A METHODOLOGY TO DEVELOP PROCESS ONTOLOGY FROM ORGANIZATIONAL GUIDELINES WRITTEN IN NATURAL LANGUAGE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

ÖZGE GÜRBÜZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

JUNE 2017

A METHODOLOGY TO DEVELOP PROCESS ONTOLOGY FROM ORGANIZATIONAL GUIDELINES WRITTEN IN NATURAL LANGUAGE

Submitted by ÖZGE GÜRBÜZ in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in the Department of Information Systems Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin _______________
Director, **Graduate School of Informatics**

Prof. Dr. Yasemin Yardımcı Çetin _______________
Head of Department, **Information Systems**

Prof. Dr. Onur Demirörs _______________
Supervisor, **Information Systems**

Examining Committee Members:

Assoc. Prof. Dr. Altan Koçyiğit _______________
Information Systems, METU

Prof. Dr. Onur Demirörs _______________
Information Systems, METU

Assoc. Prof. Dr. Pınar Karagöz _______________
Computer Engineering, METU

Assist. Prof. Dr. Ömer Özgür Tanrıöver _______________
Computer Engineering, Ankara University

Assist. Prof. Dr. Murat Yılmaz _______________
Computer Engineering, Çankaya University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Özge Gürbüz

Signature :

ABSTRACT

A METHODOLOGY TO DEVELOP PROCESS ONTOLOGY FROM ORGANIZATIONAL GUIDELINES WRITTEN IN NATURAL LANGUAGE

Gürbüz, Özge
Ph.D., Department of Information Systems
Supervisor: Prof. Dr. Onur Demirörs

June 2017, 111 pages

Integrating ontologies with process modeling improves data representations and makes it easier to query, store and reuse processes at the semantics level. Therefore, in recent years, this topic has become increasingly popular. The studies in the literature have proposed methods for the integration process either to relate domain ontologies to process models or to transform process models to process ontologies. Another way to establish the integration between ontologies and process models is to develop process ontologies from organizational sources. Since most organizations have guidelines in natural language, it requires significant amount of time and effort to extract the roles, activities, information carriers, business rules, and relationships for process ontology development. In this thesis, a new Process Ontology Population (PrOnPo) methodology and tool is proposed that will automatically develop process ontology by extracting process information from organizational guidelines (regulations, procedures, directives and policies written in natural language). This approach will not only minimize the effort and time required for process ontology development, will also address the natural language ambiguity and provide an input for process modeling, hence improve the semantic quality of the business process models and their consistency with process ontology. As a part of this thesis work, two exploratory studies, a multiple case study and an experiment were performed in order to explore, generalize and validate the proposed approach.

Keywords: Ontology Development, Process Ontology, Natural Language Parsing, Business Process Modeling.

ÖZ

DOĞAL DİLDE YAZILMIŞ ORGANİZASYON YÖNETMELİKLERİNDEN SÜREÇ ONTOLOJİSİ OLUŞTURMAK İÇİN BİR METODOLOJİ

Gürbüz, Özge
Doktora, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Prof. Dr. Onur Demirörs

Haziran 2017, 111 sayfa

Ontolojilerin süreç modelleri ile bütünleştirilmesi, veri gösterimlerini geliştirir ve semantik düzeyde sorgulama, depolama ve yeniden kullanımı kolaylaştırır. Bu nedenle, son yıllarda bu konu giderek popüler hale gelmiştir. Literatürdeki çalışmalar, alan ontolojilerini süreç modelleriyle ilişkilendirmek veya süreç modellerini ontolojilere dönüştürmek için yöntemler önermiştir. Ontolojiler ve süreç modelleri arasındaki ilişkiyi kurmanın diğer bir yolu ise, süreç ontolojilerini organizasyon yönetmeliklerinden geliştirmektir. Organizasyonlar, doğal dilde yazılmış yönetmeliklere sahip olmalarından dolayı, süreç ontolojisi geliştirmek için roller, aktiviteler, bilgi taşıyıcıları, iş kuralları ve bunların arasındaki ilişkilerin çıkarılması, zaman ve çaba gerektirmektedir. Bu tez çalışmasında, Süreç Ontolojisi Popülasyonu (PrOnPo) metodolojisi ve aracı önerilmiştir. Bu araç sayesinde, otomatik olarak süreç bilgileri, organizasyonun doğal dil ile yazılmış yönetmeliklerinden (yönetmelikler, prosedürler, direktifler) çıkartılarak süreç ontolojisi geliştirilebilecektir. Bu yaklaşım, yalnızca süreç ontolojisi oluşturma için gerekli çaba ve zamanı en aza indirmekle kalmayıp, aynı zamanda doğal dil belirsizliğini de ele alıp, süreç modellemesi için bir girdi sağlayacak ve böylece iş süreç modellerinin semantik kalitesini ve süreç ontolojisi ile tutarlılığını geliştirecektir. Bu tez çalışmasının bir parçası olarak, önerilen yaklaşımı keşfetmek, genellemek ve doğrulamak için iki keşif çalışması, çoklu vaka incelemesi ve bir deney gerçekleştirilmiştir.

Anahtar Kelimeler: Ontoloji Geliştirme, Süreç Ontolojisi, Doğal Dil İşleme, İş Süreç Modelleme

to my beloved family

ACKNOWLEDGEMENTS

First of all, I express my sincere gratitude to my supervisor Prof. Dr. Onur Demirörs. I am thankful to him for his continuous guidance, support, patience and encouragements throughout my study.

I am grateful to Assoc. Prof. Dr. Pınar Karagöz, Assoc. Prof. Dr. Altan Koçyiğit, Asst. Prof. Dr. Murat Yılmaz and Asst. Prof. Dr. Ömer Özgür Tanrıöver for their valuable advices.

My sincere thanks go to my father Abdullah Gürbüz, my mother Sema Gürbüz and my sister Ayten Gürbüz for their endless patience, encouragement and support throughout my educational pursuit. I am really happy and fortunate to be your daughter. Additionally, I would like to thank to my grandparents for their endless love.

This thesis would not have been possible without the support I received from my colleagues and friends; Ahmet Coşkunçay, Murat Salmanoğlu, Ali Yıldız, Gökçen Yılmaz, Serhat Peker, Özden Özcan Top, Duygu Fındık Coşkunçay, Nurcan Alkış, Ayşegül Özkaya Eren, Elif Aleçakır, Rahime Belen Sağlam, Berna Bakır Batu, Şeyma Çavdar, Okan Bilge Özdemir and Mehmet Can Fal. Thank you for your contributions in this thesis and friendship, I am grateful forever. Additionally, I cannot forget Esin Gül Ölmez and Ebru Gündede's supports to motivate me with their unconditional friendships. Lastly, I would like to thank to Gökhan İşbitiren, for his support and encouragement. Thank you for believing in me and being here in my hardest time.

Also I am thankful to Sibel Gülnar, Sibel Sel and Hakan Güler for their helps in every stage of the bureaucratic tasks.

I would like to thank to The Scientific and Technological Research Council of Turkey (TÜBİTAK) for their financial support by graduate student scholarship (grant no. 2211) during my studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BPEL	Business Process Execution Language
BPMN	Business Process Model and Notation
BPMO	Business Process Modeling Ontology
CRCTOL	Concept-Relation-Concept Tuple-based Ontology learning system
DAML+OIL	DARPA Agent Markup Language and Ontology Interchange Language
eEPC	Extended Event-Driven Process Chain
EMF	Eclipse Modeling Framework
EPC	Event-Driven Process Chain
EPF	Eclipse Process Framework
Full-PrOnPo	Full Automated PrOnPo
GMF	Graphical Modeling Framework
METU	Middle East Technical University
MPM	Manual Process Modeling
MPOD	Manual Process Ontology Development
MS	Microsoft
OWL	Web Ontology Language
PrOnPo	Process Ontology Population
ProModGen	Process Model Generation
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
Semi-PrOnPo	Semi-Automated PrOnPo
SUPER	Semantics Utilized for Process Management within and between Enterprises
UML	Unified Modeling Language
UPON	Unified Process for Ontology Building
UPROM	Unified Business Process Modeling

CHAPTER 1

INTRODUCTION

Ontology is defined as, "formal specification of a shared conceptualization" (Gruber 1993). Ontologies improve communication between people, enhance reuse of domain knowledge, make domain assumptions explicit and differentiate domain and operational knowledge (Gruber 1993). Business process models are a sequential representation of all activities associated with a specific business goal. Business process modeling is used for a variety of purposes such as establishing execution consistency, optimization, automation, measurement, and certification. Integrating ontologies with business process models and creating process ontologies that will allow process information to be queried at high-level abstractions interlink the process information with business dictionaries and improve the explicit representation of process information and the consistency between external process descriptions (Haller et al. 2008). Process ontologies can incorporate reusable and comprehensive knowledge about an organization's processes, activities, roles, application systems, business rules, process interfaces, inputs / outputs, and the relationship between them.

In this thesis, Process Ontology Population (PrOnPo) methodology and tool for developing process ontology from organizational guidelines is proposed. The remainder of this chapter presents the background of the problem, purpose, contribution and significance, research strategy and the structure of the study.

1.1 Background of the Problem

Process ontology is defined as the specification of the classes and their relations representing information included in the processes and formal representation of processes. Process ontologies integrated with process models enrich data representations and increase the quality of process models (Fan et al. 2016; Belhajjame & Brambilla 2009; Alkhaldi et al. 2015). Therefore, in recent years, integrating ontologies with process models has become a popular topic with the studies in the literature being classified into three categories:

The first category includes studies that propose methods to integrate existing domain ontologies with process models (Fan et al. 2016; Francescomarino et al. 2014; Cherfi et al. 2012; Thomas & Fellmann 2009; Dimitrov et al. 2007). However, although using existing domain ontologies improves the efficiency, the formal representation of processes is limited to the knowledge that exists in the domain ontologies. The second category comprises studies that transform process models to ontologies (Leopold et al. 2015; De Cesare et al. 2014; Eisenbarth 2013; Haller et al. 2008; Höfferer 2007). In most cases, automation is possible but the process ontologies only cover the knowledge represented in the process models. The third category has only one study which targets building process models and domain ontologies together (Coskuncay et al. 2016). This study focuses on minimizing the double effort for ontology development and process modeling. In this approach, the scope of formal representation of the process models and processes is dependent on the extracted process information.

Briefly, the studies in the literature provide benefits for using ontologies for process modeling or transforming process models to process ontologies; however, they address neither the scope nor the development of process ontology. Developing more comprehensive process ontologies from scratch requires extracting activities, roles, documents, and other concepts related to the process activities mainly by conducting interviews (semi-structured sources), interpreting the organizational guidelines (unstructured sources), or using the event logs of the information system (semi-structured sources). As these activities are knowledge-intensive, building process ontologies requires significant time and effort. Process mining can be used to address the time and effort issue, in which process discovery techniques are utilized to query, create, enhance and conform process models. However, since 85% of the process information is stored in unstructured (natural language) text (Blumberg & Atre 2003), process mining techniques from event logs (Akman & Demirörs 2009; Schönig 2016) are difficult to use. Therefore, information defined in natural language texts is considered to be the most challenging sources for process discovery.

In contrast to using event logs, there are also process discovery techniques which propose methods for process modelling from text (Friedrich et al. 2011; Ghose et al. 2013; Goncalves et al. 2009). These methods do not involve ontologies but generate process models from text such as textual descriptions of process models, group stories, emails, reports, manuals and use cases by natural language parsers. In addition to these sources, organizational guidelines (regulations, procedures, policies, etc.) are other primary unstructured sources, which contain process information. Furthermore, guidelines are another type of pure natural language text but consist of a larger amount of information; thus, they are hard to interpret. Therefore, when text in pure natural language is concerned, there are challenges regarding natural language ambiguity, inconsistency, and identification of synonyms. On the other hand, using natural language parser alone will not be efficient in retrieving process information from guidelines and thus requires an extension in the algorithm. Furthermore, a natural language parser may not produce the desired results for all languages.

To sum up, existing process discovery techniques using natural language text do not efficiently address challenges that arise when using guidelines. These challenges can be overcome by integrating ontologies with extracted process information as shown in previous studies that improved the quality and consistency of process models by integrating them with ontologies and semantic annotations (Fan et al. 2016; Alkhaldi et al. 2015; Francescomarino et al. 2014). On the other hand, ontology learning techniques are intended to extract terms, concepts and relations to build domain ontologies and will not extract entities regarding the process information (Wong et al. 2012). Consequently, there is a need for a new efficient method to extract process information from guidelines written in natural language text and develop process ontologies to reduce the required time and effort and address the challenges arising from the use of natural language text (ambiguity, inconsistency, and synonyms). The process ontology developed with this way can later be used for process modeling, discovery, querying and reusing.

1.2 Purpose of the Study

This study aimed to develop an efficient methodology and tool for process ontology development. The proposed methodology presents how to develop process ontology using organizational guidelines and the tool automates the process of ontology development.

Ontology development methodologies usually consist of four distinct phases: the specification phase in which the scope of the process ontology to be developed is identified, the acquisition phase in which the process knowledge is extracted from the guidelines, the evaluation phase in which the extracted information is evaluated, and finally the conceptualization phase in which the evaluated knowledge is transformed into ontology classes and instances. The PrOnPo tool aims to automate the acquisition and conceptualization phases which are the most time-consuming phases in ontology development. The tool automatically extracts process information (activity, role, information carrier, etc.) from the given guideline and conceptualizes process information as ontology instances. Consequently, process information in the guidelines is transformed into formally specified process entities which are instances of process ontology.

The developed process ontology can be used not only for process querying and creating a shared understanding of the processes of an organization, but also for process modeling. In addition to the main purpose of the study, conceptualized process entities are transformed to process model elements to demonstrate the use of process ontologies for process discovery.

Briefly, the goals of this study are to implement a methodology and tool for building process ontologies from organizational guidelines written in natural language and to implement the transformation of process entities to process models. Achieving these goals will reduce the time and effort required for process ontology development and modeling. Additionally, the scope of the formal specification of processes produced by the PrOnPo tool will not be limited to process models since process information which is not represented in the models will be stored in process ontologies. The developed process ontologies will simplify process querying and create consistency within the organization.

1.3 Research Strategy

To achieve the goals specified above, a four-stage research strategy was developed (Figure 1), which comprised two exploratory studies, a multiple case study, and an experiment to explore, generalize and validate the proposed methodology and tool, respectively.

In the first stage, the first exploratory study was performed to explore alternative methods for process ontology development. Process ontology was manually developed from a selected case to identify the requirements for automation (PrOnPo tool). Then, Turkish and English guidelines are used to compare the language impact on automation. In the second stage, another exploratory study is undertaken to apply the proposed tool automation (PrOnPo) on a different case. Precision and recall metrics were used to investigate the effectiveness of the tool in the acquisition phase. The time and effort to measure the efficiency of the tool were recorded. In the third stage, the use of the PrOnPo tool was generalized through a multiple case study adapting it to different types of guidelines from different domains and analyzing its effectiveness based on precision and recall metrics. Lastly, in the fourth stage, the efficiency of the PrOnPo tool was validated by performing an experiment, in which the time and effort required for process ontology development and process modeling were compared between three groups of participants.

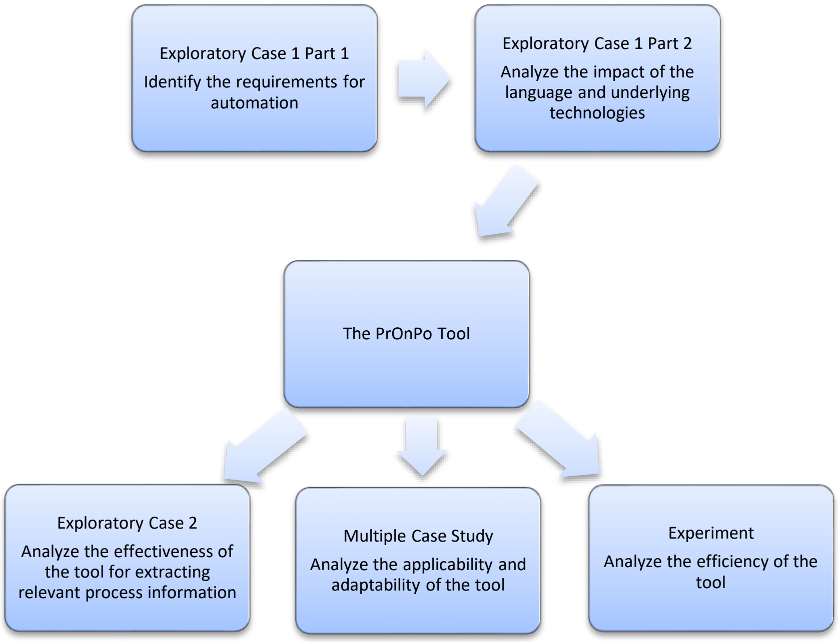


Figure 1 Research Methodology Process

1.4 Contribution and Significance of the Study

The major contribution of this study is an automated methodology and tool, called PrOnPo, which interprets guidelines written in natural language and populates process ontology. The PrOnPo methodology offers a way of developing process ontology from organizational guidelines. The PrOnPo tool consists of an algorithm that uses a natural language parser to extract process entities from sentences and transform the process entities to ontology instances. In other words, the tool automates the process knowledge acquisition and conceptualization phases of process ontology development.

The algorithm defined as part of the tool helps to resolve problems encountered when retrieving process information from natural language text using a natural language parser. Additionally, the transformation of process entities to ontology instances addresses the challenges regarding ambiguity, inconsistency, and identification of synonyms in the guidelines.

The PrOnPo tool allows analysts to develop process ontology from the given organizational guidelines in a shorter time and with less effort. Even though the analyst / user is not experienced in ontology building, the PrOnPo methodology and tool will provide guidance for process ontology development. As part of the PrOnPo tool, the Process Model Generation (ProModGen) plugin is implemented for process model creation from process ontology instances. This plugin includes an algorithm which transforms process entities extracted by the PrOnPo tool to process model elements. ProModGen generates process models using ordered and connected process activities and decreases the required time and effort for process modeling. In addition to process modeling, organizations will be able to use process ontologies to improve consistency within process information and facilitate process querying and visualization as well as data analytics.

As part of this research, the impacts of language and natural language parser on automation were also analyzed and compared. The use of keywords based on a frequency analysis for retrieving process-related sentences was investigated. The effectiveness of the PrOnPo tool was examined by implementing it in different domains using various guidelines. The efficiency of the combined use of the PrOnPo tool for process ontology development and the ProModGen plugin for process model creation was evaluated with participants.

This study is significant in terms of being the first in the literature to propose a methodology for process ontology development from organizational guidelines. Additionally, PrOnPo is the first tool to efficiently extract process information from text written in natural language in order to build process ontology. Lastly, ProModGen presents the first implementation of a plugin to transform the extracted process information to process models (in Event-Driven Process Chain notation) to formalize the specification of the processes.

1.5 Structure of the Study

The remainder of this thesis is divided into four more chapters: The second chapter presents the literature review which provides information on the state-of-art in ontology development, process discovery, and integration of ontologies with process modeling. The third chapter specifies the PrOnPo methodology and tool and the ProModGen plugin for process model generation. The fourth chapter describes the case studies and experiment which are conducted to explore, generalize and validate the proposed tool and plugin. The last chapter presents the conclusions of the study including the contributions and limitations of this thesis and recommendations for future work.

CHAPTER 2

LITERATURE REVIEW

This chapter consists of four sections. The first section describes ontology development in the literature. The second section describes process mining techniques. The third section describes integration of ontologies with business process modeling. Lastly, in the fourth section, the most related studies regarding the developed methodology and the supported tool are given.

2.1 Ontology Development

In this section ontology development is presented in three perspectives; methodologies, definition languages and lastly learning approaches for automating the development.

2.1.1 *Ontology Development Methodologies*

Some of the most well-known ontology development methodologies have been developed for building, reusing and merging domain ontologies manually (Fernández-López et al. 1997; Uschold & Gruninger 1996; Pinto et al. 2004; Tempich et al. 2005; Gangemi et al. 2012; De Nicola et al. 2005; Garcia et al. 2010; Staab et al. 2000). The common aspect of these methodologies is that they comply with the main steps for ontology development which are; identify the scope, collect data, formalize and evaluate in a waterfall like, incremental or evolutionary life cycles. Additionally, all of these studies highlight reusability when developing ontologies.

Waterfall like life cycle

METHONTOLOGY, introduced by Fernandez et al. (1997), consists of set of activities which are; specification, acquisition, conceptualization, formulation, integration, implementation, evaluation, documentation and maintenance. This set is proposed by a life cycle inspired from water fall model.

Uschold and Gruninger (1996) specify an ontology building methodology which consists of; purpose and scope identification, ontology building (capture, coding, integrating existing ontologies), evaluation, documentation and guidelines.

DILIGENT, provides a collaborative single ontology development which is composed of following five processes: build, local adaptation, analysis, revision and local update (Pinto et al.

2004; Tempich et al. 2005). However, it does not support a comprehensive approach for the development process.

NeOn Methodology, which is introduced by Gangemi et al. (2012), proposes nine scenarios for ontology development. It consists of a glossary, two life cycle models and a set of methodological guidelines. The main focus in this ontology development methodology is to reuse, reengineer and merge.

Incremental life cycle

UPON is an incremental ontology development methodology which is adapted from Unified Software Development Process (De Nicola et al. 2005). The methodology is based on cycles each of which consists of phases (inception, elaboration, construction and transition), divided into iterations, which contains workflows (requirements, analysis, design, implementation and test).

The Melting Point, proposed by Garcia et al. (2010), is composed by re-using the parts of the existence ontology development methodologies. The Melting Point is established as decentralized and incremental model which consists of the following activities; specification, conceptualization, formalization, implementation and evaluation.

Evolutionary prototyping like cycle

On-To-Knowledge, introduced by Staab et al. (2000), is defined in evolutionary prototyping which consists of the following steps; feasibility study, kick-off, refinement evaluation and maintenance.

2.1.2 *Ontology Language Definitions*

Formal languages are used to construct ontologies and are called ontology languages. Ontology languages are categorized into two; traditional and web-based.

Traditional languages are; KIF, CcyL (predicate logic); Ontolingua, F-logic and OCML (frame-based); DL, Loom (description logic based languages) (Kalibatiene 2015).

Web-based languages are; OIL, DAML+OIL, XOL, SHOE, OWL, RDF(S). These languages are based on web standards hence compatible on the internet. However, some of them can be also categorized as traditional. For instance, OWL DL is based on description logic (Kalibatiene 2015).

The most popular known languages among the ones defined above are OWL and RDF(S). RDF (Resource Description Framework) is written in XML which is interpretable by computers. It is used for adding formal semantics to the objects. It is composed of a triple; subject, predicate (property) and object. Hence, its main advantages over XML are; object-attribute structure and extensible object oriented type system (RDF Schema) (Kalibatiene 2015).

On the other hand, OWL is based on RDFS (Schema) which has additional properties such as conjunction, disjunction, existentially and variables which supports logical reasoning. Due to its complexity, sublanguages such as OWL Lite, OWL DL and OWL Full are designed. However, OWL DL loses compatibility with RDF and OWL Full may not be effective for reasoning (Kalibatiene 2015).

2.1.3 *Ontology Learning*

Ontology development is documented as a tedious process since it requires much time and effort (De Cesare et al. 2014; Jiang & Tan 2010). It is believed that the automation of ontology development will reduce costs, improve the reliability and scope of ontologies, and widen applications (Hazman et al. 2011). Ontology learning uses methods from machine learning, knowledge acquisition, natural-language processing, information retrieval, and artificial intelligence (Wong et al. 2012; Hazman et al. 2011).

Two well-known tools for domain ontology learning are; TERMINAE and CRCTOL.

TERMINAE: Biebow and Szulman (1999) introduce a method and a tool named TERMINAE in order to build domain ontology. The TERMINAE has been built to meet the requirements which are; being a linguistic-based methods, providing a typology of concepts to highlight the modeling choices, formality to avoid incoherence and inconsistencies as much as possible traceability, maintainability and back linking to texts.

CRCTOL: Jiang and Tan (2010) indicate that traditional systems for ontology learning employ shallow natural language processing techniques and focuses on concept and taxonomic relation extraction. Therefore they present a Concept-Relation-Concept Tuple-based Ontology learning system (CRCTOL) in order to enable full text parsing with statistical and lexicon-syntactic methods for mining ontologies from domain-specific text documents. CRCTOL's system architecture includes data importer, natural language processing (NLP), algorithm library, domain lexicon, user interface and data exporter (Jiang & Tan 2010).

However, the surveys which summarize methods and tools supporting the ontology learning methods claim some drawbacks for ontology learning.

- Gomez-Perez and Manzano-Macho (2003), for example, highlight the lack of a detailed methodology for ontology learning from texts; a need for user involvement since these tools cannot be fully automated; and a need for a general approach for evaluation.
- Shamsfard and Barforoush's (2003) survey findings state that most of the works concentrate on discovering taxonomic relations but there is less work on non-taxonomic relations and learning on axioms. Additionally, they claim that the current systems are evaluated according to their domain and the tools are developed as semi-automatic.
- Ding and Foo's (2002) survey underlines that the most methods are for structured data and the task of identifying relations is hard.
- Zhou's (2007) survey, lastly, remarks the importance of representing the development of ontologies and user involvement in ontology learning tools.

