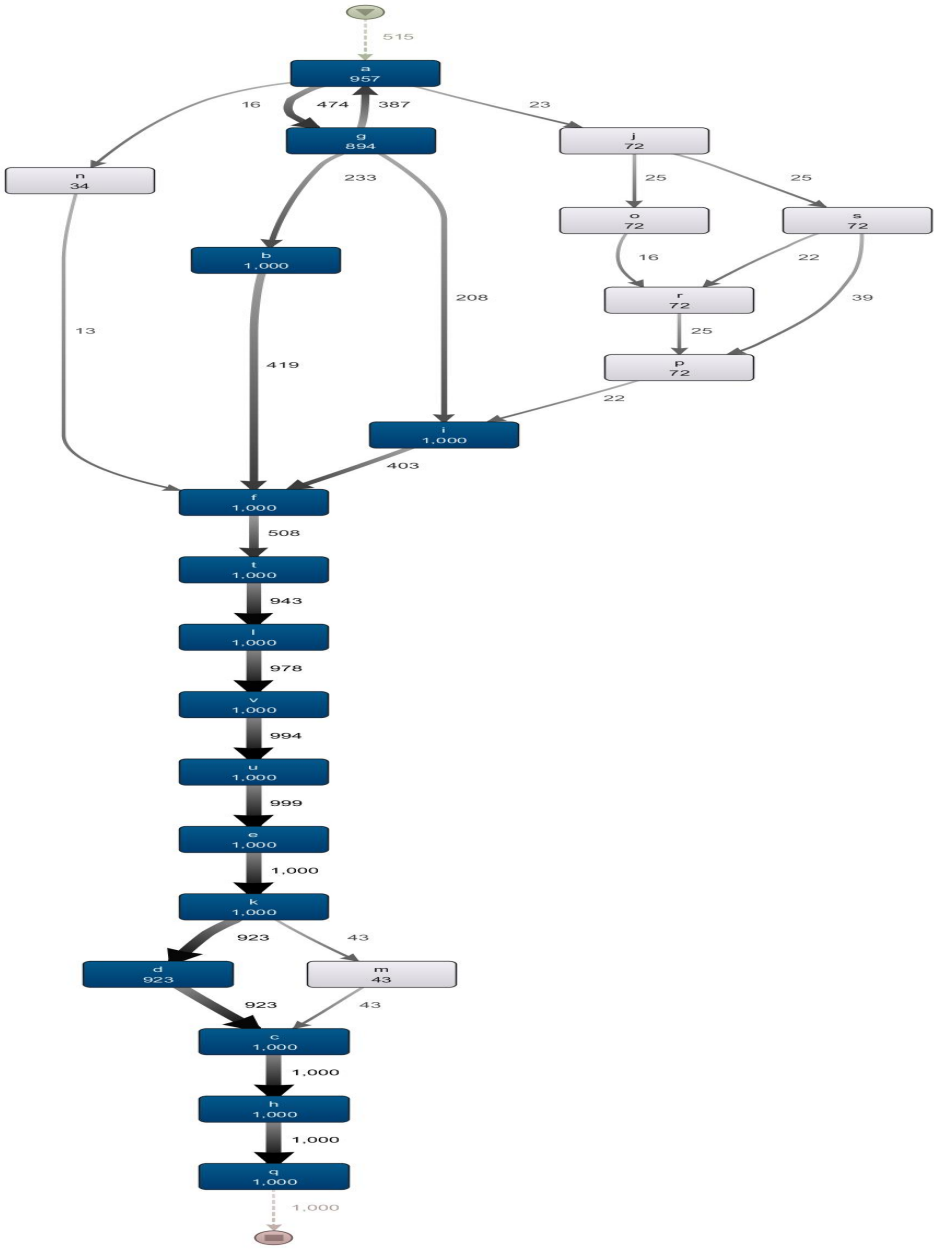
Appendix A.1.4 Petri net Model for training_log_2



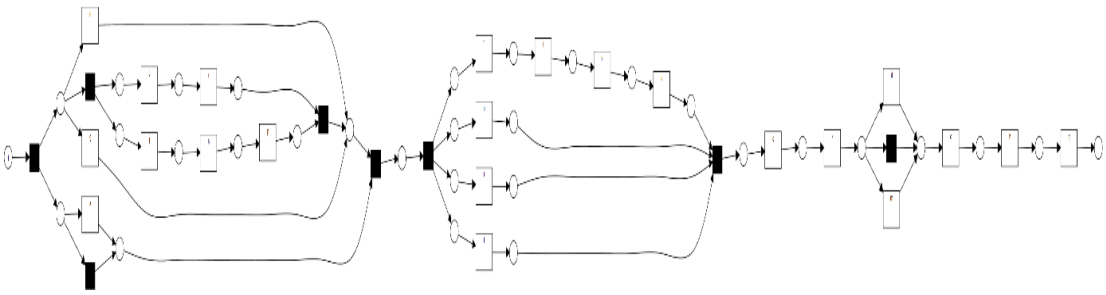
Appendix A.1.5 Fuzzy Model for training_log_3