To sum up all these findings, Wong et al. (2012) come to a conclusion that fully automating ontology learning might not be possible, there is a lack of common evaluation platforms and the discovery of relations between concepts still needs to be studied.

Because the methods defined in the literature is for building domain ontologies they will not work well when specializing the scope of the ontology for building process ontology since a new training data is required to collect if not available.

2.1 Process Mining & Discovery

Process mining is composed of the three tasks of discovery, conformance and enhancement (van der Aalst & Weijters. 2005). Process discovery is the task in which the event log is used for constructing a process model whereas in the conformance and enhancement tasks, there is an existing process model, which is analyzed and improved according to the event logs. However, there are also studies which discovers and constructs process models from text. The two approaches using event logs and text will be presented respectively.

2.1.4 *Process Discovery from Event Logs*

Studies which concentrate on process discovery techniques to construct process model uses the event logs of the information system and creates Petri nets (Agarwal & Singh 2014; de Medeiros et al. 2008; van der Aalst & Weijters. 2005; De Leoni et al. 2016; Goedertier et al. 2011).

Agarwal and Singh (2014) compare process mining tools (ProM, Fluxicon, Emit, ARIS PPM) which use event logs. ProM and Emit creates Petri-Nets based on event logs whereas Fluxicon and ARIS PPM are process performance manager to assess the existing process models.

ProM[1] is one of the most well-known process mining tool which is an extensible framework consisting of plugin for process mining techniques (van Dongen et al. 2005). The most of the studies in the literature implements a plugin on ProM with a new technique for process discovery. Some of these studies are presented below.

- Medeiros et al. (2008) focuses on process discovery to build Petri-Nets. They claim that mining techniques cannot show the precise flow of the traces in the event logs, and present a clustering approach on ProM.

- De Leoni et al. (2016) claim that due to the heterogeneous data in the logs, existing approaches do not reflect perceptive models. The present a new approach on ProM for correlate, predict and cluster event logs for process discovery.

- AGNEs is based on first-order classification learning. Initially it learns the relations between activities and artificially generates negative events to detect transitions (Goedertier et al. 2009).

- Heuristics Miner extends alpha algorithm, starts with the construction of dependency graphs to build relations and creates matrix of preconditions from the dependency graph (Weijters et al. 2006).

The common challenge arises with process mining in event logs is eliminating the irrelevant information which is considered as "noise". Goedertier (2011) overviews the state of the art of process mining techniques in in terms of their effectiveness and sensitivity to noise. These techniques are; Alpha algorithms, Genetic Miner, AGNE and Heuristics Miner. Alpha Algorithms (Van Der Aalst et al. 2004) learn structured workflow nets from complete event logs but are

[1] www.promtools.org

sensitive to noises. Heuristics Miner, on the other hand, uses frequency information for threshold setting and hence is not weak against noise. Genetic Miner employs genetic algorithm and is good at detection of non-local patterns

Information systems create event logs in a format which are considered as structured or semi-structured. Hence these techniques do not provide a solution for one of the challenge (13) proposed by Mendling et al. (2014) which suggests the potential efficiency gains are associated with transforming text to process model.

2.1.5 *Process Discovery from Text*

Ghose et al. (2013) present two approaches of text-to-model and model-to-model for the process discovery of model and text artefacts. They present a method using natural language parsers which extract the information from the text and interpret it. Their second approach of model-to-model extraction uses sequence, case and state diagrams to extract the process model elements.

Gonçalves, et al. (2009), on the other hand, present a work which used group stories for business process mining which involves people in the organization collecting their daily activities and defining their work by writing group stories. Then, these stories are used for the automated generation of process models using text mining and natural language parser techniques.

Sinha and Paradkar’s (2010) research is based on textual business use cases to create business process model notation representations. First, they parse the natural language text and create in-memory model. Secondly, the in-memory model is transformed to the process model. Lastly, they combine all the use cases of the process models to finalize the business process model and produce the final version.

Furthermore, Friedrich et al. (2011) propose an approach for process model generation from natural language text. They show that the required information for process modeling exists in policies, reports, forms, manuals, and email messages. Their approach consists of three phases: Firstly, text is parsed using a Stanford Parser and converted into a tree structure, secondly, these structures are semantically analyzed and the meaning of words and phrases are gathered, and lastly the pronouns and articles which refer to concepts are identified with anaphora resolution. Their approach is reported to be 77% successful. The details of the transformation afterwards can be found in (Friedrich et al. 2011).

The common aspects of process discovery techniques is that they use a natural language parser for mining text but they differ in using various text sources such as group stories, emails, reports or use cases. Overall, these techniques generate process models but they exclude ontologies during this process.

2.2 **Ontologies and Business Process Modeling**

Due to the similarity of creating knowledge of domain, studies have concentrated on the significance of relating the ontologies to process models (Höfferer 2007; Haller et al. 2008; Cherfi et al. 2012; Francescomarino et al. 2014; Fan et al. 2016) and creating ontologies from process models became a popular topic. These studies are categorized as those which integrate existing domain ontologies with process models and those which create process ontologies from process models.

2.1.6 *Integrating Domain Ontologies with Process Models*

Thomson and Fellmann (2007) suggest mapping of process models to ontology in order to enable linking between formal and semi-formal by matching model elements to concepts. First, they engaged in the mapping for EPC models and then for BPMN (Thomas & Fellmann 2009).

Dimitrov et al. (2007) claim that companies are struggling due to shortcomings of current process management technologies and suggest enhancing the existing process models with semantic annotations. The authors define a SUPER ontology which enhances existing BPMN and EPC models with ontologies.

Cherfi et al. (2012) propose a method to improve the semantic quality of process models by relating business process models with domain ontologies. They propose type-based and semantic-based mapping strategies between process models and ontologies.

Francescomarino et al. (2009) provide tool support for automatically populating ontology from process models. In their later studies, they suggest a collaborative specification of semantically annotated processes by constructing a framework (Chiara Di Francescomarino, Chiara Ghidini, Marco Rospocher & FBK-irst 2011). In one of their studies, more relevant to our study, they conduct an experimental study in order to investigate the semantic annotation usage in process modeling (Francescomarino et al. 2014). According to their findings, they highlight that the process models modeled using the semantic annotations are of higher quality than the one which does not use semantic annotation. However, they claim that the time spent on modeling is not affected by this treatment.

Alkhaldi, Casteleyn and Gailly (2015) use enterprise ontology for creating process models in BPMN notation and claim that there are inconsistencies in the process models when they are created by different modelers. Their aim is to improve the semantic consistency of the process models by suggesting concepts to the modeler from the enterprise ontology. However, they do not establish any transformation from the ontology to process model.

A study similar to our study is conducted by Fan et al. (2016) who use existing domain ontology for business process modeling in BPMN. They convert the domain ontology using an algorithm to process ontology. Afterwards, these ontologies assist the BPMN model in generation and validation. Their approach is not fully automated and requires user involvement for identifying the roles in the beginning and refining the logical connections at the end for the model generation. However, they claim that business process models created by using domain process ontologies have better quality validating their claim by conducting an experiment, in which one group uses process ontology for modeling and the other group does not. The result is significant showing that the group, which uses the process ontology in process modeling, produces more qualified models.

The scope of the process ontologies created in these studies are limited with the resources that exist in the domain ontologies. Hence, process information in process models may not be fully presented in the formal specifications.

2.1.7 *Transforming Process Models and Descriptions to Process Ontologies*

Koschmider and Oberweis (2005) state that using semantic meta-data for process modeling facilitates communication. Their approach is to store the process elements directly in the ontology with relations such as has-node and has-arc.

Höfferer (2007) supports using ontology with process models by claiming that it supports interoperability in process models. He suggests creation of ontologies from process models.

Haller et al. (2008) propose a method for transforming XML Process Definition Language (XPDL) models to ontologies by introducing oXPDL, process interchange ontology.

Hepp and Roman (2007) use semantics in business process modeling to support process modeling with ontology. They aim to represent the process model ontologically.

Belhajjame and Brambilla (2009) present their work which uses ontology for defining business process modeling. They describe business process ontology concepts using semantic annotations of business process sets. Then, they use the ontology for business process discovery.

Eisenbarth (2013) automatically transforms process models (EPC, BPMN or Eclipse Java Workflow Tooling) to process ontology. This process ontology is then integrated with resource ontology in which automated model adaptation, resource requirement matching, and resource classification are possible.

The De Cesare et al.'s (2014) relates process ontologies with business process modeling. In their study, they populate process ontologies from the textual descriptions of the business process models.

Leopold et al. (2015) automatically generate an annotation taxonomy for process models. Their intent is to match the similarity between process activity with a concept and process model with a taxonomy category. They use Markov Logic formalization in their automation phase to match the annotation to the process models.

The scope of the process ontologies developed from these studies is limited to the information presented on the process models or descriptions. Hence, the process querying from these ontologies might offer high-level descriptive information in the process models and will exclude the detailed process information in other sources.

2.3 **A Brief Discussion of the State of the Art**

The studies given in the first section concentrate on ontology development methodologies. The proposed methodology follows a step by step (like a waterfall) approach and follows similar steps as defined methodologies, such as; specification, acquisition, conceptualization, evaluation and documentation. However, the proposed methodology focuses on process ontology development whereas the existing methodologies are for domain ontology development.

The second section provides studies which discover processes from event logs (semi-structured/structured) and text (unstructured). Since the event logs are kept in a format by the information systems, they are considered as structured or semi-structured. The proposed study focuses on process discovery from text and hence natural language parsers are used likewise process discovery techniques from text. In contrast to the studies which discover process from text which excludes ontologies, this study conceptualizes the processes as ontology individuals/instances and then use them to create process models.

The most related studies regarding to the proposed study are given in the third section. These studies inspire us to enhance business process models with ontologies. However, some of them use the existing ontologies (Fan et al. 2016; Francescomarino et al. 2014; Cherfi et al. 2012; Thomas & Fellmann 2009; Dimitrov et al. 2007) whereas some of them transform process models to ontologies (Leopold et al. 2015; De Cesare et al. 2014; Eisenbarth 2013; Haller et al. 2008; Höfferer 2007). These studies limit the scope of the process ontologies either to the scope of the domain ontology or scope of the process models. Hence this arises the incompleteness of the process ontology and formal specification of the process information.

However, in the manner of using ontologies for process modeling, Alkhaldi et al. (2015), Fan et al. (2016) and Francescomarino et al.'s (2014) works are the most related studies to the proposed approach. They have validated the improvement in; the quality of process models using domain ontology (Fan et al. 2016), semantic consistency using enterprise ontology (Alkhaldi et al. 2015) and semantic annotation (Francescomarino et al. 2014).

Francescomarino et al. (2014) uses semantic annotations for collaborative process modeling in BPMN. Their experiment concentrates on the effect of using semantic annotations on the variables defined for collaborative process modeling and as well as the quality of process models. There is an agreement in the proposed and existing research, on preventing conflicts by using semantic annotations in process modeling. However, the proposed research differs from their study, where process ontology is used to generate process models in EPC notation. Similar to Francescomarino's (2014) study, Alkhaldi et al. (2015) use the enterprise ontology for creating consistent process models in BPMN notation. However, they do not propose a transformation approach. In this manner, the work of Fan et al. (2016) is more similar to ours in which they convert domain ontologies to process ontologies and semi-automatically generate BPMN using the process ontologies. We agree on enhancing the quality of the process models by aligning them with ontologies. However, we do not use existing domain ontologies, but instead develop process ontology from guidelines and semi-automatically generate process models.

In brief, the literature reveals that process ontology discovery from written text has not been investigated in detail. Our contribution in this study is to provide automated process ontology discovery and population from organizational guidelines, which are more comprehensive than domain ontologies and process ontologies generated from process models in terms of process information.

CHAPTER 3

A PROCESS ONTOLOGY POPULATION METHODOLOGY AND TOOL SUPPORT

This chapter presents the PrOnPo methodology and tool for process ontology development from organizational guidelines and ProModGen plugin for transforming the process ontology instances to business process models. The first section describes the PrOnPo methodology. The second section describes the PrOnPo tool and the third section presents ProModGen plugin.

3.1 The PrOnPo Methodology

Ontology development refers to creating ontology classes, individuals and properties from scratch whereas ontology population refers to creating individuals for the existing ontology classes. Throughout this study, the term ontology development is used because the ontology classes from scratch are created and the term ontology population is used because individuals for the ontology classes are created.

The PrOnPo methodology consists of the phases of specification, acquisition, evaluation and conceptualization. Specification is based on defining the scope of process ontology using competency questions and acquisition is based on uncovering the process information from organizational guidelines and creating ontology individuals. Evaluation involves evaluating the extracted process information and conceptualization is undertaken by creating ontology classes and properties and finally populating ontology classes with individuals. The following subsections will present the phases in detail.

3.1.1 *Specification*

Process ontology development starts by defining the scope of process ontology to be developed. Our approach is to construct process ontology from the organization's guidelines. First, the competency questions are established regarding the selected guidelines that the developed process ontology is expected to answer. Business processes are extracted by conducting interviews, analyzing information systems event logs, and interpreting the organization's guidelines. The organizational guidelines, which can be in a form of regulations, directives or procedures, define the set of actions and processes with business rules that should be followed within or by entities interacting with the organization. These documents are known to be a primary unstructured source (written in natural language) for extracting process information. Our scope for extracting processes, is using the organization's guidelines for constructing a process ontology. Hence in the specification phase, competency questions regarding the selected guidelines is established, which the developed process ontology is expected to answer.

3.1.2 *Acquisition*

Organizational guidelines consist of sentences containing activities and business rules, examples of which are given below;

- ✓ the supporter of the activity; i.e., the role,
- ✓ the information carrier used to perform the activity,
- ✓ the time when the activity is performed or triggered; i.e., the event,
- ✓ the business rule which should be considered when performing the activity.

In this context, these partitions of the sentences can be extracted using what (what is used), who (who supports activity), where (where the activity is performed), when (when the activity is performed, why and how (why and how the activity is performed) questions. Acquisition phase includes extracting sentences and partitions of the sentences based on these questions.

3.1.3 *Evaluation*

Evaluation is required to check if the extracted process information can answer the competency questions. In this phase, missing process information is included, misplaced individuals are replaced or irrelevant information is eliminated by user involvement. After the evaluation, the process information is conceptualized.

3.1.4 *Conceptualization*

The PrOnPo methodology uses EPC notation elements for developing the process ontology. EPC is a well-known imperative modeling notation used for business process modeling (Reijers et al. 2013). There have been studies which enhance business process modeling semantically by creating BPMN ontology (Ghidini et al. 2008; Abramowicz et al. 2007). In order to extend a previous study (Coskuncay et al. 2016) and enhance semantic quality of EPC models, the PrOnPO tool was implemented according to the EPC meta-model.

In the conceptualization phase, the process elements in Figure 3 are defined as ontology classes. The relations between the process elements in Figure 3 are defined as ontology properties. Partitions extracted during Acquisition phase are transformed to ontology individuals/instances.

The following section will describe how this methodology is supported by a tool with motivating examples which will illustrate the PrOnPo methodology in detail.

3.2 **The PrOnPo Tool**

The PrOnPo tool is implemented in order to automate the Acquisition and Conceptualization phases of the methodology. The requirements for automation of these phases are identified by conducting an exploratory study which is given in section 4.1.4. Initially, a prototype is implemented given in section 4.1.4.2.3. According to the application of the prototype, improvement potentials are identified and PrOnPo tool is implemented.

This tool is implemented in Java 8 in Eclipse programming environment. The class and sequence diagrams of the implementation are given in Appendix A. It gets the guideline in pdf/doc format as input and creates the output in RDF/OWL and CSV format. This tool can be used as semi-automated if user involves in Evaluation phase and full-automated if the Evaluation phase is skipped. During Evaluation phase, the user can assist to the CSV file, correct misplaced partitions, if necessary. The PrOnPo tool uses the Stanford Parser, which is known to be the most common parser (Manning et al. 2014) for natural language processing and Apache Jena libraries for creating RDF/OWL.

The high level process diagram of PrOnPo tool is given in Figure 2. The map between the methodology phases and tool's components is given in Table 1. The following subsections will describe each process presented in the diagram. Additionally, the user guide for the tool is provided in Appendix B.

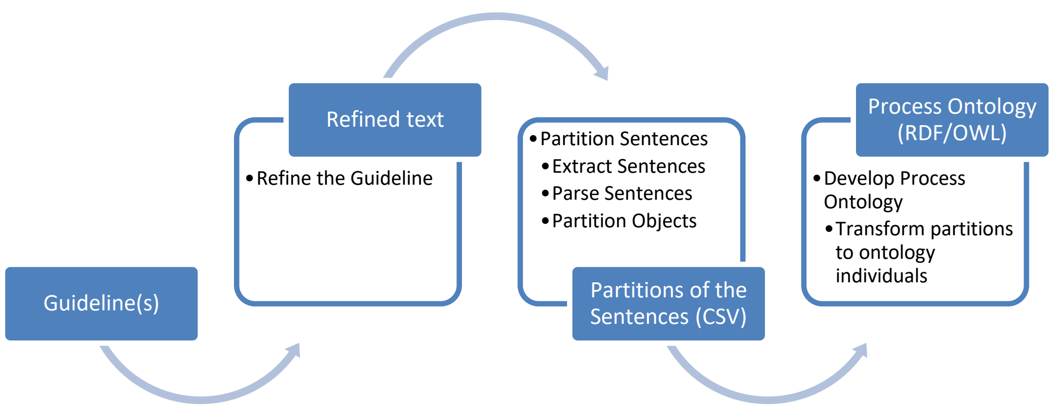


Figure 2 High Level Process Diagram of the PrOnPo Methodology

Table 1 Mapping between methodology's phases and tool's components

Methodology	**Semi-Automated**	**Full-Automated**
Specification	Refine Guideline	Refine Guideline
Acquisition	Extract Sentences Parse Sentences Partition Objects	Extract Sentences Parse Sentences Partition Objects
Evaluation	Evaluation	------------------
Conceptualization	Transform partitions to ontology individuals	Transform partitions to ontology individuals

3.2.1 *Refine the Guideline*

The guideline needs to be refined for the parser. This refinement includes; quotation marks, bullets and constructing a text file with one sentences in each line. This section will first give brief introduction about the organizational guidelines. Afterwards it will describe how the guideline is refined.

3.2.1.1 *Organizational Guidelines (Input)*

Business processes are extracted by conducting interviews, analyzing information systems event logs and interpreting the organization's guidelines. Organizational guidelines, which can be in a form of regulation, directive and procedure, define the set of actions and processes with business rules that should be followed within or by the entities interacting the organization. These documents are known to be primary unstructured source for extracting process information. We categorized the contents of these documents as;

- ✓ Terms
- ✓ Processes; Roles, Information Carriers, Functions, Applications, Technical Terms and Location
- ✓ Business rules
- ✓ Irrelevant sentences

3.2.1.2 *Quotation Marks*

Quotation marks such as {",), (, ;, /} cannot be parsed by the parser and pose corrupted sentences, in which the triples cannot be formed and result to missing information. Therefore the document is searched and these quotation marks are replaced by "" empty string.

3.2.1.3 *Bulleted Lists*

Parser cannot interpret the bulleted lists. Therefore, if applicable, the following two rules are performed;

- ✓ If the bullets are followed by the text *"following:"* or *"to:"* the beginning sentence is copied and pasted to the beginning of the bulleted text.
- ✓ If not, the bullets are replaced by empty string "" and bulleted texts are considered as normal sentences.

Motivating example is given below;

"If the customer is not located in Australia, perform the following:
- *Broker determines the customer's reason for wanting the product*
- *Broker records response on application (or attach separate paperwork)*

- *Broker collects the country of residence"* [2]

In this example sentences are formed as;

"If the customer is not located in Australia, broker determines the customer's reason for wanting the product.
If the customer is not located in Australia, broker records response on application or attach separate paperwork.
If the customer is not located in Australia, broker collects the country of residence."

3.2.1.4 *Text File*

Afterwards each sentences (begin and end if next character is ".") is copied to a text file. Note that each line contains only one sentence.

3.2.2 *Partition Sentences*

This section describes the second process of the PrOnPo tool. Partitioning sentences consists of; extracting sentences, parsing the sentences into subject-verb-object form and partitioning the object. The output of this process is partitioned sentences. At the end of this process, user can decide to use the PrOnPo tool semi-automated or full automated. In the semi-automated approach user can assist the partitions and replace the wrong ones, if necessary. Else, these partitions are directly given as input to the transformation process.

3.2.2.1 *Extracting Sentences*

In order to extract the sentences which are containing the process activity information, terms and business rules related with the processes and as well as to prevent the redundancy, keywords may be used. These keywords are chosen using frequency analysis. For instance, in the motivating example 383 nouns are extracted and their frequencies are calculated for the Regulation of Graduate Studies (studied in the Exploratory Study 4.1.5). When all of these 383 nouns are used as tags, 391 sentences are extracted. Each time the most infrequent nouns are eliminated, the number of extracted output is recorded as stable with minor decreases. At one point the variance in the number of the extracted sentences started to increase. For the case studied in the Exploratory Study 4.1.5, that point was the fifth most frequent noun. Therefore, five most frequent nouns have been chosen as keywords.

This part is optional and if the keywords words are not used, all of the sentences are given as input to the parser.

3.2.2.2 *Parsing the Sentences*

The sentences are parsed into subject-verb-object form using the Stanford Parser (Manning et al. 2014). The Stanford Parser constructs the parsing output in the Tree data structure. One motivating example can be seen below, from Multiple Case in section 4.2.

---

[2] Excerpt from INGDirect procedure used in the Experiment

Sentence: “*For work that lasts one week or more, the Contractor must submit a report of onsite activities to the Schneider Electric representative using the Contractor Safety Report form Attachment H.*”[3]

Parsed Tree:

```
(ROOT
  (S
    (PP (IN For)
      (NP
        (NP (NN work))
        (SBAR
          (WHNP (WDT that))
          (S
            (VP (VBZ lasts)
              (NP
                (NP (CD one) (NN week))
                (CC or)
                (NP (JJR more))))))))
    (, ,)
    (NP (DT the) (NNP Contractor))
    (VP (MD must)
      (VP (VB submit)
        (NP
          (NP (DT a) (NN report))
          (PP (IN of)
            (NP (JJ onsite) (NNS activities))))
        (PP (TO to)
          (NP
            (NP (DT the) (NNP Schneider) (NNP Electric) (NN representative))
            (VP (VBG using)
              (NP (DT the) (NNP Contractor) (NNP Safety) (NNP Report) (NN form) (NNP Attachment) (NNP H.)))))))))
```

Rules:
- If the NP tag is the child of the Root, then it is the subject
- Else if the S tag is the child of the Root, then it is the subject
- Extract the VP tag under the Root.
  - Extract the verb tags under VP, if available, place them in verb
- The rest is the object

Algorithm:

```
for( each sentence )
     root = parse (sentence)
     child= root.getChild()
     if( child.tag == NP)
          subject = child
     else if( child.tag == S)
          subject= child
     else if(child.tag == VP)
          child= child.getChild()
```

[3] Excerpt from Contractor Safety Procedure used in Multiple Case Study

```
if(child.tag == VP)
      verb = child
if(childs.tag == NP)
      object = child
```

Scalability:
This algorithm is a linear algorithm with complexity of O (n). Therefore the time complexity of the algorithm will increase linearly as the number of sentences in the guideline increases.

Parsed Output:
Subject: *the Contractor;*
Verb: *must submit;*
Object: *For work that lasts one week or more, a report of onsite activities to the Schneider Electric representative using the Contractor Safety Report form Attachment H.*

3.2.2.3 *Partitioning the Object*

The aim is to extract the process elements from the sentences. The verb of the sentence is corresponding to the Function element. The Role element might be in the subject if the verb is active. Else if the verb is passive then it might be in the object. The Information Carrier element might be in the subject if the verb is passive. Using Stanford CoreNLP, verb's type (nsubj if active and nsubjpass if passive) can be identified. This can help to identify if the subject is corresponding to Role or Information Carrier elements.

For the other elements, object should be partitioned. This partition can be done by asking who (to who, by who), what, where, when, why and how questions to the verb. The Who answer will give the Role element which is supporting/carrying out the function. The What answer will give the Information Carrier (default) and Technical Term elements. The Where answer will give the Application, Location or Information Carrier (default) Elements. The When answer will give the Event element. The Why and How answers will give the related Business Rule element.

We have created rules in order to partition the object and identify the process elements. The following rules are based on the tags of the words which are created by the parser (using Penn TreeBank[4]). However, we tried to avoid any rules which will work for one case but will destroy the other cases. As we run all the guidelines using this algorithm, we identified that it works best for fourth level of the parsed tree.

Rules:

- If the SBAR tag is the child of the NP tag or the ROOT then extract the SBAR tag put it under to Where
- If the PP tag is the child of the Root then put it under When
- If the ADVP tag is the child of the Root then put it under How
- If the VP tag is the child of the Root then go deeper

[4] http://www.surdeanu.info/mihai/teaching/ista555-fall13/readings/PennTreebankConstituents.html

- (Level 1) RB, TO, RP tags are added to the verb
- (Level 1) If the PP tag is the child of VP tag, then extract the PP under who
    - (Level 2) If VBG is the child of PP then extract it under How
    - (Level 2)Else if IN is the child of PP then extract it under Why
- (Level 1) If the NP tag is the child of VP tag, then extract it under What (object)
    - (Level 2)If VP tag is child of the NP tag then extract the VP tag
        - (Level 3) If the VBG tag is child of the VP tag, then extract it under How
        - (Level 3) If the TO tag is child of the VP tag, then extract it under Why
        - (Level 3) If the PP tag is child of the NP tag, extract the PP tag
            - (Level 4) If the IN tag is child of the PP tag, and it contains
                - "for", "from" or "by" then extract it under Who
            - (Level 4) If the TO tag is child of the PP tag, then extract it under Who
            - (Level 4) If the VBG tag is child of the PP tag, then extract it under How
- (Level 1) If the SBAR tag is child of VP tag, then extract it under Where
- (Level 1) If the ADJP tag is child of VP tag, then add it under What (object)
- (Level 1) If the S tag is child of VP tag, then extract it under Why
    - (Level 2) If the NP tag is child of S tag, then add it under What (object)
    - (Level 2)If the VP tag is child of S and
        - (Level 3) VBG tag is child of VP tag, then extract it under How

Algorithm:

```
for( each sentence )
        root = parse (sentence)
        children= root.getChildren()
                for(each child in children)
                        if(tag == SBAR)
                                where =child
                        if(tag == PP)
                                when = child
                        if(tag == ADVP)
                                how == child
                                if(tag == VP)
                                        children= child.getChildren()
                                        if(child in children == RB or TO or RP)
                                                verb += child
                                        if(child in children == PP)
                                                who == PP
                                        children= child.getChildren()
                                                if(child in children == VBG)
                                                        how = child
                                                else if(child in children == IN)
                                                        why = child
                                        if( child in children == NP)
                                                what = child
                                                children = child.getChildren()
                                                if(child in children == VP)
                                                        children= child.getChildren()
                                                        if(child in children == VBG)
                                                                how = child
                                                        if(child in children == VBG)
```

```
                    why = child
            if(child in children == PP)
                    children= child.getChildren()
                    if( child in children == IN)
                    who = child
                    if( child in children == TO)
                    who = child
                    if( child in children == VBG)
                    how = child
    if(child in children == SBAR)
            where = child
    if(child in children == ADJP)
            what = child
    if(child in children == S)
            why = child
            children = child.getChildren()
            if( child in children == NP
                    what = child
            if( child in children == VP
                    how = child
```

Scalability:
This algorithm is a linear algorithm with complexity of O (n). Therefore the time complexity of the algorithm will increase linearly as the number of sentences in the guideline increases.

3.2.2.4 *Partitions of the Sentences (Output 1)*

These partitions of the sentences are recorded into CSV file. At this point the user can assist the file and correct the misplaced partitions. The motivating example given in section 3.2.2.2 is partitioned as given in Table 2. As it can be seen, the phrase "using the Contractor Safety Report from Attachment F" should be replaced from Who to under How. This phrase is on the 6th level of the parsed tree and therefore it is out of the scope of the defined rules. If the user do not assist in this part, the individuals will be created as they are placed. Additionally, this CSV file is created with three more columns which are; What-Technical term, Where-Location and Where-Application. The answers for these questions are created default under Information Carrier (What and Where-document), if user does not replace them under technical term, location or application.

Table 2 Except from Contractor Safety Procedure

Subject	Verb	What	Who	Where	When	Why	How
the Contractor	must submit	a report of onsite activities	to the SE representative using the Contractor Safety Report form Attachment H.		For work that lasts one week or more		

3.2.3 *Develop Process Ontology*

This section is composed of; ontology class creation and ontology individual creation. The input to this process is the CSV file created in the previous process. This CSV file can be either edited by the user or it can directly be transformed to process ontology. The output of this process is the process ontology in RDF/OWL format. One sample Ontology Graph for Manufacturing Process is given in Appendix C.

3.2.3.1 *Ontology Class and Relation Creation*

The process elements given in Figure 3 are most common used model elements in EPC notation. Each process element belongs to a process which can be in type of application, information carries, technical term, event, location, role, business rule and function. In the process model diagram, each of the connection between process elements represents a relationship. Event connected to a function indicates that event triggers the function, whereas function connected to an event indicates function creates event. Information carrier can be input to a function or function can create an information carrier as an output. Each of these elements are defined as ontology classes. And each of these relations between the process elements are defined as object properties between ontology classes.

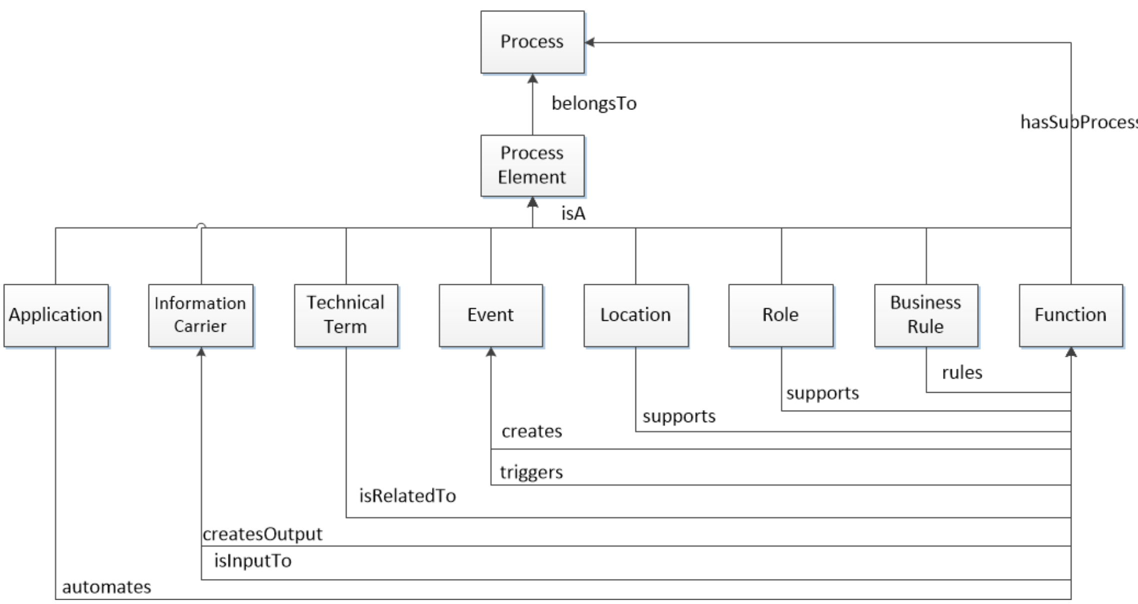


Figure 3 Meta-model for EPC elements and relations

3.2.3.2 *Ontology Class Individual Creation*

Using Apache Jena libraries, each of the partitions of the sentences are defined as ontology individuals. These individuals are then matched to the corresponding ontology classes according to Figure 4. The formal specification of two sample process sentences from Manufacturing Process studied in Multiple Case Study in RDF format is given in Table 3. The first sample is the formal specification of the sentence "Each contractor - must renew - approval - at the beginning of each calendar year". The second sample is the duties of Location Plant Engineer. The second columns corresponding to the rows give the meaning of the ontology language. The onto-graph excerpt of the Manufacturing Process for Location Plant Engineer's RDF output (second sample) is given in Figure 5.

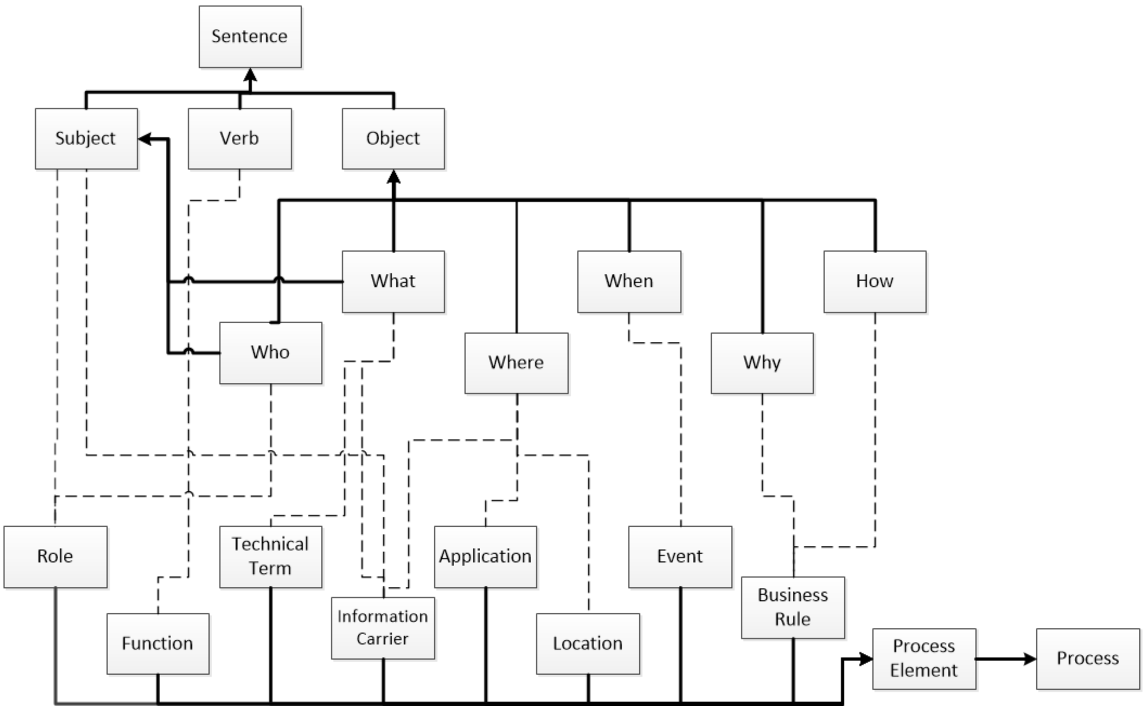


Figure 4 Meta-model for matching the sentence's entities with process entities

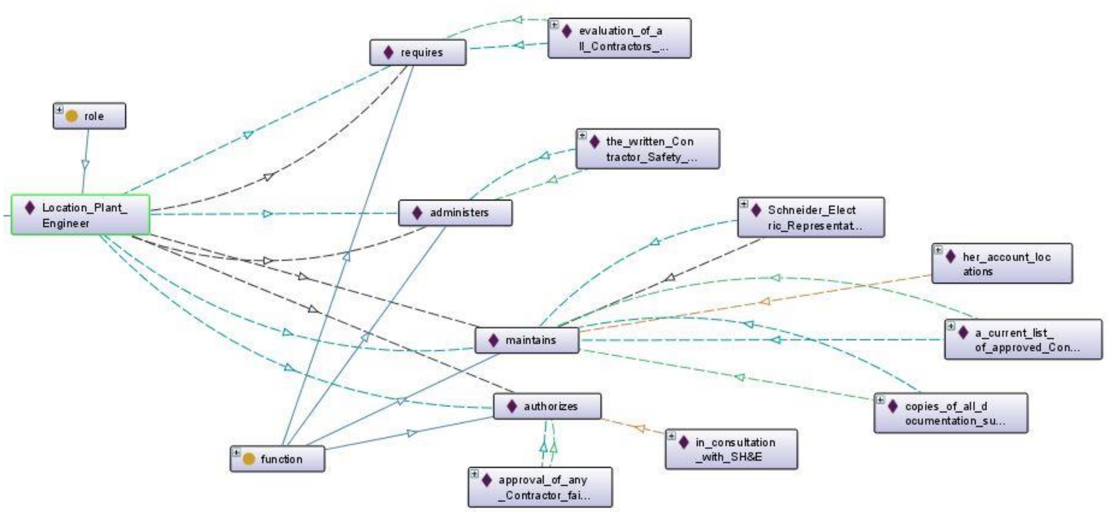


Figure 5 Populating the upper level ontology with individuals (yellow filled circles indicate classes, purple filled diamonds indicate individuals)

Table 3 RDF Excerpt from Manufacturing Process

-------------------Sample Sentence-------------------- <Each_contractor> a <processontology.com/role> ; <processontology.com/supports> <must_renew> . <must_renew> a <processontology.com/function> . <at_the_beginning_of_each_calendar_year> a <processontology.com/event> ; <processontology.com/triggers> <must_renew> . <approval> a <processontology.com/informationCarrier> ; <processontology.com/createsOutput> <must_renew> ; <processontology.com/isInputTo> <must_renew> .	(‘Each contractor’ is a role and supports ‘must renew’ where ‘must renew’ is a function. 'at the beginning of each calendar year’ is a event and triggers ‘must renew’ ‘approval’ is a information carrier and createsOutput and isInputTo ‘must renew’)
------------------Location Plant Engineer ------------ <Location_Plant_Engineer> a <processontology.com/role> , <processontology.com/supports> <administers> , <authorizes> , <maintains> , <requires> . <administers> a <processontology.com/function> . <the_written_Contractor_Safety_Program> a <processontology.com/informationCarrier> ; <processontology.com/isInputTo> <administers> . <authorizes> a <processontology.com/function> . <in_consultation_with_SH&E> a <processontology.com/businessRule> ; <processontology.com/rules> <authorizes> . <approval_of_any_Contractor_failing_to_meet_the_minimu m_valuation_criteria_as_outlined_in_this_Program> a <processontology.com/informationCarrier> ; <processontology.com/isInputTo> <authorizes> . <maintains> a <processontology.com/function> . <for_use_at_his/her_account_locations> a <processontology.com/businessRule> ; <processontology.com/rules> <maintains> . <a_current_list_of_approved_Contractors_> a <processontology.com/informationCarrier> ; <processontology.com/isInputTo> <maintains> . <requires> a <processontology.com/function> . <evaluation_of_all_Contractors_submitting_bids_for_wor k> a <processontology.com/informationCarrier> ; <processontology.com/createsOutput> <requires> .	(Location Plan Engineer is a role and supports ‘administers’ , ‘authorizes’, ‘maintains’ and ‘requires’ where ‘administers’ is a function , and ‘written Contractor Safety program’ is a informationCarrier and isInputTo ‘administers’ , where ‘authorizes’ is a function and ‘in consultation with SH&E’ is a businessRule and rules ‘authorizes’, where ‘maintains’ is a function and ‘a current list of approved contractors is a informationCarrier and isInputTo ‘maintains’, where ‘requires’ is a function and ‘evaluation of all Contractors submittin bids for work’ is a informationCarrier which ‘requires’ createsOutput.)

3.3 The ProModGen Plugin

This section describes the business process model generation from process ontology as part of process ontology development. The class and sequence diagrams are given in Appendix A. The system interface and components of the ProModGen are given in Figure 6. The ProModGen view-plugin is implemented on UPROM in order to transform the process ontology instances to process model elements. In the given diagram, ProModGen plugin is added on the original diagram (Coskuncay et al. 2016) and the assistance of the analyst is only shown for the process model generation plugin (ProModGen). The class diagram for the implementation is given in Appendix A.

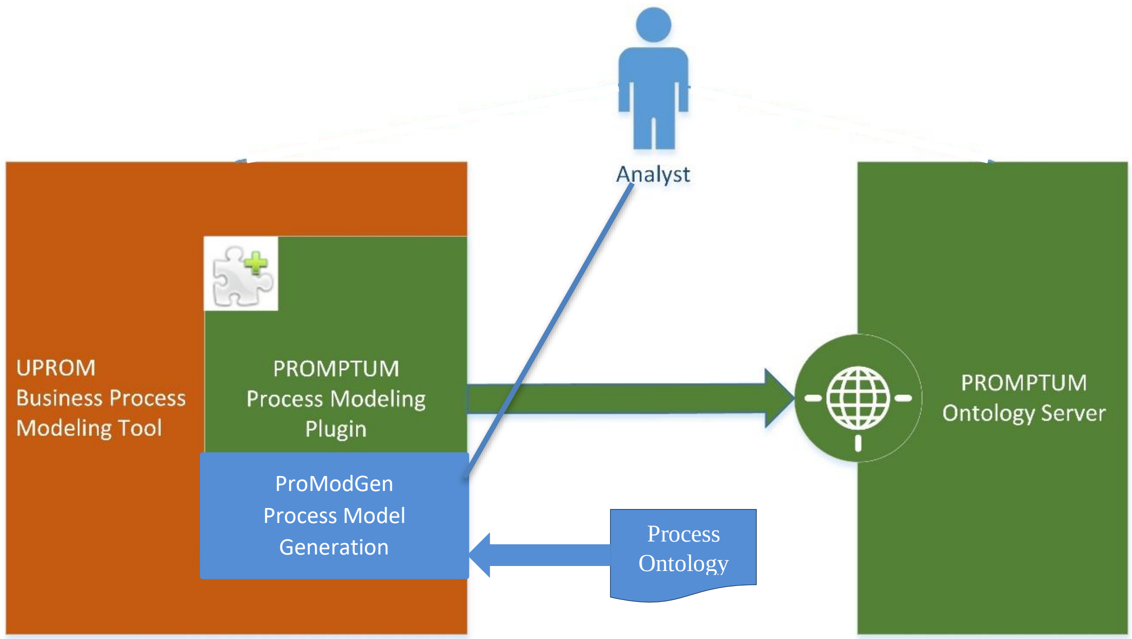


Figure 6 Components and system interfaces of PROMPTUM toolset with ProModGen plugin added

PROMPTUM; PROMPTUM (showed in green) was implemented, to enable domain ontologies to be developed by using a set of business process models and existing domain ontologies be used in business process modeling (Coskuncay et al. 2016). In other words, the relevant concepts, labels etc. in business process models are stored as domain ontology and updates in the stored domain ontology is reflected to the process models.

UPROM; PROMPTUM was implemented on UPROM tool (showed in orange) (Aysolmaz & Demirörs 2015), which is the medium where process modeling is performed and business process models are stored. UPROM tool uses bflow* Toolbox (Laue et al. 2015) as core by following Eclipse Modeling Framework (EMF) and Eclipse Graphical Modeling Framework (GMF). The user interface of the UPROM modeling environment with PROMPTUM view plugin (Coskuncay et al. 2016) is given in Appendix D.

The ProModGen, requires user assistance and is composed of three processes which is given in Figure 7, and described in the following sections. This transformation will reduce the time and effort required for process modeling as well as process model elements would be stored on PROMPTUM Ontology Server as ontology resources.

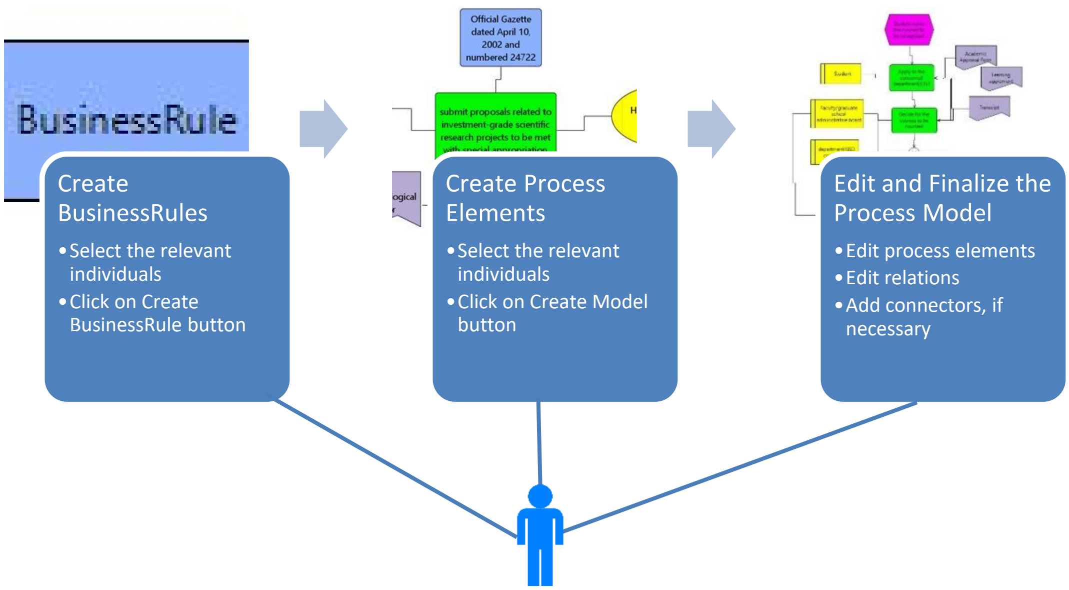


Figure 7 High Level Processes of ProModGen

3.3.1 *Retrieve the Process Ontology*

The process ontology instances can be uploaded to ProModGen as shown in Figure 8 by clicking on the Show Ontology button. These instances are listed as ordered in the document. Therefore, the modeler does not need to decide the order in which the process activities are executed.

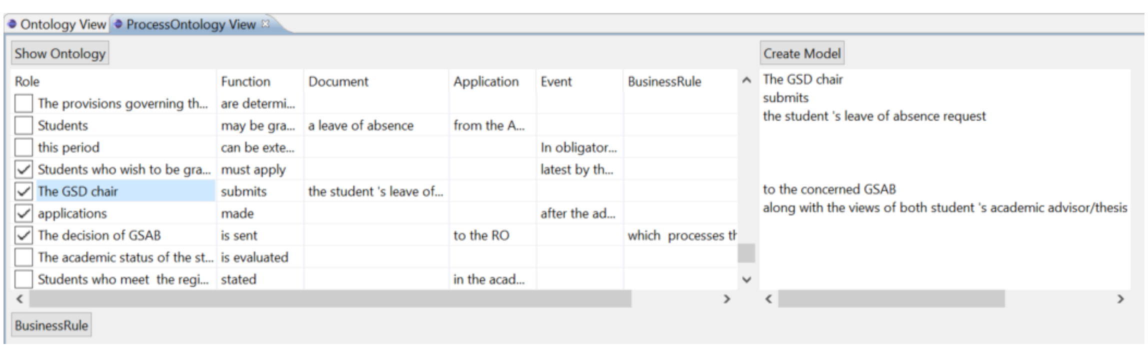


Figure 8 Excerpt from ProModGen

3.3.2 *Creating Business Rule Elements*

User can click on a row and preview the individuals on the right side as shown in Figure 9. If the user consideres that these individuals should be presented as a sentence in the Business Rule element, s/he can check the related row box and click on the BusinessRule button. This will keep all the individuals stored as a sentence under the Business Rule and will be created as Business Rule.

3.3.3 *Creating Process Elements*

This process can be illustrated as highlighting a sentence from the text that is being read that should be in the process model. The user needs to decide which instances to use in the process model. As s/he previews every row shown in Figure 9, s/he selects those which are relevant by clicking the checkbox. The user can also include the business rule class instances, which are created in the previous step. The user can create only one process model at a time. S/he should follow the same steps for every process model generation. Therefore, the user should decide which process model is to be created, and select the instances accordingly. From one organizational guideline, one process ontology is created but from one organizational guideline, more than one process model can be created.

3.3.4 *Generating Business Process Model*

After selecting the instances, the user can click on the Create Model button. The tool interprets the instances and according to their types, it automatically creates the corresponding model elements in the UPROM modeling environment as shown in Figure 9. The rule for the process element creation is based on extracting the subject and the who answer as Roles, the where-Location answer as Location and the remainder of the sentences as Function. Additionally, if there is an answer for the where-Document, an Information carrier is created. How, why and when questions are also created as Business Rules. Consequently, the model elements are generated from left to right according to their order in the document.

3.3.5 *Editing and Finalizing the Process Model*

To complete the process model, it may be necessary to delete unnecessary model elements, add certain model elements such as events, functions and logical connectors, and rename the model elements, and these actions require user assistance. For instance, functions having passive verbs should be converted to the active and present voice and events and connectors should be inserted. Figure 9 shows an example where the Role element is generated as “to the concerned GSAB” and requires to be renamed as “concerned GSAB”. The Function name “is sent…” should be renamed as “send the decision…”.

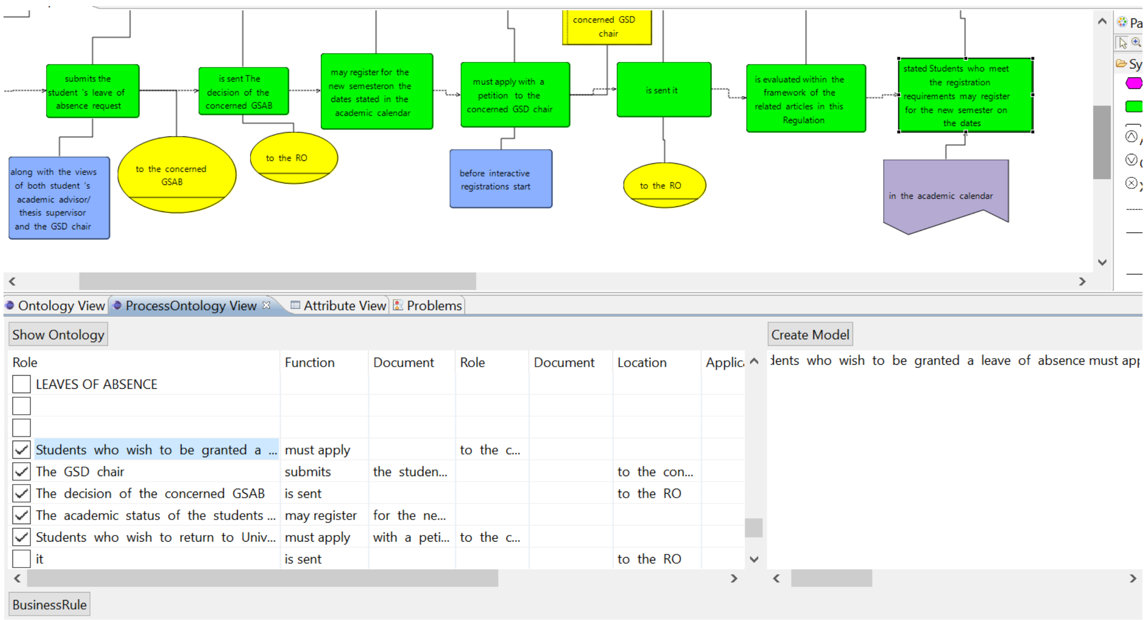


Figure 9 First version of the generated process model

CHAPTER 4

CASE STUDIES AND EXPERIMENT

In this chapter, the exploratory studies, multiple case and an experiment are performed in order to explore, generalize and validate the PrOnPo methodology and tool.