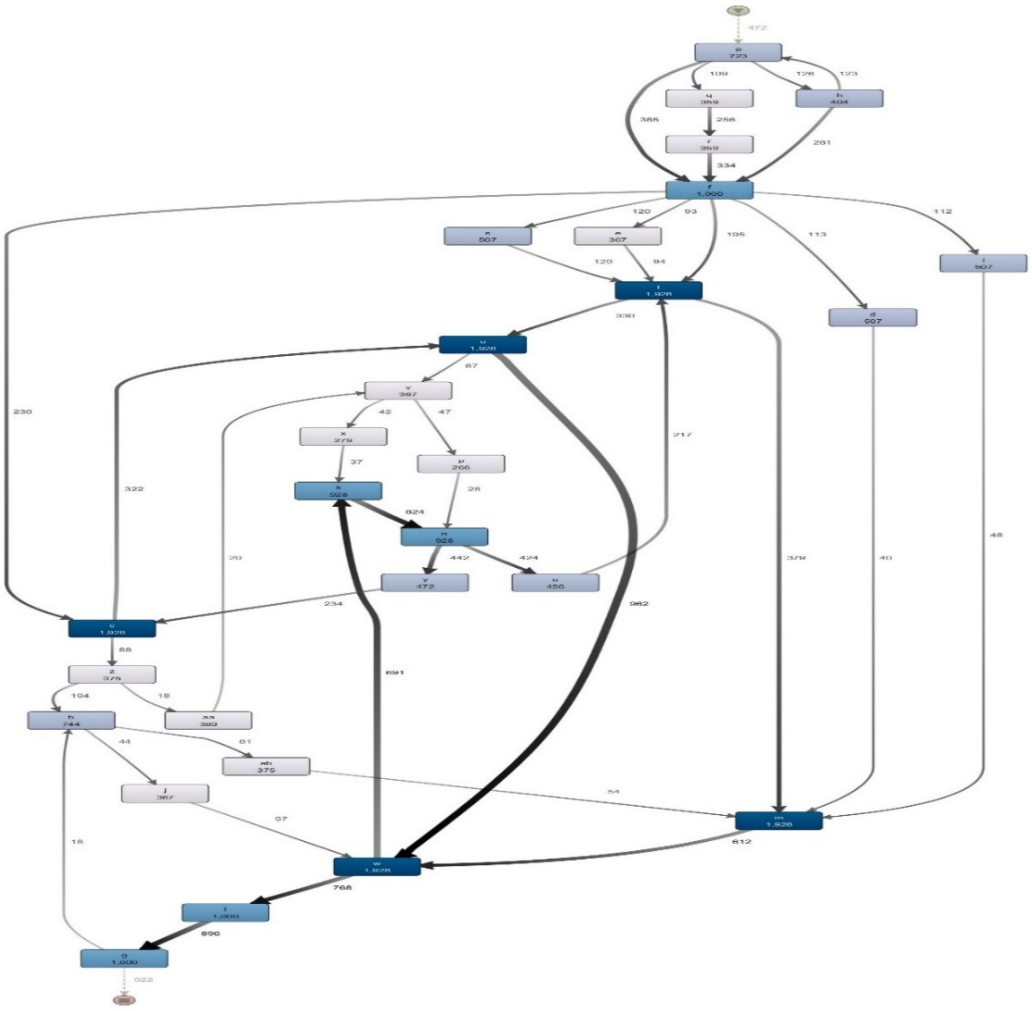
Appendix A.1.6 Petri net Model for training_log_3



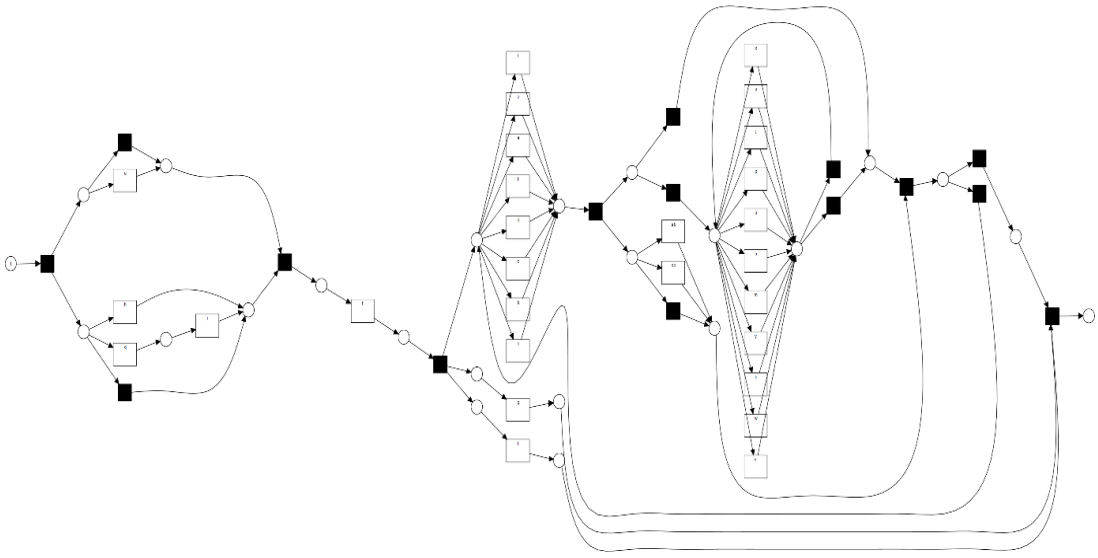
Appendix A.1.7 Fuzzy Model for training_log_4



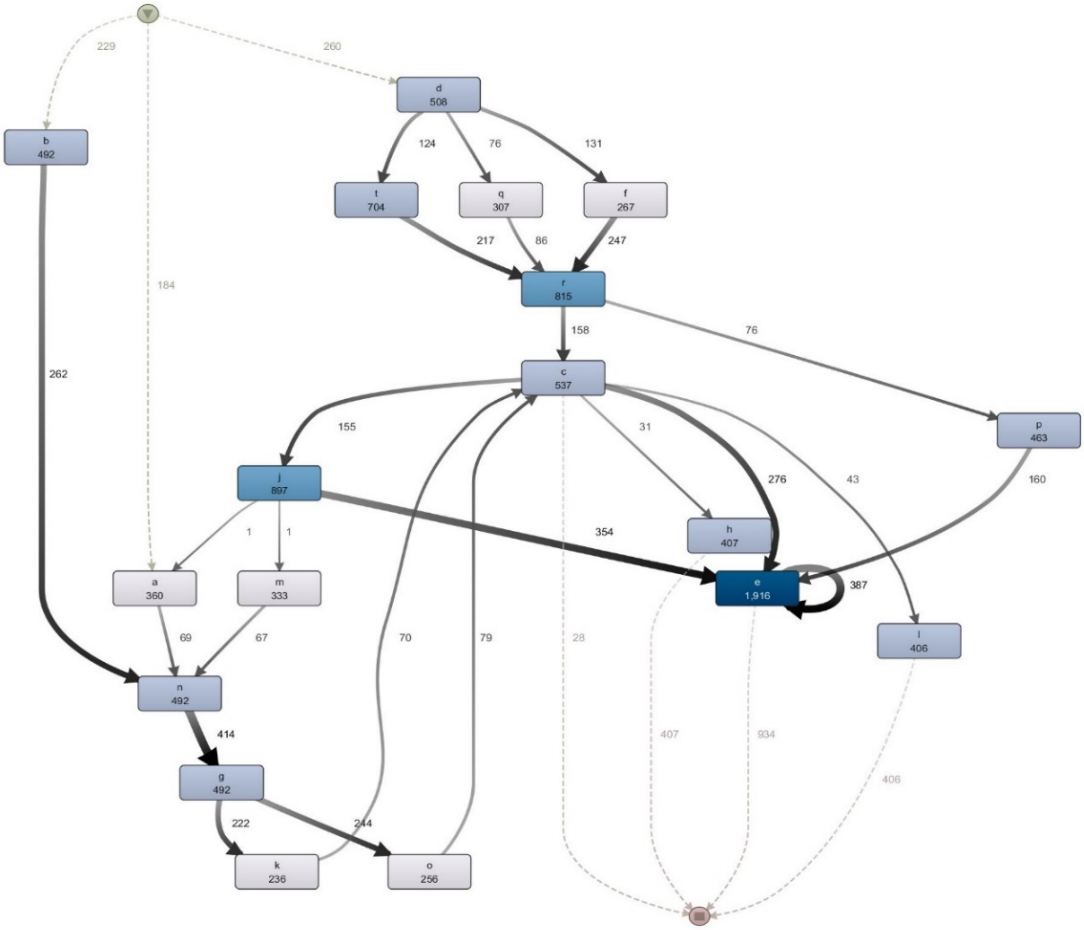
Appendix A.1.8 Petri net Model for training_log_4