Briefly, in high level, the PrOnPo methodology and tool for process ontology development from organizational guidelines which are presented in Chapter 3, has been analyzed in four dimensions such as:

- Time and effort: Does PrOnPo decrease the time and effort?
    - Addressed in Exploratory Study 2 in section 4.1.5
    - Addressed in Multiple Case Study in section 4.2
    - Addressed in Experiment in section 4.3
- Coverage: Does PrOnPo cover all the relevant information in the guideline?
    - Addressed in Exploratory Study 2 in section 4.1.5
    - Addressed in Multiple Case Study in section 4.2
    - Addressed in Experiment in section 4.3
- Language Impact: Does another language effect its applicability?
    - Addressed in Exploratory Study 1 in section 4.1.4
- Adaptation: Is PrOnPo applicable on different types of guidelines from different domains?
    - Addressed in Multiple Case Study in section 4.2

Similarly, the business process model generation (ProModGen) which is presented in Chapter 3 for in section 3.3, has been analyzed three dimensions.

- Applicability: Is process model generation applicable on a real case?
    - Addressed in Exploratory Study 2 in section 4.1.5
- Time and effort: Does process model generation decrease the time and effort?
    - Addressed in Multiple Case Study in section 4.2
    - Addressed in Experiment in section 4.3
- Coverage: Does process model generation cover all the relevant information in the guideline?
    - Addressed in Multiple Case Study in section 4.2
    - Addressed in Experiment in section 4.3

In the following sections Exploratory Studies, Multiple Case Study and Experiment's design and results are presented respectively. The overall discussion of these studies are given in the last section of this chapter.

4.1 Exploratory Studies

In this section two exploratory studies are presented. The first exploratory study (Gurbuz & Demirors 2017a) is conducted in order to explore the process ontology development and identify automation requirements. This study led to the first implementation of PrOnPo. After the application of the PrOnPo on Turkish and English guidelines the automation potentials are identified and PrOnPo tool is implemented.

The second exploratory study (Gurbuz & Demirors 2017b) is performed in order to investigate the PrOnPo tool in terms of precision and recall metrics for creating relevant triples. ProModGen plugin is also analyzed for creating process models.

This section continues with the Research Questions, Validity Threats and Case Selection for exploratory cases. The design and execution of each exploratory study is given separately in respective sections.

4.1.1 *Research Questions*

Research Question 1: How to discover and develop process ontology from organizational guidelines with least time and effort?

The aim is to define an automated approach for process ontology discovery from natural language written text. The activities required for extracting process activities, roles, documents, applications and other process elements and relating them with each other will be explored and defined.

Research Question 2: What is the automation potential of process ontology development for Turkish and English written guidelines?

PrOnPo prototype will be applied on Turkish and English guidelines in order to analyze the effect of underlying technologies used as well as the effect of language.

Research Question 3: Is PrOnPo tool efficient in terms of time and effort and effective for extracting the relevant process information?

Validation method for the Research Question 3: The required time and effort will be compared for both manual process ontology development and using the PrOnPo tool. The recall and precision metrics will be used to compare the developed process ontology with the manual process ontology.

Research Question 4: Is ProModGen plugin for process model generation from process ontology applicable on a case?

Validation Method for the Research Question 4: The process models of the exploratory studies will be created using ProModGen plugin. The process models will be analyzed according to the number of process elements created and improvement potentials will be identified.

4.1.2 *Mitigation of Threats to Validity in Exploratory Studies*

The executer needs to have the capability to discover the potential improvements of the method and to evaluate the process ontology. Hence, the cases will be selected according to the executer's domain knowledge and familiarity. The evaluation of the developed process ontology will create opportunity to improve the method. In order to prevent any oversight in the evaluation phase due to exhaustion, we limited the scope of the ontology to be developed, to the selected case's guideline.

Regarding to the comparison of the application of the PrOnPo prototype, in order to prevent domain bias, we selected three guidelines from two domains, which the two guidelines are the same documents but one written in Turkish and one written in English, and the third document is written in Turkish but from a different domain than the other two. We selected most widely used natural language parser for English, Stanford Parser (Manning et al. 2014), and for Turkish, Zemberek (Akın & Akın 2007).

4.1.3 *Case Selection Criteria*

Table 4 Summary of Selected Cases for Exploratory Cases

Case	**Guideline for Investment Program Preparation**[5]	**Directive for Exchange Programs**[6]	**Regulation for Graduate Studies**[7]
#	**Exploratory 1**	**Exploratory 1**	**Exploratory 2**
Focus Organization	Ministry of Development	Middle East Technical University	Middle East Technical University
Ecological Validity	Good	Good	Good
Complexity	Fair	Fair	Good
Number of Documents	1	1	1
Number of sentences	412	159 in English, 162 in Turkish	432
Language	Turkish	English and Turkish	English

- ✓ Selected guidelines should be one of the following; procedures, regulation, policy, directive or official website
- ✓ The selected guideline shall include information for at least one process model.
- ✓ The selected cases shall reflect the real-life context and problem complexity should at least be fair.
- ✓ The selected cases shall be ecologically valid.

---

[5] http://www.kalkinma.gov.tr/Lists/YatirimProgramiHazirlamaEsaslari/Attachments/33/2014-2016YPHR.pdf

[6] http://oidb.metu.edu.tr/en/middle-east-technical-university-directive-exchange-programs

[7] http://oidb.metu.edu.tr/en/middle-east-technical-university-rules-and-regulations-governing-graduate-studies-0

This Exploratory Study is accepted as full paper by 43rd Euromicro Conference on Software Engineering and Advance Applications to be presented in August 30th (Gurbuz & Demirors 2017a).

- ✓ Three guidelines shall be chosen in which;
    - ▪ They must be from two different domains
    - ▪ The author must be familiar with
    - ▪ One of the selected domain should have the same guideline written in Turkish and English.
    - ▪ The language of these documents will be English and Turkish. Preferably, we will use the same guideline written in English and Turkish and guidelines written by native speakers.

According to these criterions, the selected cases for Exploratory Cases are given in Table 4.

4.1.4 *Exploratory Study 1*

This exploratory study is performed in order to;

- Investigate the manual development of process ontology from organizational guidelines.
- Define the requirements for automated process ontology development
- Explore and compare the application of the PrOnPo on two different languages

The output of this study is;

- ✓ How to discover and develop process ontology from organizational guidelines with least time and effort?
    - o Requirements for automated process ontology development
- ✓ The PrOnPo prototype
- ✓ What is the automation potential of this approach for Turkish and English written guidelines?
    - o The comparison results of the application of PrOnPo prototype on guidelines written in Turkish and English.

4.1.4.1 *Design in Exploratory Study 1*

The following activities are designed to perform the exploratory studies.

Case Study Environment; This case study will include ontology building. Protégé[8] will be used for modeling the ontology. For the manual extraction phase, process elements and their relations will be stored in MS Excel. For automated development, PrOnPo prototype will be used.

Identifying Information Sources; Guidelines selected according to the case selection criteria will be the input for this study.

Defining the scope of the Ontology; The scope of the ontology is limited to the information presented in the selected guidelines. We aim to develop process ontology regarding to the given

---

[8] http://protege.stanford.edu/

information in the guideline. Additionally, Competency Questions are defined as given in Table 5.

Table 5 Competency Questions

Competency Questions
What are the activities of this domain?
What are the roles which are executing the activities?
What are the input and output documents of these activities?
What are the information systems used for executing the activities?
What are the business rules for executing the activities?

Identifying the automation potential of process ontology development; Process elements and their relations will be discovered manually throughout both guidelines and will be recorded in MS Excel. The types of the process elements (function, event, information carrier etc.) will be used for creating ontology classes in Protégé. Afterwards extracted process elements in Excel sheet will be stored as ontology class individuals under the related classes in Protégé. During this phase, the requirements for the first version of the PrOnPo will be defined.

Implementation of the PrOnPo prototype; According to the requirements defined in the previous step, the PrOnPo prototype is implemented.

Application of the PrOnPo; After the development of the PrOnPo prototype, the process ontology will be developed using the prototype.

Evaluation of the Ontology; Hazman & Rafea defines two types of ontology evaluation (Hazman et al. 2011). The first one is ontology content evaluation which is for avoiding inconsistent, incorrect and redundant ontologies. The second one is ontology technology evaluation which ensures that the technology can be easily integrated to other software environment. According to Hazman, most popular evaluation method is human evaluation, which answers how the ontology meets a set of predefined standards or requirements by identifying with user involvement. For this approach, the best known is defining Competency Questions (Uschold & Gruninger 1996) at the beginning. If the developed process ontology can answer the competency questions, we can classify it as sufficient. Therefore, we will evaluate the content of the manually developed process ontology using competency questions given in Table 5.

Comparison of the applications; The output of the tool will be compared with the manually developed and evaluated ontology. For this comparison, recall and precision metrics will be used (Brewster et al. 2004). These metrics are known to be used for information retrieval process in order to define the percentage of the correct and relevant retrievals. Finally the recall and precision metrics of both execution results will be compared.

4.1.4.2 *Case Study Execution*

4.1.4.2.1 *Identifying the Automation Potential for Process Ontology Development*

We followed the given steps in order to develop process ontology development from organizational guidelines. According to these steps, the requirements for the implementation of the PrOnPo is also defined.

We defined a meta-model relations for each process element as shown in Figure 10. The documents are read through and all the sentences which are containing the elements that are given in the meta-model are extracted to Excel as shown with a motivating example in Table 6.

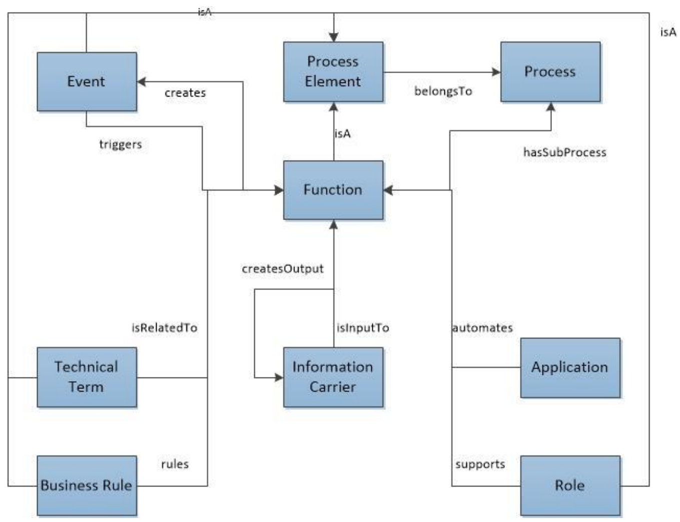


Figure 10 Initial Version of Meta-model for EPC elements and relations

Table 6 Sample Extracted Sentence

Sentence	**Translation**
10 Nisan 2002 tarih ve 24722 sayılı Resmi Gazete'de yayımlanan Yükseköğretim Kurumları Bilimsel Araştırma Projeleri Hakkında Yönetmeliğin 12'nci maddesi uyarınca yüksek öğretim kurumları, özel ödenekle karşılanacak yatırım nitelikli bilimsel araştırma projeleriyle ilgili tekliflerini de Teknolojik Araştırma Sektörü teklifi içinde Kalkınma Bakanlığına sunacaklardır.	In accordance with Article 12 of the Regulation on Scientific Research Projects of Higher Education Institutions published in the Official Gazette dated April 10, 2002 and numbered 24722, higher education institutions will submit their proposals related to investment-grade scientific research projects to be met with special appropriation to the Ministry of Development within the proposal of Technological Research Sector.

Afterwards, each of these elements in the sentences are copied to another excel sheet as shown in Table 7. The translation column in the following tables are given for the sake of the reader. The ones which they have relation to the other elements (this information is interpreted from the sentences) are written on the next corresponding column as shown in Table 8 . The relation types of these elements are specified according to the meta-model. We defined the process elements as ontology classes and their relations as object properties on Protégé. Finally, we used these extracted instances of the process elements to populate the upper level ontology defined in Protégé as shown in Figure 11.

Table 7 Identifying Process Elements

Process Elements	**Translation**	**Type**
Yükseköğretim Kurumları Bilimsel Araştırma Projeleri Yönetmeliği	Regulation on Scientific Research Projects of Higher Education Institutions	Information Carrier
10 Nisan 2002 tarih ve 24722 sayılı Resmi Gazete	Official Gazette dated April 10, 2002 and numbered 24722	Business Rule
Yüksek Öğretim Kurumları	Higher Education Institutions	Role
özel ödenekle karşılanacak yatırım nitelikli bilimsel araştırma projeleriyle ilgili tekliflerini sunacaklardır	will submit their proposals related to investment-grade scientific research projects to be met with special appropriation	Function
Teknolojik Araştırma Sektörü teklifi	proposal of Technological Research Sector	Information Carrier
Kalkınma Bakanlığına	Ministry of Development	Role

Table 8 Relations between the Process Elements

Instances	**Relations**	**Instances**
Yüksek Öğretim Kurumları	supports	özel ödenekle karşılanacak yatırım nitelikli bilimsel araştırma projeleriyle ilgili tekliflerini sunacaklardır
Teknolojik Araştırma Sektörü teklifi	is input to	özel ödenekle karşılanacak yatırım nitelikli bilimsel araştırma projeleriyle ilgili tekliflerini sunacaklardır
Kalkınma Bakanlığına	supports	özel ödenekle karşılanacak yatırım nitelikli bilimsel araştırma projeleriyle ilgili tekliflerini sunacaklardır
Yükseköğretim Kurumları Bilimsel Araştırma Projeleri Yönetmeliği	is input to	özel ödenekle karşılanacak yatırım nitelikli bilimsel araştırma projeleriyle ilgili tekliflerini sunacaklardır
10 Nisan 2002 tarih ve 24722 sayılı Resmi Gazete	rules	özel ödenekle karşılanacak yatırım nitelikli bilimsel araştırma projeleriyle ilgili tekliflerini sunacaklardır

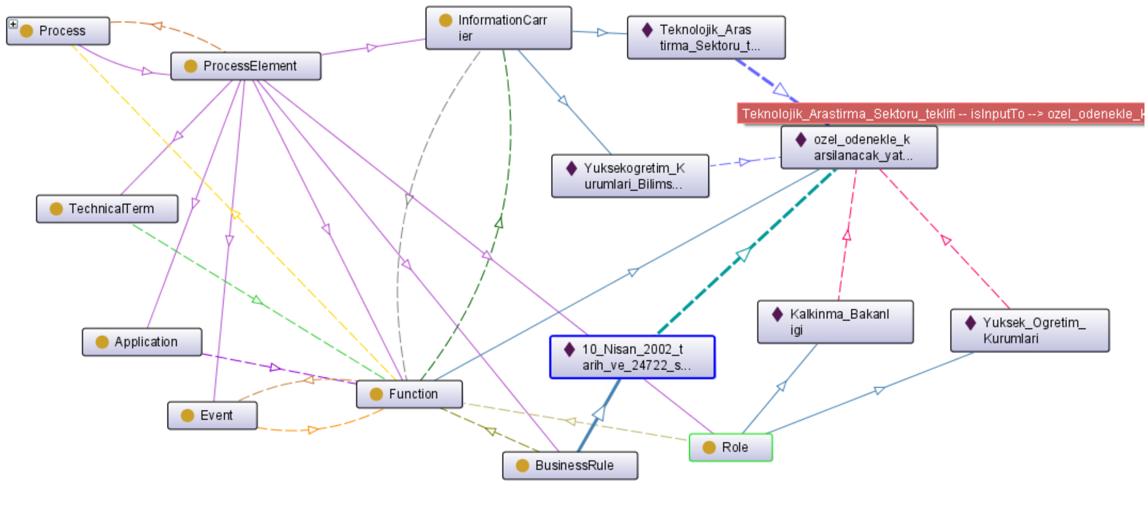


Figure 11 Onto-Graph Excerpt from Guideline for Investment Program Preparation[9]

4.1.4.2.2 *High Level Requirements for the Automation*

We identified following requirements for the automation of process ontology development.

Sentence extraction; Sentences shall be extracted using parsers.

Process element extraction; Since the different parsers will be used for Turkish and English process ontology development, requirements are defined separately.

English guidelines; Sentences shall be parsed into subject-verb-object form using natural language parsers. Stanford Parser creates the output of the parsed sentences in Tree data structure. The verb of the sentence is the child of the subject and the object is the child leaf of the verb. For English text, subject-verb and object can be extracted. Since the object may contain more than one process element and type, the object needs to be partitioned into smaller parts. 5W1H (what, where, when, why, who and how) questions will help to partition the sentences. Rules shall be defined according to the parser tags of the noun phrases and object can be partitioned according to the rules. The sample table of rules (for English) is given in Table 9.

Table 9 Initial version of template for partitioning the object

What	**Where**	**Why**	**When**	**Who**	**How**
No (PP) only (NP) as object	(IN)- on	(TO)+(VB)/(VBG) (ex: to obtain, regarding)	(IN)-after, before, following	(IN)-by (TO)-to Role tag	(IN)-by+(VBG) (ex: by applying) (IN)-with

[9]http://www.kalkinma.gov.tr/Lists/YatirimProgramiHazirlamaEsaslari/Attachments/33/2014-2016YPHR.pdf

Turkish guidelines; Zemberek only gives the root, the type and the suffixes of the words (ex: [Kok (Root): bakan, ISIM (Noun)] Ekler (Suffixes): ISIM_BULUNMA_LIK + ISIM_SAHIPLIK_O_I + ISIM_YONELME_E). In order to divide the sentences we can create two groups. The first group will contain the subject and object and the second group will contain the verb. The first group can be created by selecting all the words that are next to each other until the next word is a verb. The second group can be created by selecting the word with the "verb" tag. For separating the subject and object from the first group, commas can be used. Adverbs and conjunctions can help partitioning the object. But user guide is necessary for correct partitioning and matching the partitions to the correct elements.

Afterwards, process elements can be discovered using the model in Figure 12 for matching the answers of these questions to the process elements. Note that user guide is necessary for correcting the irrelevant matches and incorrect parsing and partitioning.

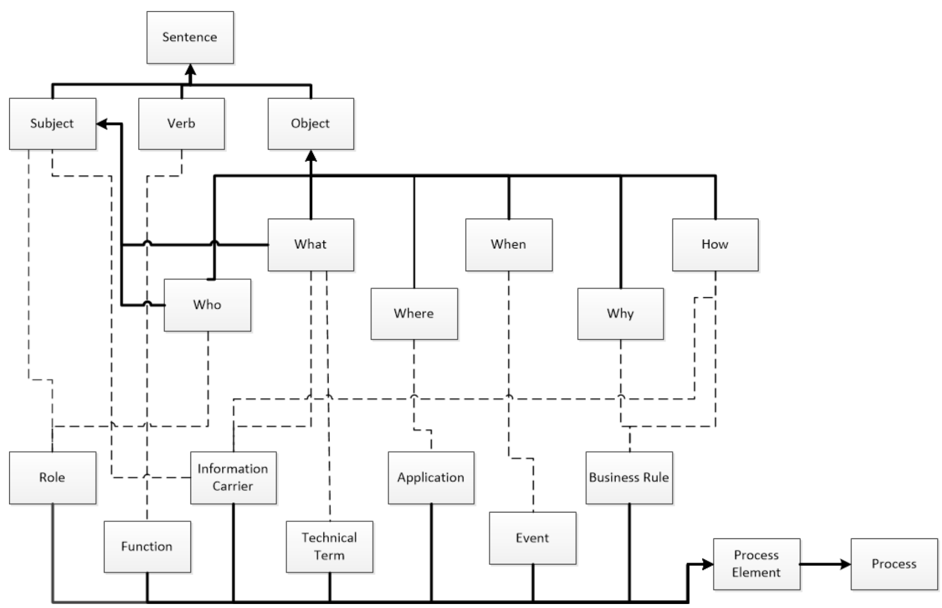


Figure 12 Initial Version of Meta-Model for matching Process Elements with Questions

Identifying the relations; The upper process ontology classes according to the meta-model is defined as given in Figure 10 in Protégé. Afterwards each extracted process element is defined as ontology individual to the related process element classes.

The RDF of the ontology classes we defined in Protégé, can be given as input to the tool. Once the sentences are partitioned automatically or manually, a simple algorithm can be implemented which can automatically define these partitions as individuals to the related classes.

4.1.4.2.3 *The PrOnPo Prototype*

According to the requirements identified in previous subsection, we implemented a prototype in Java to support our approach. The prototype can either be used full automated or semi-automated. The reason for semi-automation option is that the structure of the sentences have an impact on the quality of the output, thus might require user assistance. The effectiveness of both approaches (full and semi-automated) will be discussed in section 4.2. Although the extended description of the PrOnPo tool is further described in Chapter 3 section 3.2, the phases of the prototype is given in Table 10. The prototype is composed of sentence extraction, natural language parser (automated), object partitioning (user guided – semi-automated) and process ontology transformation (user guided – semi-automated). Zemberek (Akın & Akın 2007) is used for supporting Turkish language parsing and Stanford Parser (Manning et al. 2014) is used for supporting English language parsing. Stanford Parser is chosen as it is the most widely used tool for English (van der Aa et al. 2016) and Zemberek is the most widely used for Turkish.

Table 10 The PrOnPo Prototype High Level Phases

Phases	**Input**	**Output**
Tag Word Identification and Sentence Extraction	Organizational Guideline(s)	Extracted Sentences
Process Element Identification	Extracted Sentences	Parsed and partitioned sentences =triples
Ontology Class Individual Creation	EPC Ontology Class RDF	Populated Process ontology RDF

4.1.4.3 *Exploratory Study 1 Execution Results*

The method and the prototype has been applied on the selected documents in Turkish and English. Turkish comes from Ural-Altaic language family in which the morphology is mostly agglutinative and suffixing. The word order in Turkish is as subject-object-verb. Sometimes, subject can be hidden in the verb and interpreter figures it out by the suffixes of the verb. Consequently, with respect to English, it has a more complex organizational structure.

First of all we practiced the method manually on both documents. After conforming that the developed ontologies answer the competency questions defined in Table 5, we used these ontologies in order to evaluate the practice of the method by using the method and the tool. According to this evaluation we formed Table 11 and calculated precision and recall metrics.

Table 11 Summary of the Extracted Triples

	Guideline of Investment Program Preparation	Directive for Exchange Programs	
Language	Turkish	English	Turkish
Number of Actual Triples	330	148	148
Number of Extracted Triples	587	130	163
Number of Correct Triples	89	114	12
Number of Missing Triples	241	34	132
Precision	89/587 = 15%	114/130 = 88%	12/163 = 8%
Recall	89/330 = 27%	114/148 = 77%	12/148 = 8%

From Guideline of Investment Program Preparation (written in Turkish) 330 triples (subject-verb-object) are extracted manually. These triples are formed from the sentences which are considered to be containing the process elements. Interpreting the document, extracting the sentences, dividing them into subject-verb-object form and partitioning the objects took 8 hours. The tool on the other hand, extracted 587 sentences and only 89 of them is correctly parsed into subject-object-verb form. 241 of the triples were missing in the output. 498 of the extracted sentences were irrelevant. Among all the relevant triples, the prototype could find 27% (recall = 89/330) of the relevant triples whereas among all the extracted triples 15% (precision = 89/587) is relevant and extracted correctly.

From Directive for Exchange Programs (written in English) 148 triples are extracted manually. Interpreting the document, extracting the sentences which contain the process elements and dividing them to subject-verb-object form and partitioning the object took 105 person-minutes. The tool extracted 130 triples and 114 of the triples are parsed correctly into the subject-verb-object form. 6 of the extracted triples are irrelevant and 10 of them are parsed incorrectly. According to manually developed ontology, 34 of the triples are missing. Among all the relevant triples, the tool can find 77% (recall = 114/148). From all extracted triples 88% (precision = 114/130) is relevant and extracted correctly. The automated process for extracting and dividing sentences, partitioning the object based on the rules (if possible) and human interpretation for correcting the triples took 22 person-minutes. The tool could partitioned 69 of the objects of the sentences according to Table 9 and for 45 objects, user interpretation is needed.

For the Directive for Exchange Programs written in Turkish, the natural language parser for Turkish (Zemberek) extracted 163 triples and 14 of them are parsed correctly into the subject-verb-object form. 6 of the extracted triples are irrelevant which the 2 of them are parsed correctly. Therefore among the all extracted triples (163), 12 of them are relevant and parsed correctly, thus this makes the precision 8%.

4.1.4.4 *Exploratory Study 1 Discussion*

According to the results, the effectiveness of the PrOnPo prototype is significantly larger for English than the Turkish documents. This significant difference has two reasons. Firstly, Zemberek is a shallow parser whereas Stanford Parser is a full parser (Manning et al. 2014). Unfortunately there was no choice for full parsing in Turkish. Secondly, due to Turkish's complex grammar and suffixing morphology, tagging the words, finding the subject and grouping the noun phrases pose significant challenges. During the comparison of Turkish and English parsing, the following issues are identified.

- Zemberek cannot tag the words 100% correctly. For instance, a word which the root is a verb is changed to a noun with suffixes. But Zemberek, still, tags it as verb.
- When the subject of the sentences is hidden in the verb, Zemberek cannot identify it and thus cannot form a triple.
- For both languages, rules should be defined for extracting the subject, object and verb. Stanford gives the parsing output in a Tree structure in which subject is the root, verb is the child of the subject and object is the child of the verb. This structure enables dividing the sentences as well as to partitioning the objects. For Turkish, Zemberek only gives the tags of the words, therefore rules for partitioning the sentences does not apply like it does in English.

- For Turkish, defined rules caused more triples to be extracted than it should.
- For both languages, long and complex sentences cannot be effectively handled by the defined methodology.
- None of the parsers enable identification of punctuations and statements like and/or.
- None of the parsers can be used effectively when there are more than one verbs in the sentences. And thus this explains most of the missing triples.

The following improvement opportunities are identified;

- Object partitioning is not very effective and the Tree structure should be further studied in order to reduce user assistance.
- Process elements and partition matching should be reviewed.

Considering these results, the PrOnPo prototype is extended and PrOnPo tool is implemented for English language. The PrOnPo methodology and the tool are discussed in the previous chapter (Chapter 3).

4.1.5 *Exploratory Study 2*

After the Exploratory Study 1, PrOnPo prototype is extended as a tool and which is presented in Chapter 3. In order to explore the PrOnPo methodology and the tool, second Exploratory Study is performed. This study addresses the following research questions;

Research Question 3: Is the PrOnPo methodology efficient in terms of time and effort and successful for extracting the relevant information?

Research Question 4: Is ProModGen plugin for process model generation from process ontology applicable on a case?

The mitigation of validity threads and case selection criteria for this study are given in 4.1.2 and 4.1.3 respectively. In the following subsections, design and results of the study are presented.

4.1.5.1 *Case Study Plan and Execution*

4.1.5.1.1 *Analyzing the Selected Case*

Regulation for Graduate Studies is selected for this case considering the case selection criterions in section 4.1.3. The content summary of the regulation is given in Table 12. The document content is categorized as 4 parts; terms, activities, business rules and irrelevant sentences. We considered each parsed sentences as triples. We identified 421 relevant triples to be retrieved.

4.1.5.1.2 *Evaluation of the PrOnPo Methodology*

We compared the number of extracted and correctly parsed triples (sentences) with the manually identified triples using precision and recall metrics.

4.1.5.1.3 *Process Ontology Development with PrOnPo*

The following steps will be performed for process ontology development (Refer to section 3 for detail);

1. Give the document as input to the PrOnPo
2. Analyze the output according to the manual process ontology
    a. Record the number of parsed sentences
    b. Record the number of corrupted sentences (the sentences which are incorrectly parsed)
    c. Record the number of missing sentences

4.1.5.1.4 *Process Model Generation with ProModGen*

The following steps will be performed for process model generation (Refer to section 3.3 for detail);

1. Upload the process ontology to ProModGen
2. Select the related triples
3. Generate process model
4. Edit and finalize the process model

4.1.5.2 *Results Regarding the Research Question 3*

According to the frequency analysis, the algorithm selected the top 5 frequent nouns; student, program, course, thesis, semester. These keywords extracted 385 sentences as shown in Table 12. The parser created 352 triples from these sentences where 33 of the sentences were corrupted during parsing (in other words, incorrectly parsed therefore triples were not formed properly and can also be considered as missing). 11 of the parsed sentences are parsed in incorrect form and 8 of the sentences are irrelevant. Consequently, 333 triples are generated which are relevant and parsed correctly and therefore considered as true positives. To prevent the validity concerns about the keywords usage, we also applied the method without using the keywords of which the same details are shared in Table 12.

In light of this data, we created the confusion matrix as shown in Table 13. Correctly parsed sentences are considered as true positives, which are retrieved and relevant. Incorrectly parsed sentences and irrelevant sentences are considered as false positives, which are retrieved but irrelevant. According to the manually developed process ontology, the missing triples (corrupted/not extracted) are considered as false negatives, which are not retrieved but relevant. It should be noted that this correct triples only cover the parsing the sentence into subject-verb-object form and discards the correctness of partitioning the object, which will be analyzed in the next section (Multiple Case Study).

Table 12 Summary of the Graduate Study Regulation

Tagged				**No Tag**		
Total number of sentences and types	438	34 term		438	34 term	
		93 activities			93 activities	
		294 business rule			294 business rule	
		17 irrelevant			17 irrelevant	
Total number of extracted sentences	385	34 term		438	34 term	
		93 activities			93 activities	
		248 business rule	46 not extracted		294 business rule	
		10 irrelevant	7 not extracted		17 irrelevant	
Total number of correctly parsed sentences	352	34 term		391	34 term	
		83 activities	10 corrupted		83 activities	10 corrupted
		227 business rule	21 corrupted		263 business rule	31 corrupted
		8 irrelevant	2 corrupted		11 irrelevant	6 corrupted
Total number of correctly parsed and relevant sentences (True Positives)	333	34 term		366	34 term	
		82 activities	1 incorrect parse		82 activities	1 incorrect parse
		217 business rule	10 incorrect parse		250 business rule	13 incorrect parse

According to the Table 13, we calculated the recall and precision metrics as shown in the below formulas and presented the results in Table 14. Recall and precision are used for evaluating information retrieval and natural language processing tasks (Brewster et al. 2004). Recall metric calculates the correctly identified triples over all the relevant triples whereas precision metric calculates the correctly identified triples over all the retrieved triples. The comparison between extracting sentences with and without keywords shows that the precision is a little high when using keywords which means that the proportion of correct triples to all retrieved triples is high. On the other hand, recall is higher without keywords which shows that without keywords we can achieve higher success of retrieving relevant triples over all the relevant triples. Note that, this comparison should further be analyzed with other cases in order to come up with a conclusion.

True Positive / (True Positive + False Positive) = Precision.

True Positive / (True Positive + False Negative) = Recall.

In terms of time, manual development took 200 person-minutes which includes reading the document, interpreting the sentences and partitioning. The tool has reduced this time to 50 person-minutes, which includes the extracting, parsing and the interpretation for the non-partitioned objects. 1134 class individuals are created and only 152 of the individual partitions is replaced by the user assistance.

Table 13 True-False Matrix

With Tags	*Relevant (333+88=421)*	*Irrelevant*
Retrieved (333+19=352)	True positives = 333	False positives= 8+11
Not Retrieved	False negatives= 88	0
Without Tags	*Relevant (366+55=421)*	*Irrelevant*
Retrieved (366+25=391))	True positives = 366	False positives= 11+14
Not Retrieved	False negatives= 55	0

Table 14 Process Ontology Discovery Results

Case Execution Steps	**Graduate Studies**		
Tag words	student, program, course, thesis, semester		
Extracted sentences	With tags	385	
	Without tags	438	
Parsed sentences	With tags	352	
	Without tags	391	
Evaluating the process ontology	With tags F-measure= 0.86	Precision	95% (333/352)
		Recall	79% (333/421)
	Without tags F-measure= 0.90	Precision	94% (366/391)
		Recall	87% (366/421)
Time comparison	Manual	200 person-minutes	
	PrOnPo	50 person-minutes	

4.1.5.3 *Results Regarding the Research Question 4*

Using the process ontology 8 process models are generated as given in Table 15. This generation is assisted by the user, in the process instance choosing and finalizing phases. However, generating process models by choosing related individuals, creating and finalizing the model required 85 person-minutes. The process models are given in the Appendix E.

In the given Table 15, two rows for each process model are created: the first row presents the total number of each element in the process model and the second row presents the number of process elements inserted after generating the process model, for finalizing. As it is seen, all of the events and connectors are inserted afterwards to the initial models.

It is noted that if the process information in the guideline is very descriptive, the ProModGen is effective at reflecting the process on the model. In this example, PhD Process and Masters Process are the two longest processes which the steps are explicitly defined in the guideline. ProModGen required 20 person-minutes to reflect the process on the model with minimum insertions.

Table 15 Generated Process Models of Graduate Studies

Total Number of **Process Name**	Functions	Events	Documents	Connectors	Applications	Roles	Business Rules
Academic Deficiency Process	3	2		2		1	
Number of Inserted Elements		2		2			
Application Process	3	4		1		2	2
Number of Inserted Elements	1	4		1			2
Student Leave of Absence Process	8	9		3		10	1
Number of Inserted Elements	1	9		3			
Master with Non-Thesis Program Process	6	4	1	1		3	1
Number of Inserted Elements	4	4	1	1		1	
PhD Program Process	18	15	2	7		21	1
Number of Inserted Elements		15	2	7			
Registration Process	3	2	1			3	1
Number of Inserted Elements	1	2	1			1	
Semester Registration Process	6	4	1	1	1	8	3
Number of Inserted Elements	1	4		1			
Master Program Process	16	8	2	5	1	17	
Number of Inserted Elements	4	8		5	1		

During this process we have noted the following issues;

- It is hard to create process element when the sentences are not descriptive or explicit and human interpretation requires
- When the verb is passive, the name of the process element should be edited
- The placement of the process element on the diagram is not controllable, therefore replacing it also requires time.
- According to the process information given in the guideline, the ProModGen can create models which consists very low level detail or high level descriptive information.

4.2 Multiple Case

A multiple case study is performed in order to analyze the adaptation of the PrOnPo tool on different domains using different types of guidelines and the ProModGen efficiency and coverage. This study addresses the following research questions;

Research Question 5: What is the adaptation of the PrOnPo methodology in different domains with different types of organizational guidelines?

Validation method for the Research Question 5: The PrOnPo is used to develop process ontology from organizational guidelines selected from five different domains with two treatments; fully automated and semi-automated. The results of both treatments are compared with the manually developed process ontology using precision and recall metrics. The time and effort required in the semi-automated development are compared with the manual development.

Research Question 6: Does creating process models from process ontology reduce the required time and effort and affect the coverage of the process models?

Validation method for the Research Question 6: The process models generated by ProModGen are compared with the existing process models in terms efficiency and coverage. The efficiency is evaluated based on the time effort required for process modeling. The coverage is evaluated by comparing the number of process elements in both models.

4.2.1 *Mitigation of Threats to Validity in Multiple Case Study*

We performed the following actions to prevent any threats that could affect the internal, construct and external validity, and reliability (Wohlin et al. 2012). Internal validity can be determined by comparing the relation between two factors while ignoring a third factor that can affect the comparison. For this case study, one potential threat is the comparison of the process models which are created by the same modeler. In order to prevent this, process models must be created by a different analyst to compare the manual and automated process modeling. Additionally, the length of the text in the selected guidelines could also affect the internal validity. Thus, we limited the length of the documents, in order to prevent any bias that could result from the exhaustion of the analyst during the process ontology development and evaluation. On the other hand, in order to avoid another potential threat regarding the length selection, we increased the number of selected cases, thus preventing a mono-operation bias.

External validity concerns the generalizability of the study. We prevented this threat by conducting a multiple case study by collecting different types of guidelines from different domains. Finally, construct validity concerns what is being examined by the research questions and whether it is really reflected in the execution. This study focuses on the process ontology development from the selected guidelines. The focus of the research question is the extent to which the tool is successful in developing process ontology. Thus, during the execution, regarding the evaluation of the developed process ontology, competency questions are used to prevent any threat regarding the construct.

Lastly, reliability concerns the evaluation by another researcher repeating the case study. The aim of a multiple case study is to discover the generalizability of the developed tool regardless of the person who implements it. In order to avoid any validity threat regarding the reliability of the efficiency and effectiveness of the PrOnPo tool, we conducted an experiment with participants.

4.2.2 *Multiple Case Study Design*

This section presents the case selection, environment and scope and evaluation of ontology respectively.

4.2.2.1 *Case Selection Criteria*

We defined case selection criteria for performing the Multiple Case study and. The first aim is to analyze the applicability of PrOnPo to different types of guidelines such as; procedures, regulations, policies, directives, and official websites. The second aim is to analyze the adaptability of PrOnPo on different domains. Therefore, the cases must be selected from a different domain. For a scalable evaluation of the developed process ontology, the number of pages of the selected guideline must not exceed 10 pages. The selected guideline must include information for at least one process model. The selected cases must reflect the real-life context and problem complexity should at least be fair. The selected cases must be ecologically valid. The language of the selected cases must be in English, preferably written by a native speaker.

To support the case selection, we grouped the domains as given in Figure 13.

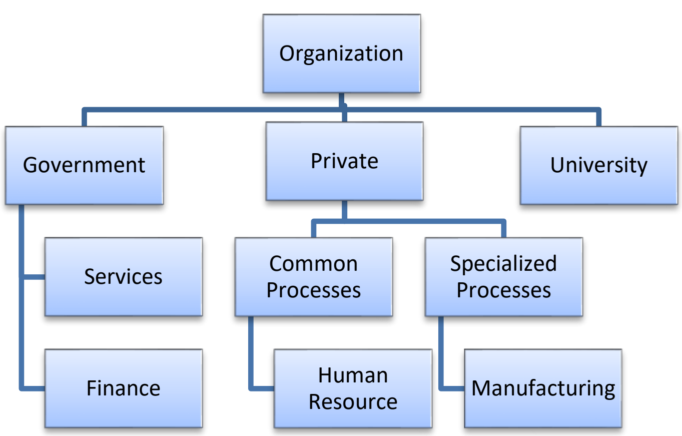


Figure 13 Domain Selection Summary

We grouped the type of organizations into; government, public, and university. We selected the registration process for university domain. This task was given as process modeling homework as part of the Business Process Management course. Therefore, a registration process model was available to be compared with the generated process model. We categorized the government into the most common processes; service and finance. The process information belonging to the government is usually not explicitly defined. Regulations and policies contain process information together with the related laws. Therefore, we searched the publically available guidelines, which explicitly define a process from the countries in which the native language is English. We observed that the most common processes belonging to government were visa and license applications. The private organization were categorized into common and specialized processes. For these organizations the most common processes were in the domain of human resources. For the specialized processes, we selected the manufacturing domain to generalize the application of PrOnPo. The following focus processes were selected according to the Domain Selection Summary diagram in Figure 13.

The cases that were selected according to the defined criteria are given in Table 16.

Table 16 Summary of the Selected for Multiple Case

Case Characteristics	**Multiple Case**				
	Case 1	***Case 2***	***Case 3***	***Case 4***	***Case 5***
Name	The Banking and Financial Institutions Regulation[10]	Family Reunification Policy[11]	Contractor Safety Procedure[12]	Leave Policy[13]	Registration of Graduate Students[14]
Type	Regulation	Policy	Procedure	Policy	Official Web-site
Focus Domain	Finance	Services	Manufacturing	Human Resources	University
Focus Process	Application and decision making for bank license	Family reunification visa approval process	Approval process, Orientation process, Training process	Application and monitoring of annual leave process	Registration process
Number of Pages	4	10	8	6	2
Number of Process Model	1	1	3	1	1
Real Life Context	Good	Good	Good	Good	Good
Problem Complexity	Good	Good	Fair	Good	Fair

10 http://www.bot.go.tz/BankingSupervision/documents/New%20Docs/The%20Banking%20and%20Financial%20Institutions%20(Licensing)%20Regulations,%202014.pdf

11 http://www.inis.gov.ie/en/INIS/Family%20Reunification%20Policy%20Document.pdf/Files/Family%20Reunification%20Policy%20Document.pdf

12 https://www.schneider-electric.us/documents/company/supplier-resource-center/P-P_Contractor_Safety.doc

13 https://www.nice.org.uk/media/default/about/who-we-are/policies-and-procedures/leave-policy.pdf

14 http://oidb.metu.edu.tr/en/registration-graduate-students-0

4.2.2.2 *Scope and Evaluation of the Ontology*

The scope of the ontology to be developed is limited by the process information presented in the selected guidelines. According to Hazman, the most popular evaluation method is human evaluation, in which it is determined whether ontology meets a set of predefined standards or requirements identified through user involvement (Hazman et al. 2011). The best known approach for this method is to define competency questions (Uschold & Gruninger 1996) at the beginning of the process. If the developed process ontology can answer the competency questions, it can be considered as adequate. Therefore, the content of the manually developed process ontology is evaluated using the competency questions given in Table 17. Additionally, the competency questions are used for defining the scope of the ontology.

Table 17 Competency Questions

Competency Questions
What are the activities of this domain?
What are the roles which are executing the activities?
What are the input and output documents of these activities?
What are the information systems used for executing the activities?
What are the business rules for executing the activities?
What are the terms relating to this domain?

4.2.3 *Multiple Case Study Implementation*

This section presents process ontology development, evaluation of the process ontology, and the process model generation.

4.2.3.1 *Process Ontology Development*

The process ontology is developed in three ways; manual, semi-automated (user-guided), and fully automated. For each development, time and effort information is collected.

4.2.3.1.1 *Manual Process Ontology Development*

For the selected cases, the competency questions are considered in order to extract the relevant information. For manual development, the following steps are followed: (Refer to section 4.1.4.2.1 for detail);

1. Analyze the document to identify the number of processes, business rules, terms, and irrelevant information it contains.
2. Extract the sentences which contain activity, role, input/output sources, information systems, business rules, or terms related to the selected domain.
3. Construct the relation of the extracted process elements by partitioning the sentence.
4. Match the partitions with the related process ontology classes.
5. Build the process ontology.
6. Investigate the process ontology according to the competency questions.

7. Record the number of ontology individuals that are required.

This manually developed process ontology was used as the baseline to evaluate the process ontologies developed by PrOnPO. The manual process ontology development is established using an Excel sheet. The output RDF of the developed process ontology from the tool is stored in Protégé 4.3.

4.2.3.1.2 *Full-Automated Process Ontology Development*

The following steps are followed for the full-automated process development (Refer to section 3.2.2 for detail):

1. Use the document as an input in PrOnPo.
2. Analyze the output of the tool according to the manual process ontology and record the number of parsed sentences, corrupted sentences, missing sentences, extracted ontology individuals, missing ontology individuals.

4.2.3.1.3 *Semi-Automated Process Ontology Development*

In addition to the steps given in the fully automated process ontology development, this process includes recording the number of replaced ontology individuals. (Refer to section 3.2.2 for detail). The semi-automated and fully automated process ontology development are both performed using the PrOnPo tool and methodology.

4.2.3.2 *Evaluation of the Process Ontology*

The semi-automated and full-automated process ontology individuals are compared using recall and precision metrics (Brewster et al. 2004) with the manually developed process ontology individuals. These metrics are utilized in the information retrieval process to define the percentage of the correct and relevant retrievals.

4.2.3.3 *Process Model Generation*

As part of the multiple case study, we also included process model creation. Registration process from university domain was modeled as part of a class work. In addition to university domain, we selected manufacturing domain since it had more processes according to other domains. Contractor approval, orientation and training processes are modeled by a different analyst using the guideline. We also created process models using ProModGen to compare the process models created by different sources in terms of time, effort and coverage. Furthermore, we also practiced the other domains' processes for process model creation on ProModGen. The pre-modeled process models are given in Appendix F.

4.2.4 *Results and Analysis Regarding the Research Question 5*

The results of the analysis are given in three dimensions; guideline, ontology individuals, and time. The metrics collected for the comparison are given as a true-false matrix and precision / recall results. The analysis regarding the guidelines is given in Table 18. The sentences presented in the guideline are grouped into four categories of irrelevant sentences, terms, business rules, and processes as discussed in section 3.2.1.1 of Chapter 3. Table 18 also contains the results of the analysis on the PrOnPo tool regarding the parsed and corrupted sentences. Number of sentences row shows the actual number of sentences in the guidelines. Parsed sentences row shows the number of sentences which could be parsed by the parser. Parsed and corrupted sentences row shows the number of sentences which were incorrectly parsed and therefore triples were not created and resulted to missing individuals. Lastly, missing sentences row shows that the number of sentences which could not be parsed by the parser, in other words forming subject-verb-object was unsuccessful.

Table 18 Summary of Selected Guidelines

		Manufacturing	**Visa**	**University**	**HR**	**Banking**
Number of Sentences	**Total**	**153**	**37**	**40**	**29**	**36**
	Irrelevant	7	1	0	2	0
	Terms	28	0	0	0	0
	BusinessRules	71	21	27	15	17
	Process	47	15	13	12	19
Parsed	**Total**	**129**	**27**	**39**	**26**	**32**
	Irrelevant	5	0	0	2	0
	Terms	27	0	0	0	0
	BusinessRules	58	12	26	14	15
	Process	39	15	13	10	17
Parsed but Corrupted	**Total**	**3**	**1**	**1**	**1**	**6**
	Irrelevant	0	0	0	1	0
	Terms	0	0	0	0	0
	BusinessRules	2	1	1	0	3
	Process	1	0	0	0	3
Missing	**Total**	**23**	**10**	**1**	**3**	**4**
	Irrelevant	2	1	0	0	0
	Terms	1	0	0	0	0
	BusinessRules	13	6	1	1	2
	Process	8	3	0	2	2

The numbers of the actual ontology individuals identified in the manual process ontology development is given in the first row of Table 19 and the successive rows give the number of extracted, replaced, missing and irrelevant individuals, respectively. Using these numbers, the true false metrics are created. The correctly retrieved individuals are considered as true positives and the retrieved but irrelevant individuals are considered as false positives. The missing individuals which were not retrieved but are relevant are considered as false negatives. The true false matrix summary given in Table 20 is created for the semi-automated and fully automated process ontology development. Regarding the fully automated matrix, the individuals which should be modified by the user are considered either as false positive since they are in the wrong place or as false negative since they are relevant but not retrieved.

Table 19 Summary of the Ontology Individual Analysis

	Manufacturing	**Visa**	**University**	**HR**	**Banking**
Number of Actual Individuals	486	134	164	126	168
Number of Individuals Extracted	448	100	157	114	136
Number of Individuals Replaced by User	52	17	28	24	32
Number of Missing Individuals	63	34	7	15	32
Number of Irrelevant Individuals	25	0	0	3	0

Table 20 True-False Matrix Summary (TP = True Positive, FP = False Positive, FN= False Negative)

	Manufacturing		**Human Resources**		**University**		**Visa**		**Banking**	
Semi	Relevant 486	Irrelevant	Relevant 126	Irrelevant	Relevant 164	Irrelevant	Relevant 134	Irrelevant	Relevant 168	Irrelevant
Retrieved	TP = 423	FP = 25	TP = 111	FP = 3	TP = 157	FP = 0	TP = 100	FP = 0	TP = 136	FP = 0
Not Retrieved	FN= 63		FN=15		FN = 7		FN = 34		FN = 32	
Full	Relevant 486	Irrelevant	Relevant 126	Irrelevant	Relevant 164	Irrelevant	Relevant 134	Irrelevant	Relevant 168	Irrelevant
Retrieved	TP = 371	FP = 25+52	TP = 87	FP = 3+24	TP = 129	FP = 28	TP = 83	FP = 17	TP = 104	FP = 32
Not Retrieved	FN= 63+52		FN= 15+24		FN = 7+28		FN = 34+17		FN = 32+32	

In the next step, the precision and recall metrics are calculated according to the formulas given below. The summary of the metrics are given in Table 21. The recall metric calculates the correctly extracted individuals over all the relevant individuals whereas the precision metric calculates the correctly extracted individuals over all the retrieved individuals. According to the precision and recall metrics, it is obvious that the precision and recall metrics for semi-automation are higher than full-automation. These results provide an insight concerning the effectiveness of the user-guide during the process ontology development. The success of the PrOnPo tool for retrieving the relevant individuals (recall) ranged from 75% to 96% with an average of 85%. Discarding the missing individuals, the success of the PrOnPo tool for retrieving relevant individuals over all the retrieved individuals (precision) ranged from 93% to 100% with an average of 98%. These results show that the PrOnPo tool and methodology are adaptable / applicable to different domains and guideline types.

True Positive / (True Positive + False Positive) = Precision.

True Positive / (True Positive + False Negative) = Recall.