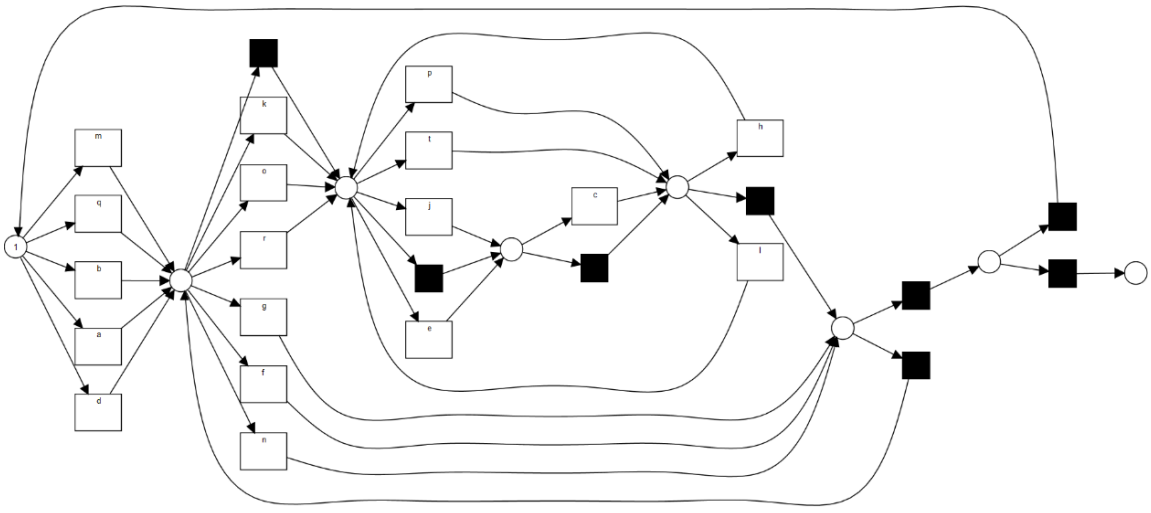
Appendix A.1.9 Fuzzy Model for training_log_5



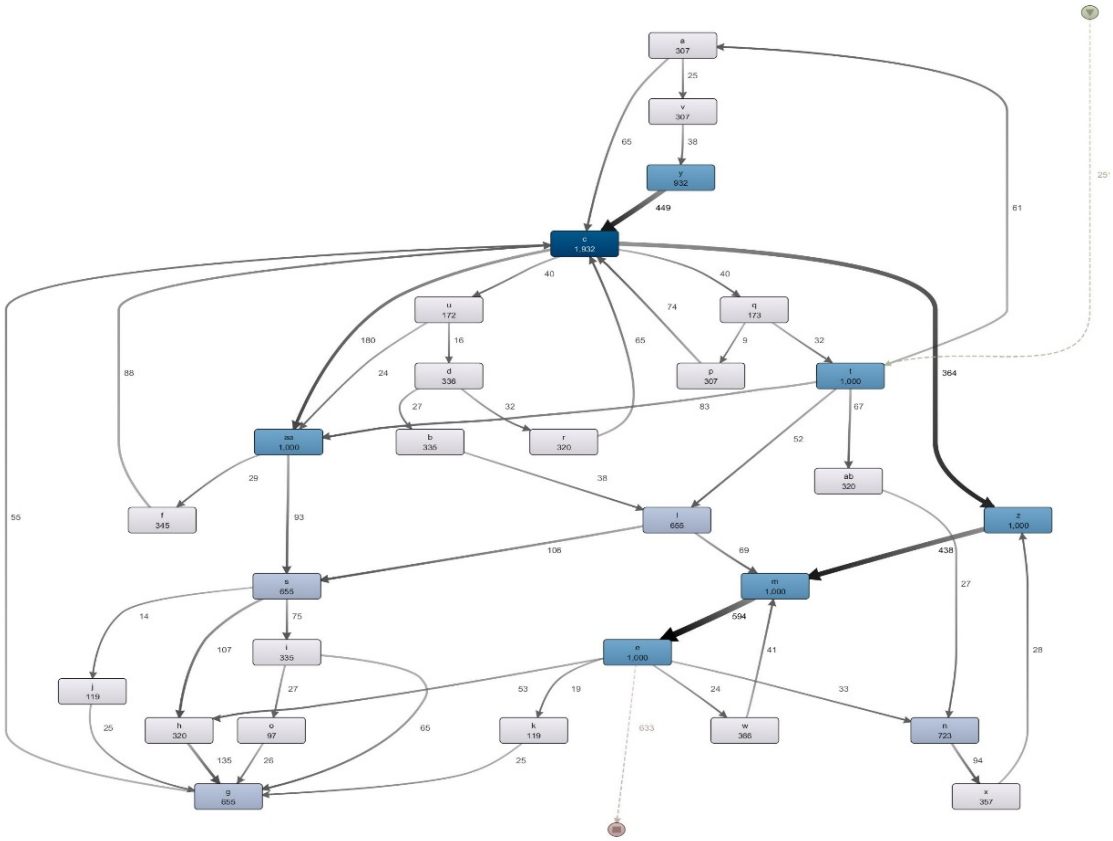
Appendix A.1.10 Petri net Model for training_log_5