Table 21 Summary of Precision and Recall Metrics

Case		**Precision**		**Recall**	
		TP/(TP+FP)		*TP/(TP+FN)*	
Manufacturing	**Semi**	423/448	94%	423/486	87%
	Full	371/448	83%	371/486	76%
Visa					
	Semi	100/100	100%	100/134	75%
	Full	83/100	83%	83/134	62%
University					
	Semi	157/157	100%	157/164	96%
	Full	129/157	82%	129/164	79%
Human Resources					
	Semi	111/114	97%	111/126	88%
	Full	87/114	76%	87/126	69%
Banking					
	Semi	136/136	100%	136/168	81%
	Full	104/136	76%	104/168	62%

The time and effort data is collected in terms of person-minutes. The summary of the time and effort required to perform the studies for the manual and semi-automated process ontology development are given in Table 22. The time and effort for semi-automation includes the process ontology development time of the PrOnPo tool and the replacement of the individuals. The total effort for manual development was 250 person-minutes whereas for the semi-automated development, it was 57 person-minutes. These results showed that the process ontology development with the PrOnPo methodology and the tool is applicable to different guidelines from various domains and more efficient than the manual process ontology development.

Table 22 Summary of the Time Analysis

	Type	**Manufacturing**	**Visa**	**University**	**HR**	**Banking**
Time	**Manual**	120 p-m	30 p-m	30 p-m	30 p-m	40 p-m
	Semi	15 p-m	12 p-m	10 p-m	10 p-m	10 p-m

4.2.5 *Results and Analysis Regarding the Research Question 6*

The summary of the comparison of the process models created by ProModGen to the process models created manually is given in Table 23. M stands for manual creation (existing models) and P stands for the ProModGen models. The required time and effort is recorded in person-minutes.

Table 23 Process Model Coverage and Time

Domain	**Manufacturing**						**University**	
Process Models	*Approve Contractor*		*Give Orientation*		*Train Contractor*		*Registration Process*	
Treatments	*M*	*P*	*M*	*P*	*M*	*P*	*M*	*P*
Notation	eEPC	eEPC	eEPC	eEPC	eEPC	eEPC	BPMN	eEPC
Time and Effort	38 p-m	10 p-m	23 p-m	14 p-m	8 p-m	6 p-m	3 people 12p-h	1person 23 min
Functions	5	7	8	6	2	3	30	9
Events	4	5	5	6	2	2	X	3
Business Rules	1	10	0	7	0	4	9	19
Information Carriers	9	3	5	X	2	X	14	5
Roles	4	8	9	5	3	3	2	11
Connector	2	3	2	3	0	X	12	1
Application	X	X	X	X	X	X	7	X
Total	25	36	29	27	9	12	77	48

In terms of time and effort, the process models created by ProModGen was significantly less in relation to manual process modelng. In terms of coverage, it can be seen from Table 23 that some elements are used more or less according to the existing models. The generated process models are given in Appendix G. We analyzed the process models according to their information coverage. We categorized the structure of sentences of the guideline for process model generation into three categories:

1. The sentences that reflect the process information in detail.
   - These sentences require minimum insertions in the finalizing phase of the process model.
2. The sentences which require interpretation to extract process information.
   - These sentences require maximum insertions in the finalizing phase of the process model.
3. The sentences containing high-level process information.
   - Using these sentences, high-level descriptive process models are generated.

The process information of Contractor Safety Procedure guidelines is mainly composed of Type-3 sentences. The time comparison results of the manufacturing processes reveal that the process model generation was more effective when the process model was complex. The required time is decreased from 38 to 10 person-minutes for the approval process, which is the most complex process model, from 23 to 14 person-minutes for the orientation process and from 8 to 6 person-minutes for the training process.

Approval Process; all the functions are the same, but the generated model have one additional function which is not stated in the existing model. All the existing models have more documents; however, the generated model has more business rules which also refer to documents used in the functions.

Orientation Process; although both models cover the same information, the generated model is more descriptive and the existing model is more general. The generated model has more business rules.

Training Process; the generated model has one more function and hence has information that the existing model does not have. The other two common functions are the same.

Registration Process; (includes Type-3 sentences) since the notation of both of the process model notation is different, the information coverage of the models are compared by comparing activities with functions. The existing model has more detail whereas the generated model is high-level descriptive. The difference between the two models is that the existing model gives more detailed information in relation to the document collection for registration. Nevertheless, all our functions are covered in their activities.

The processes for Visa Decision, Bank License and Annual Leave are also modeled with ProModGen which are given in Appendix G. The details of the process models are given in Table 24. Each process is allocated two rows; the first row presents the total number of process elements and the second presents the inserted elements. The time spent on finalizing the processes consists of individual selection, interpretation of the model regarding the decision points, editing the function names in which the passive verbs are used, and the insertion of necessary process elements. The Bank License regulation consisted of Type-1 and Type-2 sentences. For some parts of the sentences the process model contained low-level details and for other parts, it required effort in the interpretation of decision points. The Annual Leave process consisted of Type-3 sentences; hence, it was a high-level descriptive model. The Visa Decision process consisted of Type-1 and Type-3 sentences; thus, it required minimum insertions and was a high-level descriptive model.

Table 24 Process Details for Visa Decision, Bank License and Annual Leave

	Time	**Function**	**Event**	**Role**	**Business Rule**	**Connector**	**Document**	**Location**
Visa Decision	14 p-m	7	2	8	7	2	1	1
Insertion			2			2	1	
Bank License	20 p-m	17	3	15	11	3	1	1
Insertion		2	3			3	1	
Annual Leave	15 p-m	6	4	5	14	2	1	
Insertion		1	4			2		

Regarding the answer for Research Question 6, ProModGen is efficient in terms of time and effort according to manual development. It created process models with a high-level description which contains the same information as the manually developed process models.

4.3 **Experiment**

This study analyzed the development of process ontology and process models using the proposed and traditional methods to evaluate their efficiency from the perspective of the analyst. In the scope of this study, the analysts were PhD students, who developed process ontology and created process models from organizational guidelines.

The following research questions are addressed:

Research Question 7: Does process ontology development with PrOnPo require less time and effort than manual development?

Validation method for Research Question 7: Two groups of participants developed process ontology from the given guideline; one group manually (control) and the other using PrOnPo. The time and effort expended by the two groups are compared. Additionally, the process ontologies developed by the control group are analyzed to establish a baseline and the process ontologies developed with PrOnPo are compared with this baseline based on precision and recall metrics.

Research Question 8: Do the process models generated by ProModGen have the same coverage as the manually created process models?

Validation method for Research Question 8: The process models created by the control groups are analyzed to create a baseline. Then, the generated process models are compared to the baseline process model in terms of the number of process elements.

Research Question 9: Do the process models generated by ProModGen require less time compared to the manually created process models?

Validation method for Research Question 9: The time and effort required for process modeling is compared between the treatments.

4.3.1 *Mitigation of Threats to Validity in Experiment*

Four types of validity (conclusion, internal, construct, and external) were considered for the experiment (Wohlin et al. 2012). One potential threat to the general validity of the exploratory and multiple case studies may be their implementation by the same person. In order to minimize this threat, the experiment was conducted with multiple participants.

Conclusion validity concerns how the experiment reflects a real situation. One general threat to the conclusion validity of this experiment was the low number of samples and participants. In order to minimize this threat, the experiment was undertaken using a guidelines document from a real life context with participants who had previously taken a course on process modeling and participated in ontology development projects. The PrOnPo methodology and tool was implemented with a multiple case study to test the hypotheses. Since ontology development using the PrOnPo tool requires less knowledge of ontology development, in this stage, the participants that had less familiarity with ontology were grouped together as the Auto group. This was to prevent bias in results since grouping these participants in the manual group could increase the average time for process ontology development and may not represent a real life context.

Both groups are asked to develop process ontology and then to create process models (automated and manually) to learn about the ontology development process. However, the traditional method for process modeling is to use organizational sources (guidelines, event logs, and interviews). In order to avoid any bias when comparing the manual and automated development of process models, a third group is formed to create process models using the organizational guidelines.

Internal validity is related to the selection of the participants and instrumentation. In this study, the participants are selected based on their experience in process modeling and ontology development to reduce threat to internal validity due to an inexperienced participant taking much longer than the average time. Another important consideration is related to the duration of the experiments. Increased duration of experiments would reduce the motivation and concentration of participants, potentially resulting in bias. In order to prevent such biases, only one set of guidelines containing information on a single-process model is used and the conceptualization phase is excluded for the manual development group. The PrOnPo tool and UPROM modeling environment are introduced to the participants prior to the experiment. Furthermore, selecting a domain about which all participant have extensive knowledge would create bias in the results as a result of paying less attention to the interpretation of the guidelines; on the other hand, a completely unknown domain would also create bias since they might have trouble to figure out the process. To avoid these biases, in this study, the guidelines are selected from the banking domain with which the participants are only familiar. Furthermore, to prevent any bias regarding the evaluation of the developed ontology, the scope of the ontology is limited to the competency questions and information written in the guidelines.

Construct validity is related to the concerns of the concept behind the experiment. For this study, the participants could present a threat if they knew the concept behind the experiment. To minimize this threat, the hypothesis was not discussed with the participants. Since external validity is related to the concerns of conducting the experiment in different environments and obtaining the same results, participants having experience in industry were selected. Regarding generalizability, this study was performed in the context of a multiple case study.

4.3.2 *Case Selection Criteria*

The following criteria were defined for the case selection: guidelines had to be a procedure, regulation, policy or directive from the banking domain. The number of pages of the guideline must not exceed 10 pages. The selected guidelines must be publicly available and must include information for creating process model. The selected case must reflect a real-life context and the problem complexity should at least be moderate-level good. The selected case must be ecologically valid. The language of the selected cases must be in English, preferably written by a native speaker. According to the defined criteria, the Customer Identification Procedure was selected as shown in Table 25.

Table 25 Case Summary for Experiment

Case Characteristics	**Experiment**
Name	INGDirect Procedures for Brokers[15]
Type	Procedure
Focus Domain	Finance
Number of Pages	7
Number of Process Model	1
Real Life Context	Good
Problem Complexity	Good

4.3.3 *Selection of Participants*

The following selection criteria are defined for selecting the participants;

- Having successfully completed a course involving process modeling
- Having participated in an ontology development project.

Nine PhD students from the Informatics institute that met the selection criteria and had experience in the area were included in the study. Three groups each consisting of three participants were created; the control (manual) group, who was asked to manually developed process ontology and process models, the experiment (auto) group, who used the PrOnPo methodology and tool for this task, and the traditional group, who did not develop process ontology but only used the guideline to create process models. People with more experience in ontology development were included in the manual group. Since the PrOnPo tool did the majority of the work for process ontology development, people with less experience in ontology was included in the auto group. People who only had experience in process modeling was included in the traditional group.

4.3.4 *Background of the Participants*

All the selected participants were students at the Information Systems Department of the Informatics Institute of Middle East Technical University, Ankara, Turkey. They had all taken the IS526 Software Quality Management course, in which they learned about process modeling. In addition to attending this course, all the participants had process modeling experience in the related industry. The participants in the manual group had also taken part in the ontology development phase of a research and development project for Constructing Knowledge Map of the Ministry of Development. Therefore, they already knew how to acquire process information from guidelines. The participants in the auto and traditional groups were involved in process modeling in the industry; thus, being able to interpret the extracted process information, the members of these two groups were able to evaluate the output of PrOnPo and create process modeling from the given guideline, respectively.

[15] https://introducer.ingdirect.com.au/assets/pdf/IM382_ID_Procedure_Brokers.pdf

4.3.5 *Instrumentation*

As given in Case Selection, the INGDirect Customer Identification Procedures for Brokers from the finance domain was selected according to the given criteria. The auto group was given a soft copy of the regulation, the PrOnPo tool and the ProModGen plugin. The manual group was provided with a soft copy of the regulation as well as a PC running MS Excel and an UPROM modeling environment. The traditional group used the soft copy of the regulation and the UPROM modeling environment.

4.3.6 *Treatments*

Within the scope of the experiment, two treatments for process ontology development were defined: The first was manual process ontology development (MPOD) and the second one was automated process ontology development using the PrOnPo tool. For the creation of the business process model using the guideline, three treatments were determined as manual process modeling (MPM), ProModGen, and traditional process modeling (TPM).

4.3.7 *Variables and Data Collection*

Dependent variables change according to independent variables. Thus, in this experiment, for the process ontology development with MPOD or PrOnPo treatments, time and effort were dependent variables whereas for process modeling using MPM, ProModGen or TMP treatments (process modeling), time, effort, and coverage were the defined dependent variables.

Table 26 Summary of Data Collection

Process Ontology Development	Manual			Auto			Traditional		
Participants	S1	S2	S3	S4	S5	S6	S7	S8	S9
Specification	Scope of the process ontology is the Procedures for the Brokers						--		
Acquisition	a1	a2	a3	-	-	-	-	-	-
Evaluation	e1	e2	e3	e4	e5	e6	-	-	-
Conceptualization	-	-	-	-	-	-	-	-	-
Overall Time for Process Ontology	t1	t2	t3	t4	t5	t6	-	-	-
Overall Time for Process Modeling	m1	m2	m3	m4	m5	m6	m7	m8	m9
Total Number of Process Elements	p1	p2	p3	p4	p5	p6	p7	p8	p9

The summary of the data collection regarding the variables is given in Table 26. Acquisition of process information was automated for the auto group; thus, the data on the effort variable was only collected from the manual group. The data on evaluation effort was collected from both manual and auto groups. Conceptualization was automated for the auto group but not for the manual group in order to shorten the experiment time. The total effort required for process modeling and the number of process elements were determined for all groups. The data regarding time and effort was collected in person-minutes.

4.3.8 *Hypothesis Formulation*

The proposed method was expected to be more efficient than the traditional method. To test this expectation, the following three assumptions were formulated:

1. The members of the auto group, who use PrOnPo, will develop process ontology in a shorter time than those of the control group, who develop process ontology manually.
2. The members of the auto group, who use ProModGen will create the process model in a shorter time than those of the control group, who create process models manually.
3. The process models of the auto, control and traditional groups will have the same coverage.

According to the defined assumptions, the following formal hypotheses were defined:

1. Null Hypothesis, H_0: The time and effort required for developing process ontology manually is less than that required using the semi-automated PrOnPo tool.

Alternative Hypothesis, H1: Time (semi-automated PrOnPo) < Time (manual process ontology development).
Measures needed: the time and effort in person-minutes for process ontology development.

2. Null Hypothesis, H_0: The coverage of the process models created using ProModGen is less than that of the manually created process models.

Alternative Hypothesis, H1: Coverage (ProModGen) > Coverage (manual process modeling)
Measures needed: the number of process elements and activity checklist.

3. Null Hypothesis, H_0: The time required to create the process model manually is less than required using ProModGen.

Alternative Hypothesis, H1: Time (ProModGen) < Time (manual process modeling).
Measures needed: the time and effort in person-minutes for process model creation.

4.3.9 *Validation Strategy*

Validation of the results was performed according to the data collection information in Table 26 in three steps;

1. Validation of the efficiency of Process Ontology Development;

$a1 + e1 = t1$ and $a2 + e2 = t2$ and $a3 + e3 = t3$ and
$(t1 + t2 + t3) / 3 > ((t4 = e4) + (t5 = e5) + (t6 = e6)) / 3$

2. Validation of the efficiency of Process Modeling;

$(m4 + m5 + m6) / 3 < (m1 + m2 + m3) / 3 < (m7 + m8 + m9) / 3$

3. Using and Activity Checklist in Appendix H to evaluate coverage.

4.3.10 *Operation*

Before starting the experiment, the participants were briefly introduced to process ontology development. Although they had experience in developing domain ontologies, they were asked to find the process elements (functions, events, information carriers, roles, applications, and technical terms) in the given regulation. An example for process ontology development was provided from a previous multiple case study.

The UPROM modeling environment was introduced to all the groups by describing how to create and edit process elements. The participants were asked to perform a sample modeling in order to get familiarized with the environment. The time and effort required for the phases of process ontology development and process model generation were recorded separately.

4.3.10.1 *Activities of the Auto Group*

The auto group used PrOnPo for process ontology development and ProModGen for process model generation. The following activities were planned for this group;

1. Upload the regulation to PrOnPo
2. Analyze the output
3. Replace the ontology individuals if needed
4. Upload the developed process ontology to ProModGen
5. Select the related instances
6. Generate the process model
7. Edit and finalize the process model

4.3.10.2 *Activities of the Manual Group*

The manual group was involved in MPOD using a soft copy of the regulation, an Excel sheet and the UPROM modeling environment. The following activities were planned for this group;

1. Interpret the regulation
2. Find the process elements in the document
3. Record the process elements on the Excel sheet
4. Record the relations between the process elements on the Excel sheet
5. Create the process elements on UPROM
6. Create relations on UPROM
7. Finalize the process model

4.3.10.3 *Activities of the Traditional Group*

This group only used a soft copy of the regulation and the UPROM modeling environment to create process models. The following activities were planned for the traditional group;

1. Interpret the regulation
2. Create process models on UPROM

4.3.11 *Results and Analysis Regarding Research Question 7*

Does process ontology development with PrOnPo require less time and effort than manual development?

1. Null Hypothesis, H_0: The time and effort required for developing process ontology manually is less than that required using the semi-automated PrOnPo.

Process ontology development was divided into four phases; specification, acquisition, evaluation, and conceptualization. Specification was defined prior to the study. In order to answer the research question regarding the time and effort required for process ontology development, the results of these variables in person-minutes for three phases (acquisition, evaluation, and conceptualization) were compared between the manual and auto groups (the traditional group was excluded since the members of this group did not develop process ontology). As shown in Table 27, since the acquisition phase was automated for the auto group, it took approximately 1 minute for the tool to perform the task. For the manual group, the acquisition phase took 71 to 101 person-minutes. Since the information was extracted by the participants, it was more accurate according to the PrOnPo's output; therefore, the time for evaluation for the manual group was around 20 person-minutes whereas it was 51 to 65 for the auto group. The values of t1, t2 and t3 were assigned for creating classes and individuals from the extracted information. These values were assumed to be greater than 0. In any circumstances, the overall time for process ontology development was higher in manual development compared to PrOnPo. Furthermore, using this semi-automated tool, the total effort decreased from 108 person-minutes (114 + 121 + 90 / 3) to 60 person-minutes (61 + 52 + 66 / 3). The results of the comparison showed a significant difference between the manual and auto groups; thus, null hypothesis 1 was rejected.

Table 27 Summary of Results on the Required Time for Process Ontology Development

Process Ontology Development	Manual				PrOnPo Semi-Automated		
Participants	S1	S2	S3		S4	S5	S6
Specification	Scope of the process ontology is the Procedures for the Brokers						
Acquisition	94 p-m	101 p-m	71 p-m		1 min	1 min	1 min
Evaluation	20 p-m	20 p-m	19 p-m		60 p-m	51 p-m	65 p-m
Conceptualization	t1	t2	t3		0 min	0 min	0 min
Overall Time	114+t1 p-m	121+t2 p-m	90+t3 p-m		61 p-m	52 p-m	66 p-m

PrOnPo extracted 132 triples consisting of 401 individuals. The summary of the extracted process ontology is given in Table 28. This table shows that human acquisition can cause missing individuals; however, it does not mean that this will always be the case. In manual development 1, the number of extracted individuals was closer to the number of actual individuals. Table 29 gives the output summary of PrOnPo. Based on these numbers, the precision and recall metrics were calculated and the results are presented in Table 30. The replaced individuals were considered as irrelevant, thus not retrieved. Hence, this calculation resulted in 74% precision and 71% recall for the fully automated PrOnPo.

Table 28 Output Summary of Developed Process Ontologies

Number of	Manual 1	Manual 2	Manual 3	PrOnPo
Triples	119	111	106	132
Individuals	429	384	362	401
Missing Process	0	2	0	3
Missing Business rule	1	9	6	6
Actual Individuals	417	417	417	417

Table 29 Output Summary of Semi-Automated PrOnPo

Number of	PrOnPo
Irrelevant Sentences	9
Replaced Individuals	100
Inserted Individuals	20
Deleted Individuals	4

Table 30 The Results on Precision and Recall for Fully Automated PrOnPo

PrOnPo- Full	Relevant (417)	Irrelevant			
Retrieved (401)	297	4+100	Precision	297/401	74%
Not Retrieved	20+100		Recall	297/417	71%

4.3.12 *Results and Analysis Regarding Research Question 8 and 9*

Do the process models generated by ProModGen have the same coverage with the manually created process models?

2. Null Hypothesis, H_0: The coverage of the process models created using ProModGen is less than that of the manually created process models.

Do the process models generated by ProModGen require less time and effort according to the manually created process models?

3. Null Hypothesis, H_0: The time and effort required to create process model manually is less than that required using ProModGen.

The process models created by Traditional, Manual and Auto groups are given in Appendix I, Appendix J and Appendix K respectively. The output summary of the total number of process elements created and the time and effort (in person-minutes) required for each group member are given in Table 31. The Activity Checklist given in Appendix H showed that the process models generated by all groups covered the same correct information. Additionally, the generated process models were consistent with the guidelines. The names of the process elements of the manual and traditional groups were mostly interpreted from the sentences extracted from the guideline. According to the results, there was no significant difference between the groups regarding the number of process elements. Thus, null hypothesis number 2 was rejected.

Regarding time and effort, the traditional group spent the most time and effort for modeling whereas the manual group took moderate time for modeling since they had already learned the process when developing process ontology. The auto group spent the least time and effort for modeling. According to the results, process model generation was more efficient for the auto group compared to the manual and traditional groups (52 + 42 + 40 / 3 > 18 + 19 + 22 / 3); therefore, null hypothesis number 3 was rejected.

Table 31 Output Summary Regarding Process Model Creation

Totals Number of / **Process Name**	Functions	Events	Documents	Connectors	Roles	Business Rules	total	Required time
Traditional 1	23	11	3	12	7	0	56	62 p-m
Traditional 2	17	12	13	10	8	0	60	70 p-m
Traditional 3	21	15	6	6	4	2	54	57 p-m
Auto 1	19	10	3	6	21	17	76	18 p-m
Auto 2	18	8	3	6	20	15	70	19 p-m
Auto 3	24	6	1	7	23	23	84	22 p-m
Manual 1	23	12	6	14	9	6	70	52 p-m
Manual 2	17	10	4	7	3	0	41	42 p-m
Manual 3	22	12	2	9	3	6	54	40 p-m

4.3.13 *Participant's Reviews*

The participants in the auto and manual groups were asked the following questions in order to evaluate the approach and the tool.

1. Would you prefer to extract process information from scratch or to evaluate the extracted and structured process information?

The members of both groups agreed that extracting process information from scratch would take longer time and require more effort. Therefore, they stated that they would prefer to evaluate the extracted and structured process information.

2. Would you prefer to create process models from scratch or finalize an initial version of the process model?

The auto group suggested that it was more efficient to work on created activities rather than creating a process model from scratch. The manual group stated that it would depend on the quality of the initial version. If the quality of the model was not sufficient, they would prefer modeling from scratch.

3. How did you find the approach and the tool overall? How would you define its strengths and weaknesses? Did you find it useful?

The auto group stated that they found the approach very useful since reading long documents and identifying activities would take significant time and effort. Regarding strength, in addition to the efficiency provided by the tool, the auto group also stated that the outputs offered a good validation method for evaluating the processes. Regarding the weaknesses, the members of this group suggested that transforming process entities to process model elements was very useful; however, the underlying technology, UPROM, had some problems during the editing process related to the response time and capabilities. These participants considered that if the process model generation was built on a different process modeling environment, the efficiency would be better.

After the experiment, the manual group was introduced to the PrOnPo tool and asked for their opinions concerning the potential strengths and weaknesses of the proposed approach. The manual group commonly agreed that this approach would facilitate acquisition of information; thus, it would reduce a large part of the workforce required for process modeling as well as increase the quality. In process modeling, it may be necessary to use unstructured documents as the sole or main source. Therefore, the manual group considered that this approach would also reduce the cognitive effort for structuring the process information or modeling the process. Regarding the weaknesses, the focus of the manual group was on process modeling. Some documents may contain information about many processes. In this case, draft process models to be produced from these documents would be very large and need to be allocated to different processes, which could greatly increase the workforce and lead to errors. Thus, it may be necessary to divide the process according to the sub-processes covered by large documents.

4.4 Discussion and Summary of the Case Studies and Experiment

Throughout this chapter, we explored, generalized and validated the PrOnPo methodology and tool.

The research started with an Exploratory Study 1, in which the requirements for automating process ontology development are identified as given in section 4.1.4. As result of the first exploratory study, the PrOnPo prototype is implemented. The first case study for the PrOnPo prototype is conducted to analyze the underlying technologies for different languages (Turkish and English) (Gurbuz & Demirors 2017a). The results of the study showed that the PrOnPo tool is more efficient for English language than Turkish language. Regarding to this application on English and Turkish guidelines the PrOnPo tool is implement for English language as given in Chapter 3.

Secondly, Exploratory Study 2 is conducted to explore the PrOnPo tool and to identify improvement potentials regarding the tool which is given in section 4.1.5 (Gurbuz & Demirors 2017b). It is observed that the quality of the guideline plays an important role for the success of the approach. We identified the quality aspects of the guideline as; the language usage; the proportion of the number of irrelevant sentences to activities, business rules and terms and lastly the number of long and complex sentences. The chosen guideline was written by a non-native author, contained small amount of irrelevant and complex sentences. Nevertheless, the precision and recall metrics are satisfying. Moreover, the chosen keywords also effected the results in terms of recall metric which decreased from 87% to 79% in contrast to precision metric which increased from 94% to 95%. F-Measure (measured for comparing the keyword usage) for both methods are 0.86 and 0.90 respectively. Hence, we recorded no big difference between using keywords and not using keywords.

Thirdly, Multiple Case Study is conducted to generalize the tool on different types of guidelines from different domains. We analyzed the PrOnPo using it as full automated and semi-automated. The full-automated use of PrOnPo differs from its semi-automated use in which the evaluation phase of the process ontology development is skipped and the extracted knowledge (partitioned objects) is directly conceptualized. The PrOnPo tool is implemented on a rule-based methodology and some sentences do not obey the rules and rules may result in misplaced individuals and corrupted sentences. Therefore, the precision and recall metrics are higher for the semi-automated use of PrOnPo. To obtain better performance of the tool, user involvement is currently required; however, a self-learning approach for partitioning objects is on our future agenda.

Regarding the precision and recall metrics for different domains and guideline types, we calculated the F-measure (the harmonic mean of precision and recall metrics) to make a comparison as given in Table 32.

F-Measure= 2*Precision*Recall / (Precision +Recall)

Table 32 F-Measure Summary

Case	Domain	F-Measure
Specialized Processes	Manufacturing	0.79
Common Process	Human Resources	0.72
University Process Website	University	0.80
Government Regulations	Bank Regulation	0.68
Government Regulations	Visa Policy	0.71

Regulation and policies which belong to the government case consist of more complex structured sentences. We observed that process ontology development for Bank Regulation and Visa Policy have the lowest F-Measure compared to the other categories. The University Registration Process has the highest F-measure since it contains the least complex structure. In other words, the website consists of to-the-point process sentences. Likewise, the private companies' specialized procedure also consists of less complex sentences which exactly define the processes. Hence, the F-measure for specialized processes is high. Consequently, the performance of the PrOnPo tool regarding the extraction of the relevant and correct process information is mostly dependent on the structure and complexity of the sentences.

As part of Multiple Case Study, process model generation (ProModGen) is practiced on the selected cases. ProModGen creates process elements in their execution order and hence reduces the time and effort for interpretation phase for process modeling. Modeler should select the related individuals and finalize the process model by inserting necessary decision points and events, edit function names when passive verbs are used or extend the scope of the process model by adding process elements. Therefore, process model generation requires user involvement. An improvement for full automated process model generation is in our future work agenda.

Lastly, an Experiment is performed to validate the efficiency of PrOnPo with 3 groups of 9 people. Manual group manually developed process ontology and modeled the process whereas the Auto group used PrOnPo for process ontology development and generated process model from process ontology. Traditional group is formed to manually model the process using the guideline. Hence for comparing the efficiency of PrOnPo there were two groups whereas for efficiency of process model generation there were three groups. According to the results the following issues are noted:

- Process ontology development with PrOnPo is more efficient than manual development
- Manual process modeling after developing process ontology is more efficient than traditional process modeling using guideline

- Process model generation is more efficient than manual process modeling.
- The generated process models covers the same information as the manually created processes.

The summary of all the case studies and experiment is given in Table 33. Note that the precision and recall metrics for exploratory studies are based on the correct triples whereas in the other cases they are based on the correct individuals created by fully and semi-automated PrOnPo. Additionally this table also includes the normalization of the studies regarding the complexity. 30 sentences are selected by random from the guidelines and each of them were analyzed according to the number of extracted, irrelevant, incorrect, replaced and correct instances. According to these results the precision is calculated in order to normalize and compare the studies. The highest precision is 89% whereas lowest is 67%. The average precision based on the random selection for fully automated PrOnPo is 80%.

Table 33 Summary of Applications and Normalization

	Exploratory		Multiple Case					Experiment
Guideline	Directive for Exchange Programs	Regulation for Graduate Studies	Contractor Safety Procedure	Visa Policy	Registration Process	Leave (HR) Policy	Regulation for Bank License	Procedures for Brokers
Number of Sentences	159	440	153	37	41	29	36	121
Number of Triples	130	421	133	33	45	30	38	132
Precision (Full)	-	-	83%	83%	82%	76%	76%	74%
Precision (Semi)	88%	94%	94%	100%	100%	97%	100%	99%
Recall (Full)	-	-	76%	62%	79%	69%	62%	71%
Recall (Semi)	77%	87%	87%	75%	96%	88%	81%	95%
Random 30 Sentences								
Number of Instances	97	102	104	95	113	114	106	102
Irrelevant Instances	x	1	8	x		3	x	1
Incorrect Instances	3	4	x	5	3	x	x	x
Replaced Instances	10	12	12	15	21	24	12	34
Correct Instances	84	85	84	75	89	87	94	68
Precision (Full)	84/97	85/102	84/104	75/95	89/113	87/114	94/106	68/102
	87%	83%	81%	79%	78%	76%	89%	67%

Consequently, the parser creates the output in a Tree structure. We defined the rules accordingly to extract the leaves of the Tree. We avoided any rules which will work for one case but will destroy the other cases. Therefore the defined rules for partitioning the object covers until the 4th depth level of the Tree. For some sentences, partitions were in deeper level, hence the rules do not work. And some other reasons for corrupted sentences and misplaced individuals are;

➢ Parser cannot parse long and complex structured sentences

- Parser cannot parse when there is no subject-verb-object structure in the sentences
- In some cases, e.g. when the sentences are very long, parser cannot divide the sentences into two when there is a conjunction.

Regarding process model generation, some advantages are noted for the use of ProModGen. The first is that even though sentences are complex, as long as they contain process information, the process element is not affected. The Function element contains the verb and object of the sentences and the subject is extracted as Role. In traditional modeling, the modeler tends to insert Role to one Function element and it is assumed that the following functions are also executed by the same Role. However, this might result in confusion for people who are not familiar with the processes. ProModGen, on the other hand, attaches the Role element to each Function. Similar to Roles, Business Rules are another important aspect of process models which should be considered when executing functions. ProModGen allows the modeler to define business rules from individuals and create how, why and when questions of the sentences as business rules. However, if the process is hidden in the sentences which do not directly present the subject and the function, the modeler needs to interpret and create the process element accordingly by inserting events and decision points.

In contrast to manual process model creation, the generated process models' elements and their relations with other elements are represented in a formal semantic language. These formal specifications eases search and query of the processes. Moreover, the developed process ontologies cover more comprehensive process information than a business process model needs. For instance; for the completion of a graduate program process, students are given diplomas at the end if they are regarded successful. The developed process ontology contains this information as well as contains the information of who signs the diplomas. Even though this information do not need to be presented on the model, it can be related to the relevant process element in the background and can be seen when the analyst queries the term "diploma". Additionally, process ontologies cover additional concepts such as jargons, terms, definition of entities and business rules which belong to the organization and which are not represented in the process models but is kept in the process ontology.

The developed process ontology can be used not only for process model generation but also for querying and discovering process information as well as relating to the domain ontologies or data analytics (Milosevic et al. 2016) of the corresponding domain. Although, there are studies that address the ontology integration for reuse (Noy 2004; Shvaiko & Euzenat 2013) in a general approach, most known ontology development methodologies do not focus on reusing and guidance is needed for re-using ontologies for specific approaches (Shah et al. 2014). One of our future works concerns extending our methodology to guide users in reusing the developed process ontology when constructing domain ontologies and performing data analytics.

CHAPTER 5

CONCLUSION

This thesis presented the PrOnPo methodology and tool to support process ontology development and process model generation. As part of this thesis, two exploratory studies, a multiple case study and an experiment were performed to explore, generalize and validate the proposed approach, respectively.

This chapter presents the summary of the thesis work and results, contributions and limitations of the proposed methodology and the directions for future work.

5.1 Summary of the Thesis Study

The research undertaken for this thesis started with an exploratory study to investigate how to discover and develop process ontology from organizational guidelines written in natural language. In this stage of the study, the PrOnPo methodology was proposed, the requirements for automated process ontology development were identified, and the PrOnPo tool was implemented. The proposed tool automatically extracts process activities, roles, documents, business rules and other process elements, constructs the relations between them, and develops process ontology from organizational guidelines.

The proposed methodology for process ontology development consists of four phases: specification (defining scope), acquisition (extracting information), evaluation, and conceptualization (transforming to process ontology). The PrOnPo tool automates the specification and acquisition phases. This tool can be used with user involvement (semi-automated) or without user involvement (fully automated). In the semi-automated PrOnPo, before the conceptualization phase, the user evaluates the extracted information, which constitutes the evaluation phase. In the fully automated version, the evaluation phase is skipped and the tool directly conceptualizes the extracted information (the acquisition phase).

As part of the PrOnPo methodology, the ProModGen plugin was implemented to transform the process ontology instances to the process model elements. The PrOnPo tool creates ordered and connected process entities, which are then transformed by ProModGen to process elements to create process models. This transformation not only reduces the time and effort required for process modeling but also improve semantic quality of process models and their consistency with process ontology.

Within the scope of exploratory, multiple case and experiment studies, the PrOnPo methodology and tool were examined in terms of four dimensions; coverage (precision & recall), impact of language (Turkish and English), generalizability (adaptation to different domains), and efficiency (time and effort). Additionally, as part of these studies, ProModGen was analyzed in terms of its efficiency and coverage.

The first exploratory study was conducted using Turkish and English guidelines in order to analyze the effect of underlying technologies used and the impact of language. Using the English version of the guideline, PrOnPo was found to be more effective.

In the second exploratory study, a keyword option was included in PrOnPo to identify keywords using a frequency analysis. The reason for using this option was to eliminate the irrelevant information and increase the precision. According to the results, the precision increased as expected but recall decreased. Therefore, the F-measure (harmonic mean of precision and recall) was calculated to compare the use of keywords. The F-measure was determined to be 0.86 when keywords were used and 0.90 when these words were excluded from analysis. This indicated that using keywords did not have a significant effect on the results.

In the second stage of the study, the PrOnPo tool was tested in terms of its adaptability to different types of guidelines from different domains in a multiple case study. Five guidelines (2 policies and 1 regulation, website and procedure each) belonging to the government and private companies were selected. The precision and recall metrics for both fully automated and semi-automated process development were analyzed through a multiple case study. The average for extracting the relevant process individuals from all the extracted individuals was 98% in the semi-automated development and 80% in the fully automated development. The output process ontology of semi-automation covered 85% of the overall process information whereas the output of full-automation covered 70%. Based on these results, the PrOnPo tool is considered to be promising in terms of its applicability and adaptability to different domains.

A further evaluation of the PrOnPo tool was performed with participants in an experiment. Nine PhD students with experience in ontology development and process modeling volunteered to take part in this experiment. Three groups consisting of three participants were formed as follows: the auto group that was involved in the evaluation phase of process ontology development using the PrOnPo tool, the manual group that was involved in the acquisition phase of manual process ontology development, and the traditional group that only modeled the given process manually. A tenth participant assumed the evaluator role and performed the evaluation of the process ontology developed by the manual group. During the experiment, the precision and recall were calculated as 74% and 71% for the fully automated PrOnPo.

The overall coverage and efficiency of the full-automated PrOnPo tool was calculated. Regarding the coverage of the developed process ontology, the average precision and recall metrics of six different guidelines (5 from the multiple case study + 1 from the experiment) was determined. PrOnPo successfully extracted 79% of the relevant information from the total extracted information and covered 70% of the overall relevant information, fully-automated. However, when the user involves in the evaluation, PrOnPo could semi-automatically cover 87% of the relevant information. Additionally, 30 random sentences are selected from each case (8 cases in total) and the precision metric for fully automated PrOnPo was calculated as 80% in average.

Regarding the efficiency of PrOnPo, the effort in person-minutes required for manual and semi-automated (PrOnPo) development of process ontology was examined. According to the results of the second exploratory study, manual process ontology development required 200 person-minutes whereas using PrOnPo reduced this value to 50 person-minutes. In the multiple case study, it was observed that the required time and effort decreased from 250 to 57 person-minutes. In the experiment, the average time required for process ontology development decreased from 105 person-minutes (the manual group) to 60 person-minutes (the auto group). Consequently, the proposed methodology decreased the total required time and effort for process ontology development from 555 person-minutes to 167 person-minutes. In other words, the PrOnPo tool decreased the total time and effort required for the process ontology development of 7 guidelines by 70%. Consequently, using PrOnPo semi-automated is more effective according to full-automated and still more efficient than manual process ontology development.

Regarding the efficiency of ProModGen, the average required time and effort required for this process were calculated. In the Experiment, the average required time and effort for process modeling decreased from 45 to 20 person-minutes for process modeling. According to the results of the multiple case study, the total effort decreased from 114 to 50 person-minutes. In other words, ProModGen decreased the required time and effort for process modeling by 56%.

Regarding the coverage of the process models, the output process models produced by the three groups in the experiment were examined. The number of process elements ranged from 54 to 60 in the traditional group, 54 to 70 in the manual group and 70 to 84 in the auto group. According to the results of the activity checklist and the number of process elements of each group, there was no significant difference between the traditional models and those generated using the proposed methodology. ProModGen covered all the relevant activity information in the selected guideline.

5.2 **Contributions**

The major contributions of this thesis work are listed below;

- Proposing the PrOnPo methodology for developing process ontologies from organizational guidelines,
- PrOnPo being the first tool to support the methodology for automated process ontology development,
- Presentation of ProModGen plugin for transforming process ontology instances to ordered and connected process model elements.

The minor contributions achieved as part of this research are as follows;

- Implementing an algorithm that uses a natural language parser to extract process elements from sentences,
- Analyzing and comparing the impacts of language and natural language parser on process ontology development (the latter was found to be have a higher effect),
- Addressing the ambiguity and inconsistencies in natural language through the development of process ontologies from the retrieved process information,
- Analysis of the differences in the results of frequency analysis for extracting sentences with and without keywords (found to be insignificant),
- Examining the efficiency of the PrOnPo tool in exploratory, multiple case and experimental studies based on the time and effort required for process ontology development (reduced by 70% using the proposed tool),
- Analyzing the effectiveness of the PrOnPo tool in multiple case and experimental studies (the tool successfully and accurately extracted 79% of the relevant process information),
- Presenting the transformation from process ontology instances to process model elements,

- Demonstrating the efficiency of the ProModGen plugin in multiple case and experimental studies (56% reduction in the time and effort required for process modeling).

The results are significant since this is the first study in the literature to propose an efficient methodology and tool for developing process ontology from organizational guidelines and generating process models in EPC notation.

The proposed tool can be used by people from different organizations since it does not require any familiarity with ontology development. The developed process ontology facilitates process query and improves consistency by providing a shared understanding within the organization. Furthermore, the formal specification of the processes do not need to be limited to the information presented in the process models (which are also formally specified) but can also cover all the process information within the guideline.

5.3 Limitations and Future Work

The limitations of the study are given below;

- The identification of synonyms was not addressed as part of this thesis.
- Long and complex-structured sentences affect the success of the PrOnPo tool resulting in missing or corrupted triples.
- The PrOnPo tool is more efficient and effective when sentences explicitly present the process information.
- Process models created by ProModGen are dependent on the content of the sentences and can be either low-level detailed or high-level descriptive.
- The type of sentence effects process modeling using ProModGen. The process models generated from complex sentences require human interpretation for finalizing the model.
- The results regarding the validation of PrOnPo are based on 8 different applications conducted as part of this research.

Consequently, the necessity of user involvement remains to be a significant problem in the area of process ontology development and process model creation. Therefore, future work in this area should focus on extending the capabilities of the PrOnPo tool with machine learning to automate the evaluation phase and fully automating process modeling with ProModGen.

Based on the limitations of this research, other recommendations for future research are as follows:

- In this study, the frequency analysis for selecting keywords and eliminating the irrelevant sentences from the guideline was not successful. The selection of keywords by other techniques can be evaluated.
- The formal specification of the process information retrieved from the natural language text is a step in overcoming the problems related to ambiguity and inconsistency in natural language. However, there is a need to develop a method for the identification of synonyms in the developed process ontology.
- In this thesis, the use of process ontology for process modeling was presented. The proposed methodology should be extended to assist users in developing process ontologies in other domains such as the medical field for process discovery and the financial domain for data analytics.

- In this work, the Turkish and English versions of the most well-known natural language parser were used as part of the algorithm developed for extracting process information from the natural language text. Other natural language parsers in English such as should be explored (e.g., NLTK for English and ITU NLP for Turkish).
- It should be investigated whether process ontology developed using the proposed tool can simplify the interpretation of another guideline from the same domain.
- It should be analyzed whether the proposed method can develop software ontology from the Software Engineering Body of Knowledge.

REFERENCES

van der Aa, H., Leopold, H. & Reijers, H.A., 2016. Comparing textual descriptions to process models - The automatic detection of inconsistencies. *Information Systems*.

Van Der Aalst, W., Weijters, T. & Maruster, L., 2004. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9), pp.1128–1142.

van der Aalst, W.M.P. & Weijters., A.J.M.M., 2005. *Process Mining*, Available at: http://is.tm.tue.nl/staff/wvdaalst/publications/p272.pdf.

Abramowicz, W. et al., 2007. Semantically enhanced Business Process Modelling Notation. *CEUR Workshop Proceedings*, 251.

Agarwal, N. & Singh, L., 2014. Process Mining Tools: A comparative Analysis and Review. *Advances in Computer Science and Information Technology*, 1, pp.26–29. Available at: http://www.krishisanskriti.org/vol_image/02Jul201510075307 Neha Agarwal 26-29.pdf.

Akın, A.A. & Akın, M.D., 2007. Zemberek, An Open Source Nlp Framework for Turkic Languages. *Structure*. Available at: http://zemberek.googlecode.com/files/zemberek_makale.pdf.

Akman, B. & Demirörs, O., 2009. Applicability of process discovery algorithms for software organizations. In *Conference Proceedings of the EUROMICRO*. pp. 195–202.

Alkhaldi, N., Casteleyn, S. & Gailly, F., 2015. Enterprise-specific ontology-driven process modelling papers. *Lecture Notes in Business Information Processing*, 227, pp.472–488.

Aysolmaz, B. & Demirörs, O., 2015. Unified process modeling with UPROM tool. In *Lecture Notes in Business Information Processing*. pp. 250–266.

Belhajjame, K. & Brambilla, M., 2009. Ontology-based description and discovery of business processes. *Lecture Notes in Business Information Processing*, 29 LNBIP, pp.85–98.

Bi, B., Szulman, S. & Clement, A.J., 1999. TERMINAE : a method and a tool to build a domain ontology. *Knowledge Acquisition, Modeling and Management*, pp.49–66.

Blumberg, R. & Atre, S., 2003. The Problem with Unstructured Data. *DM Review*, 13, p.42. Available at: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=10186785&lang=pt-br&site=ehost-live.

Brewster, C. et al., 2004. Data driven ontology evaluation. *Fourth International Conference on Language Resources and Evaluation (LREC'04)*, pp.641–644. Available at: http://www.lrec-conf.org/lrec2004/article.php3?id_article=20#disambiguation.

De Cesare, S., Juric, D. & Lycett, M., 2014. Toward the automation of business process ontology generation. *Proceedings - 16th IEEE Conference on Business Informatics, CBI 2014*, 1, pp.70–77.

Cherfi, S.S. et al., 2012. Aligning Business Process Models and Domain Knowledge : A Meta-Modeling Approach To cite this version : Aligning Business Process Models and Domain Knowledge : A Meta-Modeling Approach. , pp.45–56.

Chiara Di Francescomarino, Chiara Ghidini, Marco Rospocher, L.S. and P.T. & FBK-irst, 2011. A framework for the collaborative specification of semantically annotated business processes. *Journal of Software Maintancen and Evolution*, 26(12), pp.1172–1192.

Coskuncay, A. et al., 2016. PROMPTUM Toolset: Tool Support for Integrated Ontologies and Process Models. *Dumas M., Fantinato M. (eds) Business Process Management Workshops. BPM 2016. Lecture Notes in Business Information Processing*, 281, pp.93–105. Available at: https://link.springer.com/chapter/10.1007/978-3-319-58457-7_7.

Dimitrov, M. et al., 2007. A BPMO Based Semantic Business Process Modelling Environment. *Business*, 251, pp.1–4. Available at: http://www.bibsonomy.org/bibtex/21ba7ebd8d131a554c530aeb5aacc2bc6/dblp.

Ding, Y. & Foo, S., 2002. development . Part 1 – a review of ontology generation. , 28(2), pp.123–136.

van Dongen, B. et al., 2005. The ProM framework: A new era in process mining tool support. *Application and Theory of Petri Nets 2005*, (3536), pp.444–454. Available at: http://wwwis.win.tue.nl/~wvdaalst/publications/p264.pdf.

Eisenbarth, T., 2013. *Semantic Process Models*. University of Augsburg.

Fan, S. et al., 2016. A process ontology based approach to easing semantic ambiguity in business process modeling. *Data & Knowledge Engineering*, 102, pp.57–77. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0169023X16000112.

Fernández-López, M., Gómez-Pérez, A. & Juristo, N., 1997. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. *AAAI-97 Spring Symposium Series*, SS-97-06, pp.33–40. Available at: http://oa.upm.es/5484/.

Francescomarino, C. Di et al., 2009. Semantically-Aided Business Process Modeling. In *International Semantic Web Conference (ISWC 2009)*. pp. 114–129.

Francescomarino, C. Di et al., 2014. The Role of Semantic Annotations in Business Process Modelling. *2014 IEEE 18th International Enterprise Distributed Object Computing Conference*, pp.181–189. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6972066.

Friedrich, F., Mendling, J. & Puhlmann, F., 2011. Process Model Generation from Natural Language Text. *Proc. of the 23rd International Converence, CAiSE 2011, Advanced Information Systems Engineering*, pp.482–496.

Gangemi, M.C.S.-F., Gomez-Perez, A. & Aldo, E.M., 2012. Ontology Engineering in a Networked World. , p.444. Available at: http://link.springer.com/10.1007/978-3-642-24794-1.

Garcia, A. et al., 2010. Developing Ontologies within Decentralised Settings. *Semantic E-Science*, pp.99–139. Available at: http://link.springer.com/chapter/10.1007/978-1-4419-5908-9_4\nhttp://link.springer.com/content/pdf/10.1007/978-1-4419-5908-9_4.pdf.

Ghidini, C., Rospocher, M. & Serafini, L., 2008. A formalisation of BPMN in description logics. *FBK-irst, Tech. Rep. TR*, pp.4–6.

Ghose, A.., Koliadis, G. & Cheung, A., 2013. Process discovery from model and text artefacts. *Journal of Chemical Information and Modeling*, 53(9), pp.1689–1699.

Goedertier, S. et al., 2011. Process discovery in event logs: An application in the telecom industry. *Applied Soft Computing Journal*, 11(2), pp.1697–1710. Available at: http://dx.doi.org/10.1016/j.asoc.2010.04.025.

Goedertier, S. et al., 2009. Robust Process Discovery with Artificial Negative Events. *Journal of Machine Learning Research*, 10, pp.1305--1340. Available at: http://jmlr.csail.mit.edu/papers/v10/goedertier09a.html\nhttp://www.jmlr.org/papers/volume10/goedertier09a/goedertier09a.pdf.

Gómez-pérez, A. & Manzano-macho, D., 2003. Deliverable 1 . 5 : A survey of ontology learning methods and techniques OntoWeb Consortium.

Goncalves, J.C., Santoro, F.M. & Baião, F.A., 2009. Business process mining from group stories. *Proceedings of the 2009 13th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2009*, pp.161–166.

Gruber, T.R., 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), pp.199–220. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.101.7493.

Gurbuz, O. & Demirors, O., 2017a. A Comparison of Process Ontology Discovery from Organizational Guidelines in Two Different Languages. In *43rd Conference on Software Engineering and Advanced Applications (SEAA)*. Accepted to be Published.

Gurbuz, O. & Demirors, O., 2017b. From Organizational Guidelines to Business Process Models: Exploratory Case for an Ontology Based Methodology. In *19th IEEE Conference on Business Informatics*. Accepted to be published.

Haller, A. et al., 2008. A Process Ontology for Business Intelligence. *Business*, (April), p.19. Available at: http://sib.deri.ie/fileadmin/documents/DERI-TR-2008-04-01.pdf.

Hazman, M., R. El-Beltagy, S. & Rafea, A., 2011. A Survey of Ontology Learning Approaches. *International Journal of Computer Applications*, 22(9), pp.36–43.

Hepp, M. & Roman, D., 2007. An Ontology Framework for Semantic Business Process Management. *Wirtschaftsinformatik 2007*, pp.1–18.

Höfferer, P., 2007. Achieving Business Process Model Interoperability Using Metamodels and Ontologies. *ECIS 2007 Proceedings*, (2007), pp.1620–1631. Available at: http://www.dke.at/fileadmin/DKEHP/publikationen/metamodell/Hoefferer_BP_interoperability_ontologies.pdf.

Jiang, X. & Tan, A., 2010. CRCTOL : A Semantic-Based Domain Ontology Learning System. , 61(1), pp.150–168.

Kalibatiene, D., 2015. Survey on Ontology Languages Survey on Ontology Languages. , (October), pp.124–141.

Koschmider, A. & Oberweis, A., 2005. Ontology based Business Process Description. *Proceedings of the CAiSE-05 Workshops*, (2), pp.321–333.