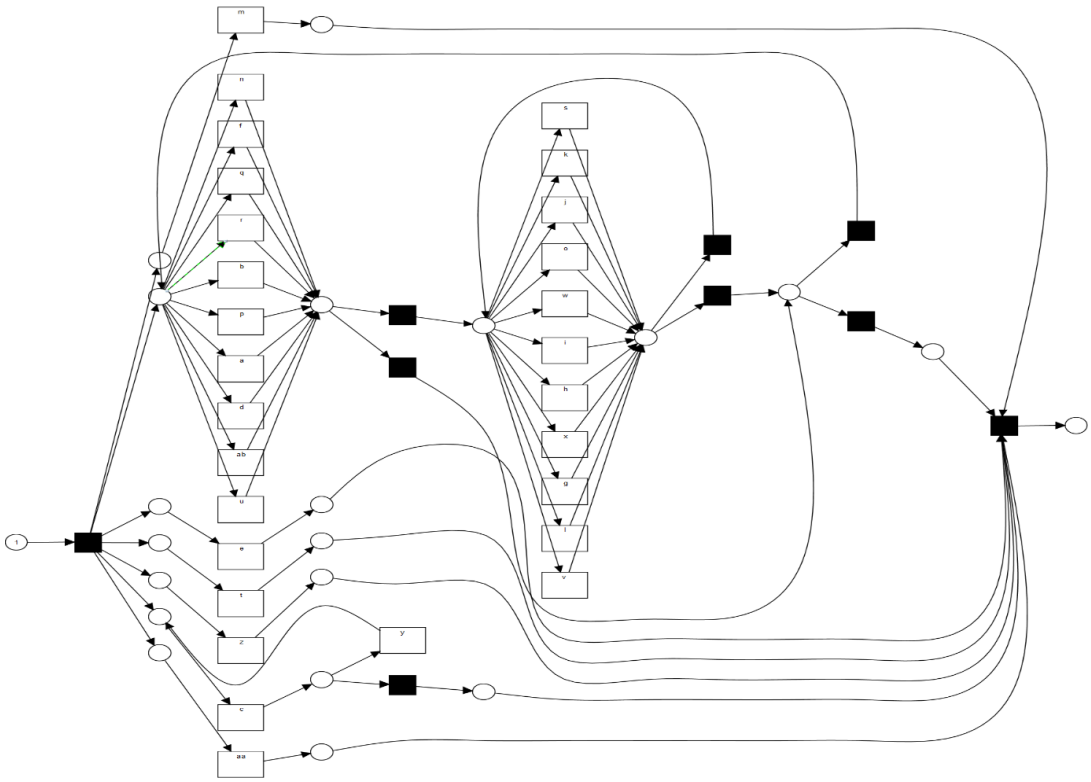
Appendix A.1.11 Fuzzy Model for training_log_6



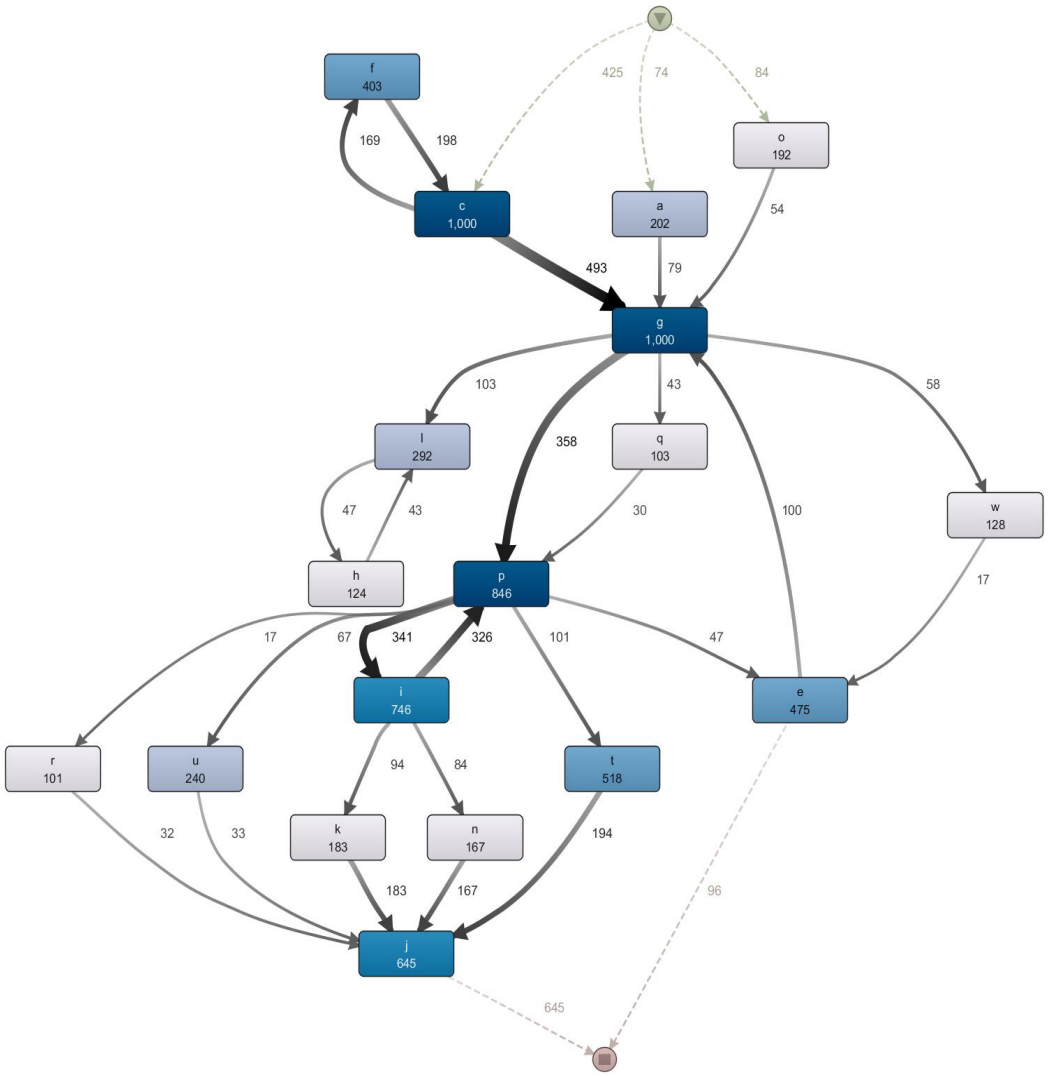
Appendix A.1.12 Petri net Model for training_log_6



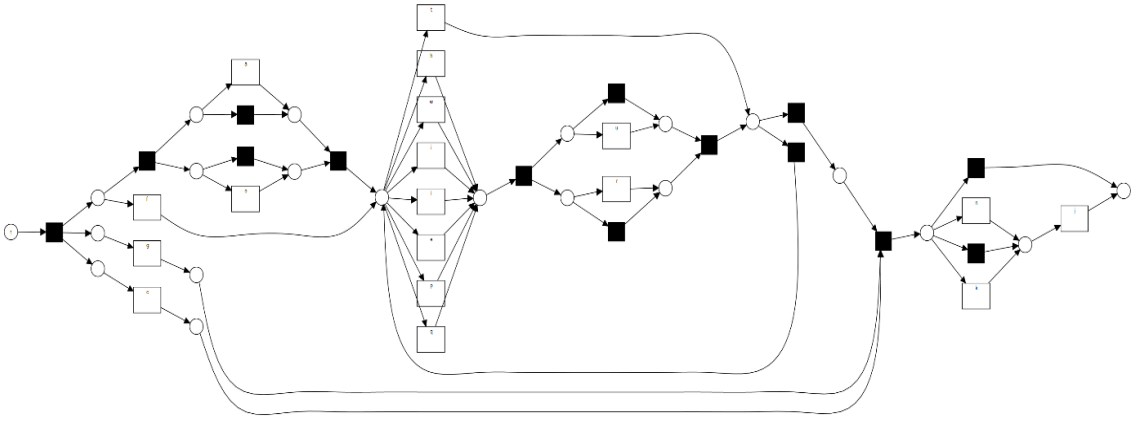
Appendix A.1.13 Fuzzy Model for training_log_7



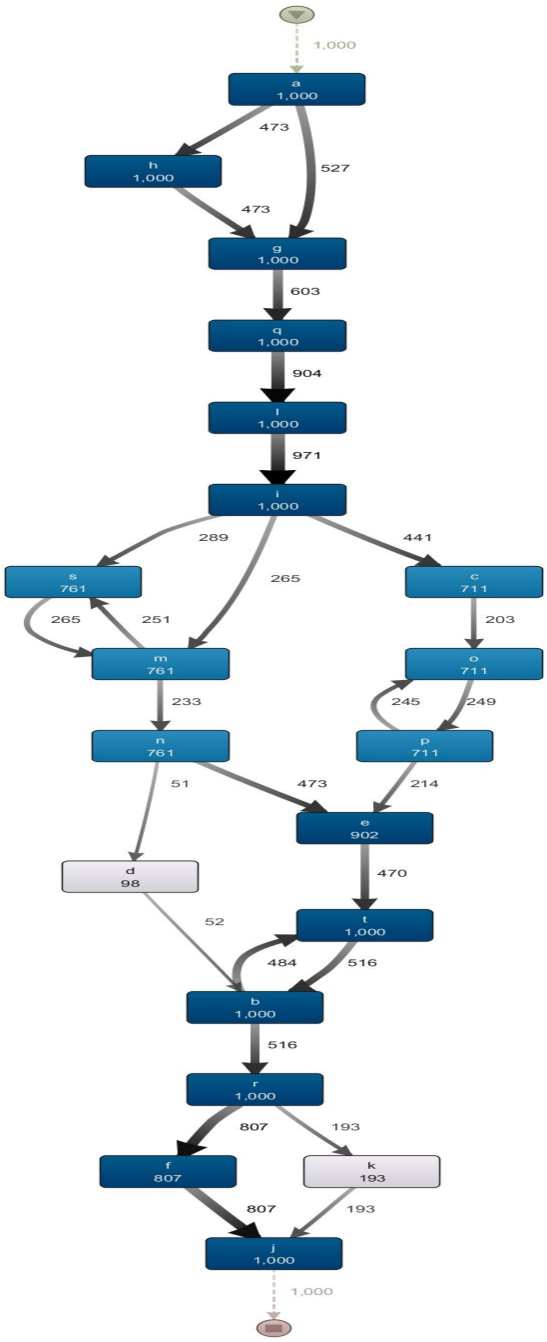
Appendix A.1.14 Petri net Model for training_log_7



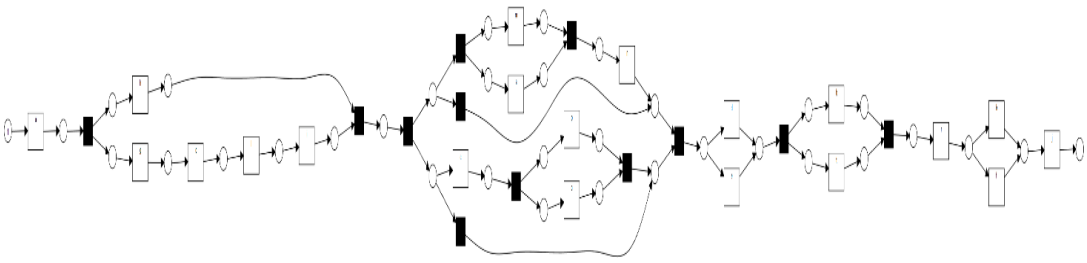
Appendix A.1.15 Fuzzy Model for training_log_8



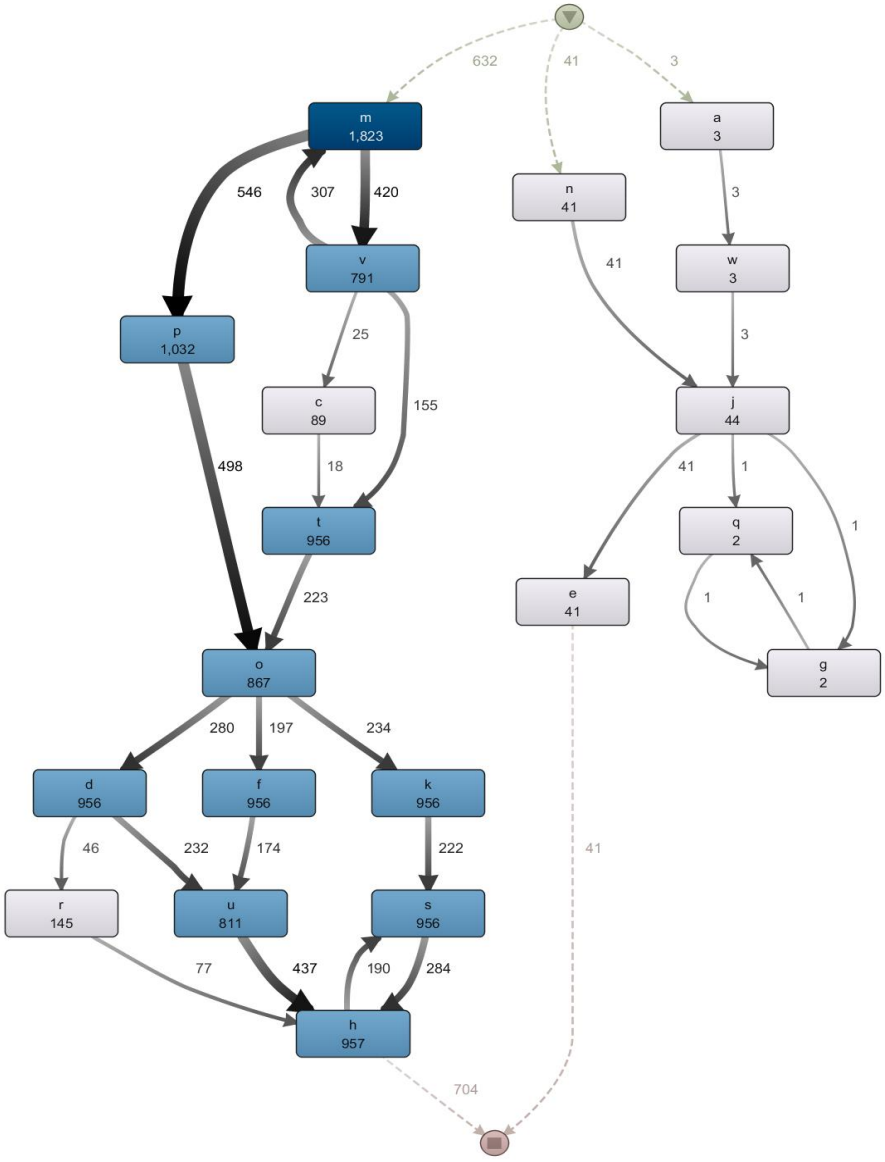
Appendix A.1.16 Petri net Model for *training_log_8*



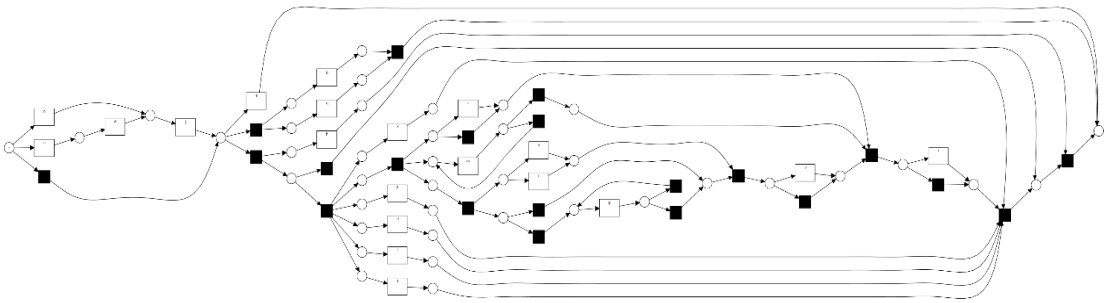
Appendix A.1.17 Fuzzy Model for training_log_9



Appendix A.1.18 Petri net Model for training_log_9

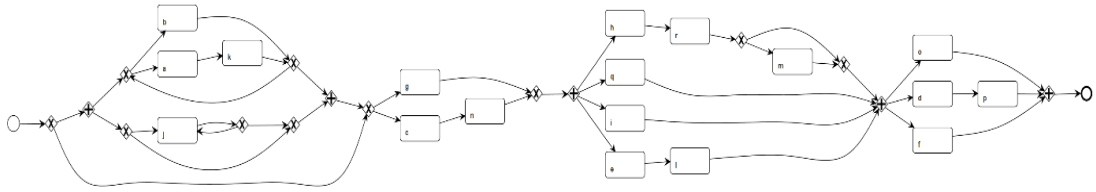


Appendix A.1.19 Fuzzy Model for training_log_10

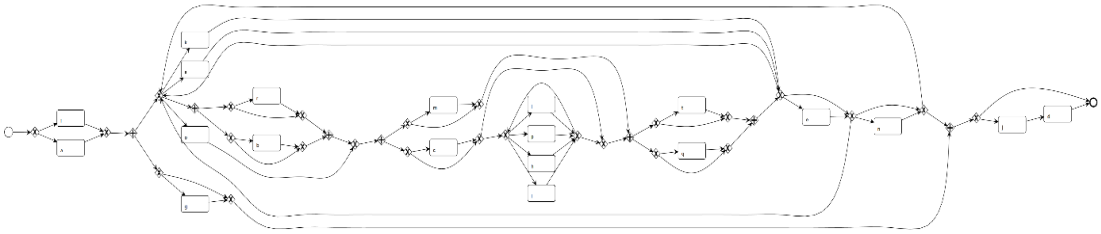


Appendix A.1.20 Petri net Model for training_log_10

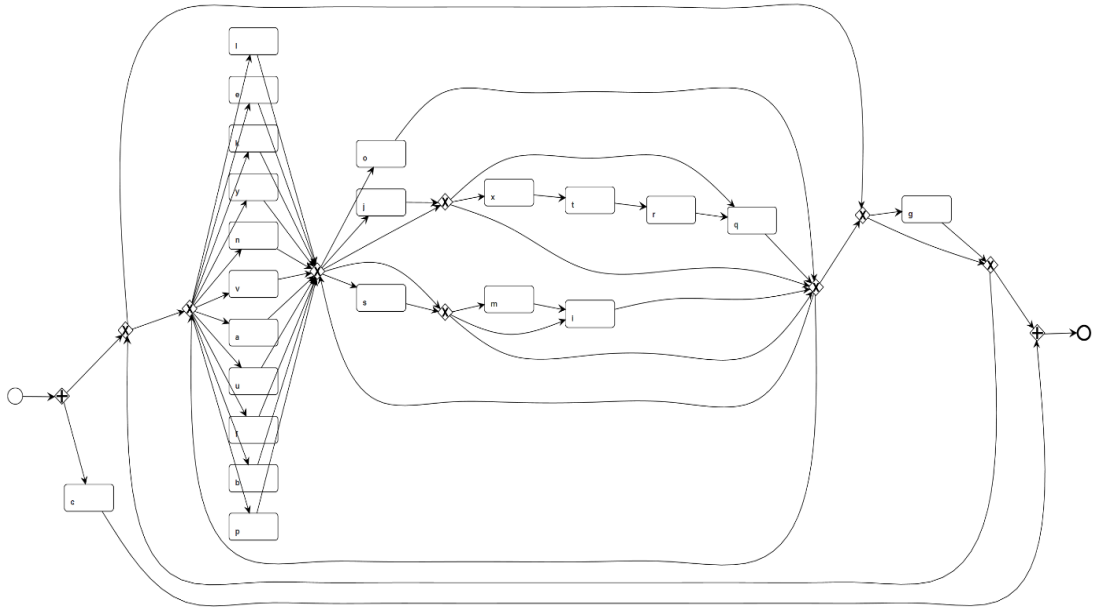
A.2 BPMN Models for the Training Logs



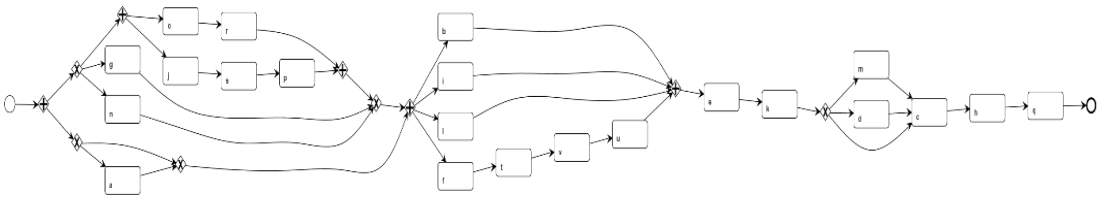
Appendix A.2.1 BPMN model for training_log_1



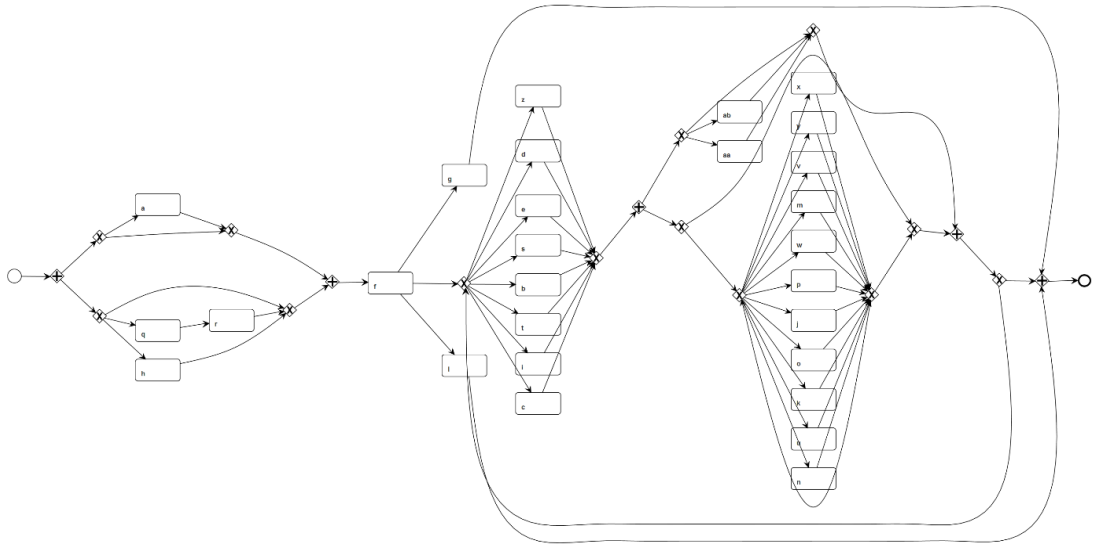
Appendix A.2.2 BPMN model for training_log_2



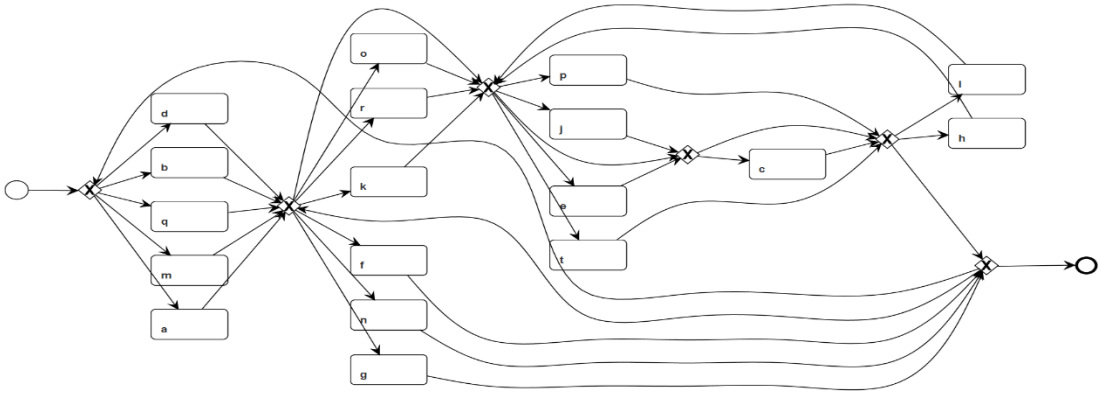
Appendix A.2.3 BPMN model for training_log_3



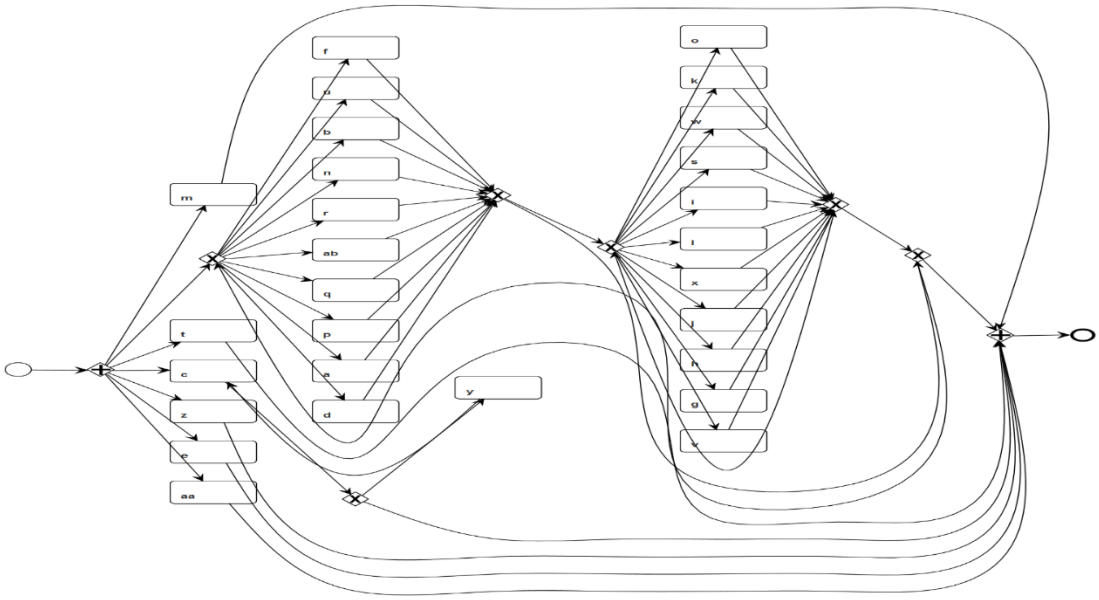
Appendix A.2.4 BPMN model for training_log_4



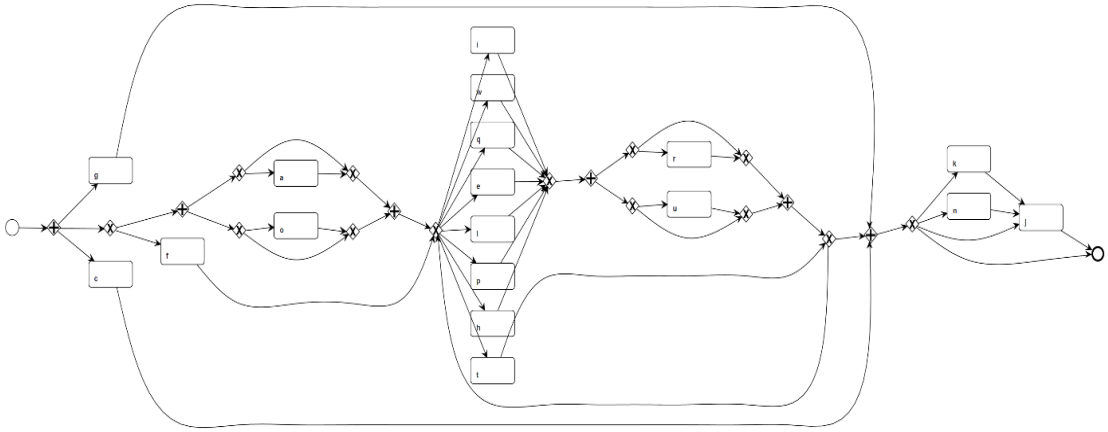
Appendix A.2.5 BPMN model for training_log_5



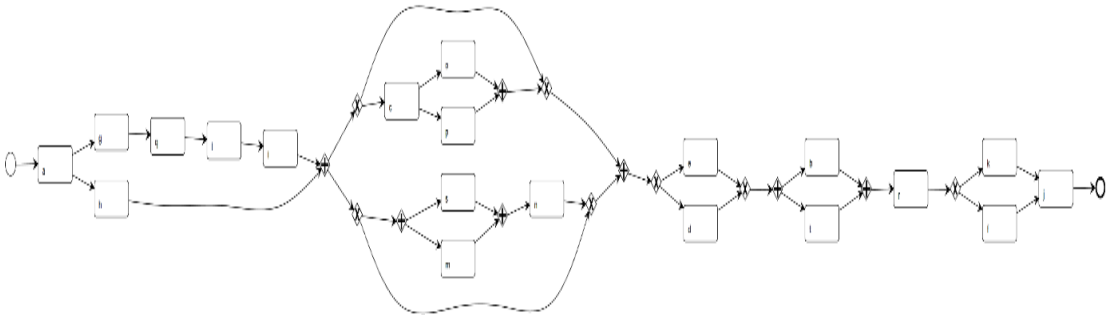
Appendix A.2.6 BPMN model for training_log_6



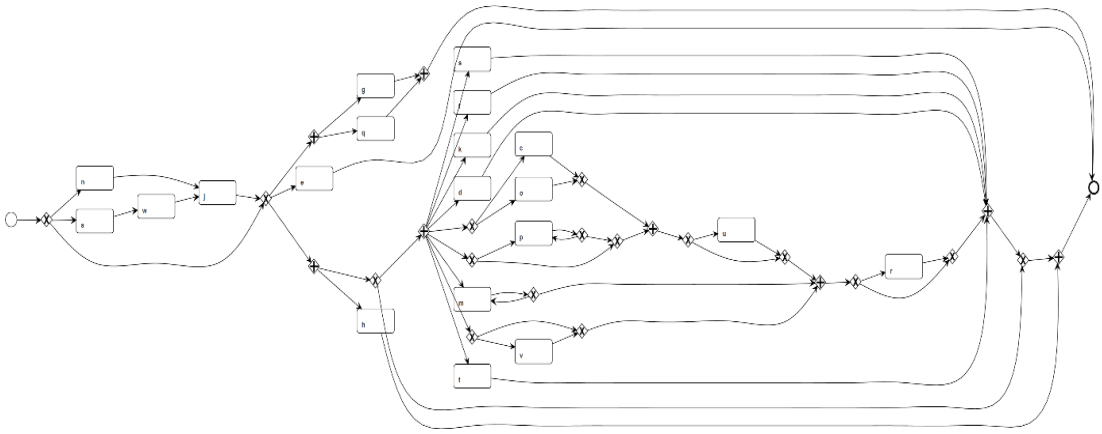
Appendix A.2.7 BPMN model for training_log_7



Appendix A.2.8 BPMN model for training_log_8



Appendix A.2.9 BPMN model for training_log_9



Appendix A.2.10 BPMN model for training_log_10