Laue, R., Storch, A. & Höß, F., 2015. The bflow* Hive - Adding Functionality to Eclipse-based Modelling Tools. In *Proceedings of the BPM Demo Session 2015 Co-located with the 13th International Conference on Business Process Management (BPM 2015)*. Innsbruck, Austria, pp. 120–124.

De Leoni, M., Van Der Aalst, W.M.P. & Dees, M., 2016. A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. *Information Systems*, 56(2016), pp.235–257. Available at: http://dx.doi.org/10.1016/j.is.2015.07.003.

Leopold, H. et al., 2015. Towards the Automated Annotation of Process Models. *Advanced Information Systems Engineering: 27th International Conference, CAiSE 2015*, pp.401–416.

Manning, C.D. et al., 2014. The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp.55–60. Available at: http://aclweb.org/anthology/P14-5010.

de Medeiros, A.K.A. et al., 2008. Process Mining Based on Clustering : A Quest for Precision. *Business Process Management Workshops*, 4928, pp.17–29.

Mendling, J., Leopold, H. & Pittke, F., 2014. 25 Challenges of Semantic Process Modeling. *International Journal of Information Systems and Software Engineering for Big Companies*, 1(1), pp.78–94. Available at: http://www.uajournals.com/ijisebc/journal/1/6.pdf.

Milosevic, Z. et al., 2016. An open architecture for event-based analytics. *International Journal of Data Science and Analytics*, 2(1–2), pp.13–27. Available at:

http://link.springer.com/10.1007/s41060-016-0029-7.

De Nicola, A., Missikoff, M. & Navigli, R., 2005. A Proposal for a Unified Process for Ontology Building: UPON. *DEXA 2005: Proceedings of the 16th International Conference on Database and Expert Systems Applications*, LNCS 3588, pp.655–664. Available at: http://link.springer.com/chapter/10.1007/11546924_64.

Noy, N., 2004. Semantic integration: a survey of ontology-based approaches. *SIGMOD Rec.*, 33(4), pp.65–70. Available at: citeulike-article-id:86456\nhttp://dx.doi.org/10.1145/1041410.1041421.

Pinto, H.S., Staab, S. & Tempich, C., 2004. DILIGENT : Towards a fine-grained methodology for DIstributed , Loosely-controlled and evolvInG Engineering of oNTologies. *16Th European Conference on Artificial Intelligence - Ecai*, pp.393–397.

Reijers, H.A., Slaats, T. & Stahl, C., 2013. Declarative modeling - An academic dream or the future for BPM? In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp. 307–322.

Schönig, S., 2016. Efficient and Customisable Declarative Process Mining with SQL Efficient and Customisable Declarative Process.

Shah, T. et al., 2014. A guiding framework for ontology reuse in the biomedical domain. *Proceedings of the Annual Hawaii International Conference on System Sciences*, pp.2878–2887.

Shamsfard, M. & Abdollahzadeh Barforoush, A., 2003. The state of the art in ontology learning: a framework for comparison. *The Knowledge Engineering Review*, 18(4), pp.293–316. Available at: http://www.journals.cambridge.org/abstract_S0269888903000687.

Shvaiko, P. & Euzenat, J., 2013. Ontology Matching: State of the Art and Future Challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1), pp.158–176. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6104044.

Sinha, A. & Paradkar, A., 2010. Use cases to process specifications in business process modeling notation. *ICWS 2010 - 2010 IEEE 8th International Conference on Web Services*, pp.473–480.

Staab, S. et al., 2000. Knowledge Processes and Ontologies. , pp.1–24. Available at:

http://www.aifb.uni-karlsruhe.de/WBS.

Tempich, C. et al., 2005. An Argumentation Ontology for DIstributed , Loosely-controlled and evolvInG Engineering processes of oNTologies (DILIGENT). *Ibis*, 2005(1), pp.241–256. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.5591.

Thomas, O. & Fellmann, M., 2007. Semantic EPC: Enhancing process modeling using ontology languages. *CEUR Workshop Proceedings*, 251.

Thomas, O. & Fellmann, M., 2009. Semantic Process Modeling – Design and Implementation of an Ontology-based Representation of Business Processes. *Business & Information Systems Engineering*, 1(6), pp.438–451.

Uschold, M. & Gruninger, M., 1996. Ontologies: principles, methods and applications. *The Knowledge Engineering Review*, 11(2), p.93. Available at: http://www.journals.cambridge.org/abstract_S0269888900007797.

Weijters, A.J.M.M., Van Der Aalst, W.M.P. & Medeiros, A.K.A. De, 2006. Process Mining with the Heuristics Miner Algorithm. *Technische Universiteit Eindhoven*, *Tech. Rep. WP*, 166, pp.1–34. Available at: http://cms.ieis.tue.nl/Beta/Files/WorkingPapers/Beta_wp166.pdf.

Wohlin, C. et al., 2012. *Experimentation in software engineering*,

Wong, W., Liu, W. & Bennamoun, M., 2012. Ontology learning from text. *ACM Computing Surveys*, 44(4), pp.1–36. Available at: http://dl.acm.org/citation.cfm?doid=2333112.2333115 [Accessed November 8, 2013].

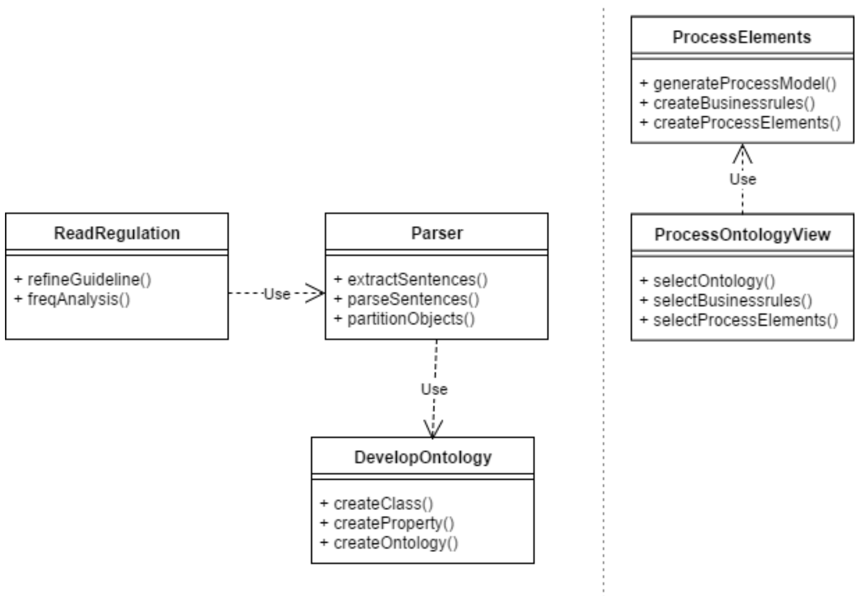
Zhou, L., 2007. Ontology learning: state of the art and open issues. *Information Technology and Management*, 8(3), pp.241–252. Available at: http://link.springer.com/10.1007/s10799-007-0019-5 [Accessed May 26, 2014].

APPENDICES

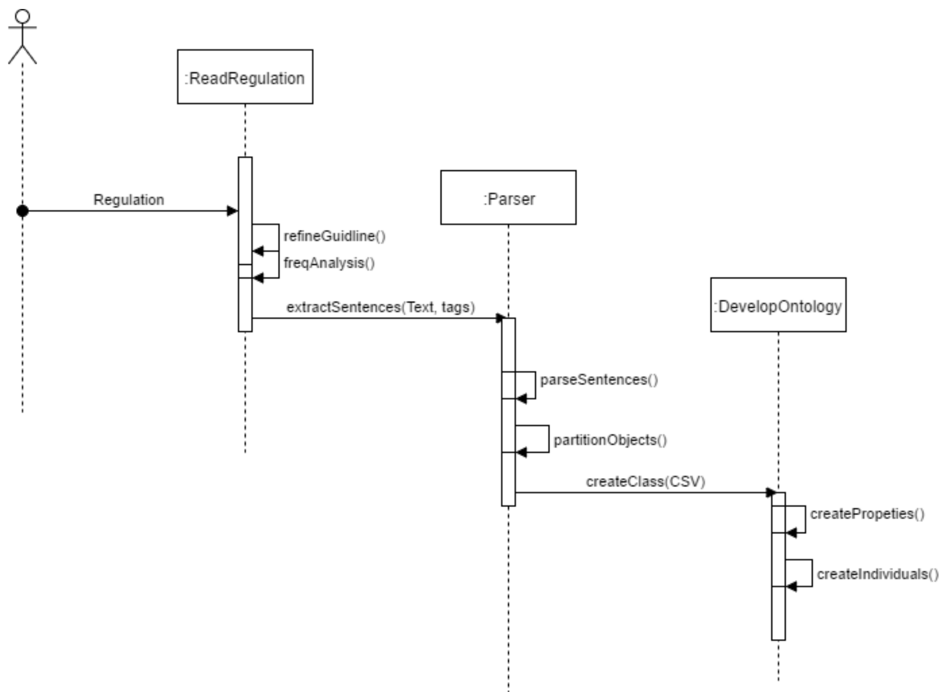
APPENDIX A

Class and Sequence Diagram for PrOnPo Tool

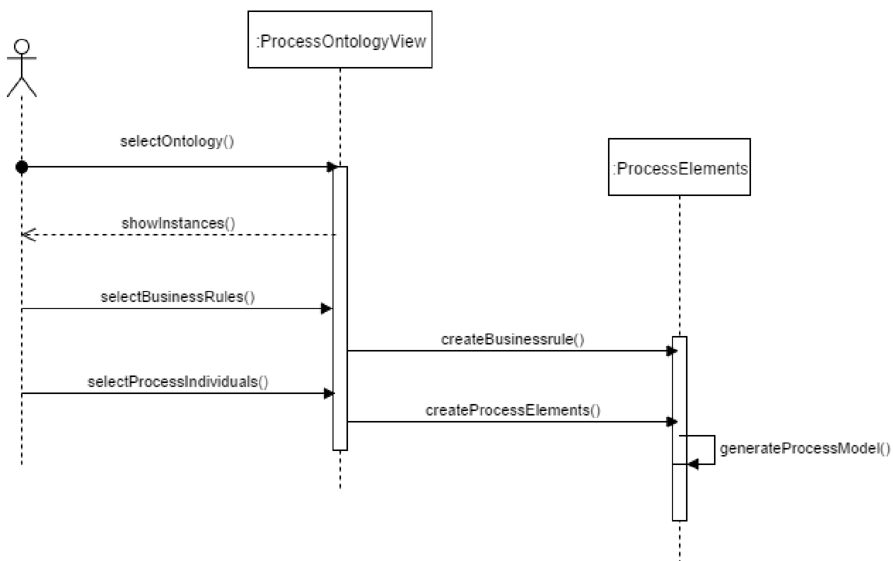
Class Diagram for Process Ontology Development & Process Model Generation



Sequence Diagram for Process Ontology Development



Sequence Diagram for Process Model Generation

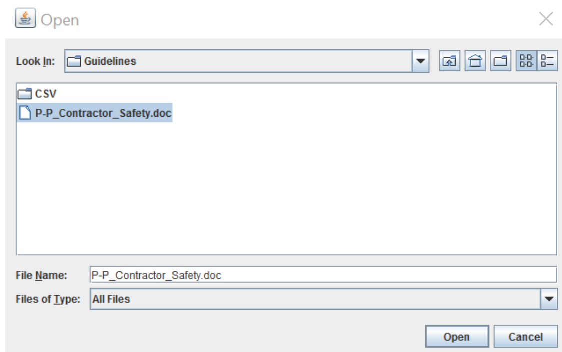


APPENDIX B

How to Use the Tool

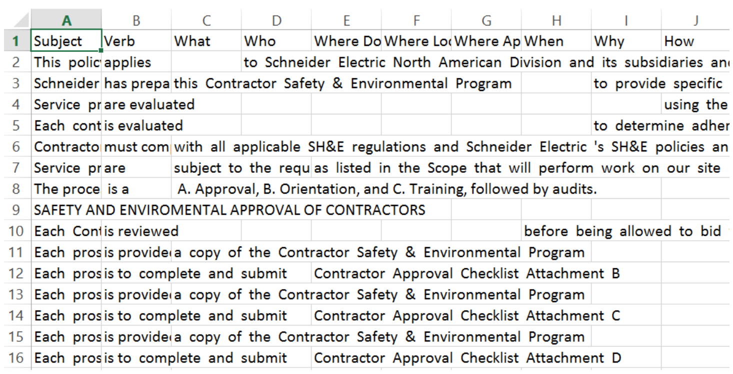
PrOnPo

1. Upload the guideline



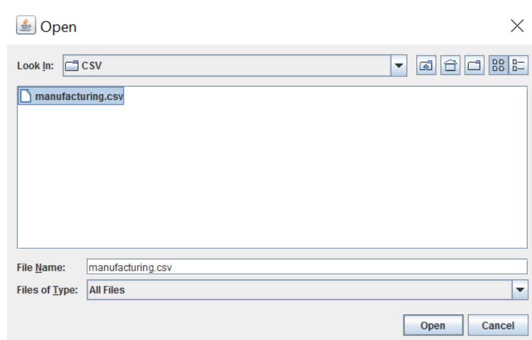
2. Click Open

3. Find and edit the generated CSV file



	A	B	C	D	E	F	G	H	I	J
1	Subject	Verb	What	Who	Where Do	Where Lo	Where Ap	When	Why	How
2	This polic	applies		to Schneider Electric North American Division and its subsidiaries an						
3	Schneider	has prepa	this Contractor Safety & Environmental Program						to provide specific	
4	Service pr	are evaluated								using the
5	Each cont	is evaluated							to determine adher	
6	Contracto	must com	with all applicable SH&E regulations and Schneider Electric 's SH&E policies an							
7	Service pr	are	subject to the requ		as listed in the Scope that will perform work on our site					
8	The proce	is a	A. Approval, B. Orientation, and C. Training, followed by audits.							
9	SAFETY AND ENVIROMENTAL APPROVAL OF CONTRACTORS									
10	Each Cont	is reviewed						before being allowed to bid		
11	Each pros	is provide	a copy of the Contractor Safety & Environmental Program							
12	Each pros	is to complete and submit			Contractor Approval Checklist Attachment B					
13	Each pros	is provide	a copy of the Contractor Safety & Environmental Program							
14	Each pros	is to complete and submit			Contractor Approval Checklist Attachment C					
15	Each pros	is provide	a copy of the Contractor Safety & Environmental Program							
16	Each pros	is to complete and submit			Contractor Approval Checklist Attachment D					

4. Upload the CSV file

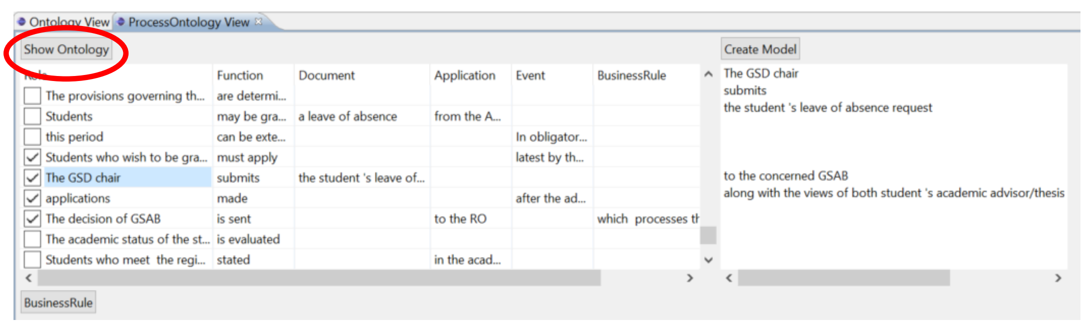


5. Click open and the tool will transform the CSV file to OWL/RDF file

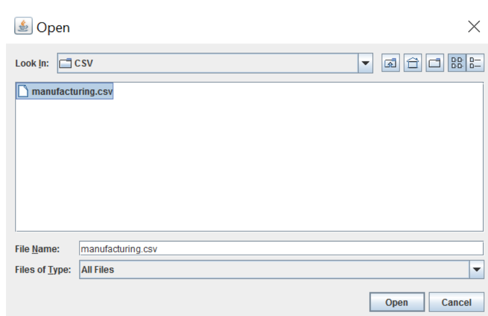
```
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .

<a_copy_of_Letter_of_Approval_and_an_updated_Contractor_Site_S
afety_Representative_form>
        a         <processontology.com/informationCarrier> ;
                  <processontology.com/isInputTo>
                  <submits> .

<must_renew>  a  <processontology.com/function> .

<upon_correction_of_the_deficiency>
        a
<processontology.com/event> ;
        <processontology.com/triggers>  <may_be_reinstated> .

<processontology.com/informationCarrier>
        a         owl:Class .

<shall_be_documented>
        a         <processontology.com/function> .
```

ProModGen

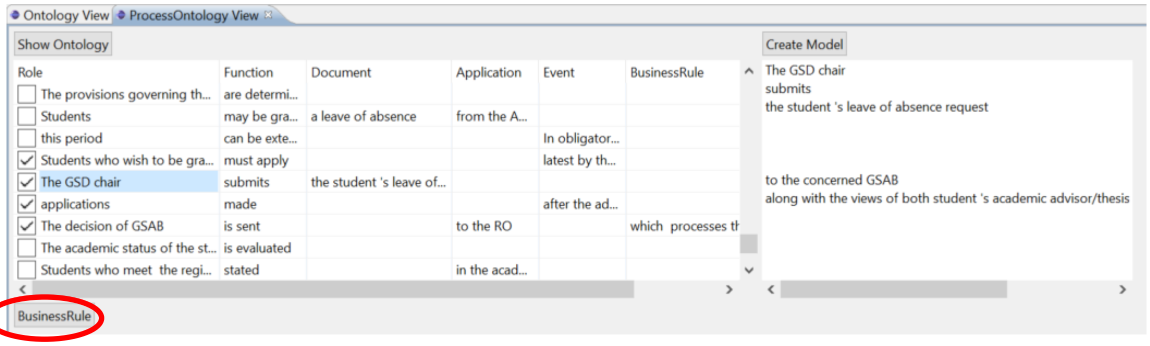
1. Click on Show Ontology Button



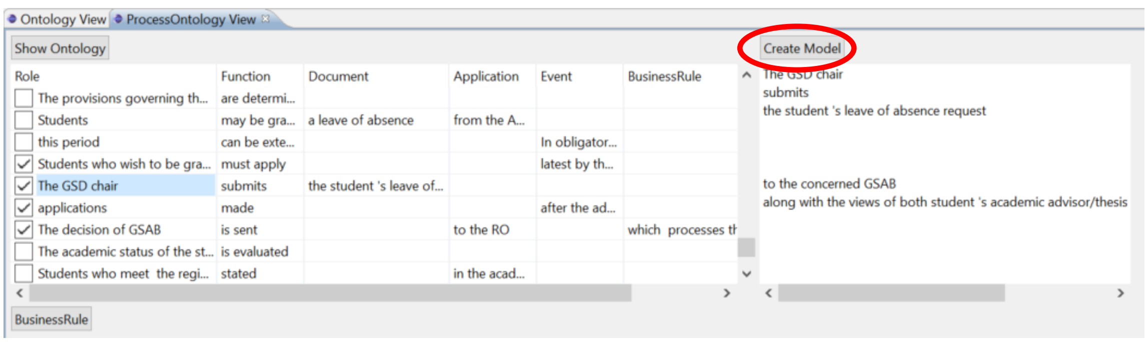
2. Upload the CSV file



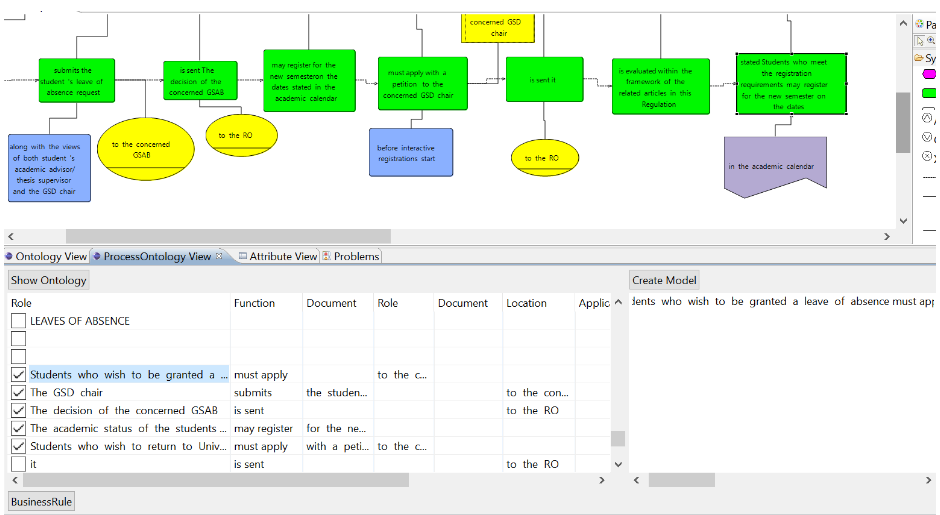
3. Select the instances to create business rules and click on BusinessRule button.



4. Select the instances to create process model elements and click on Create Model button

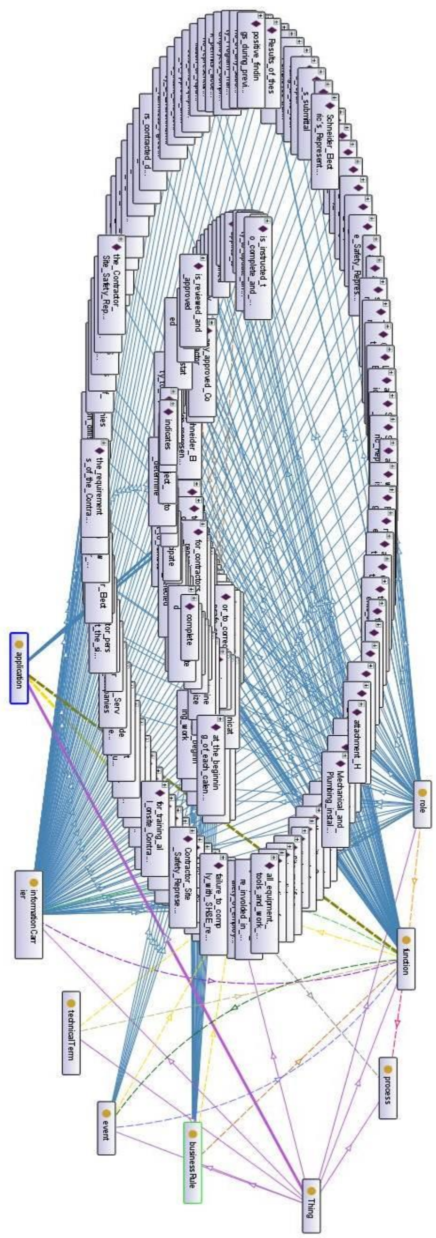


5. Finalize and edit the model



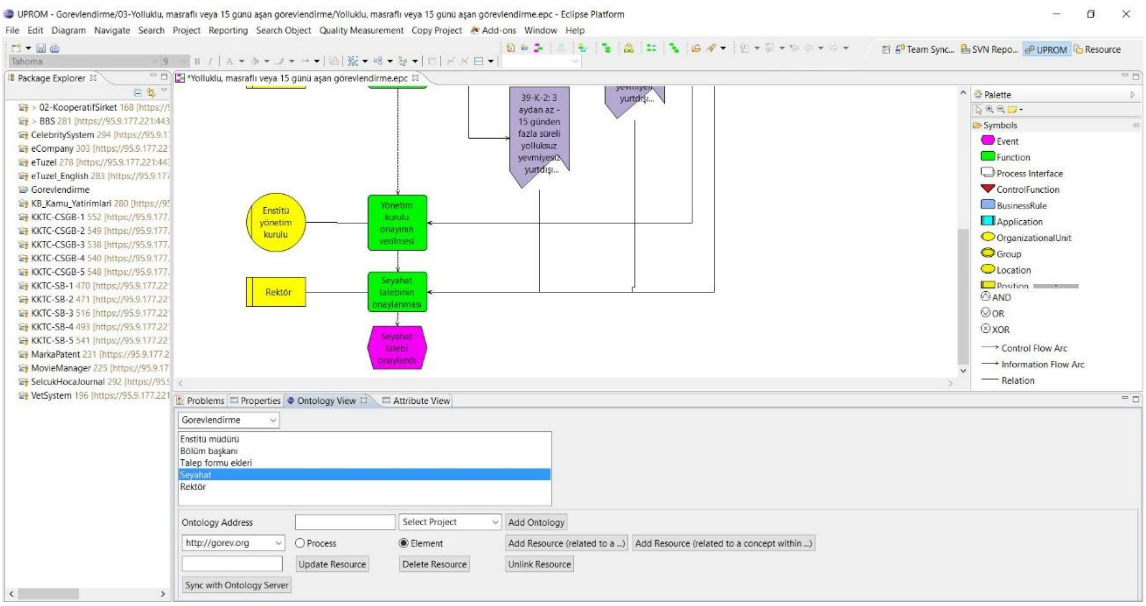
APPENDIX C

Ontology Graph for Manufacturing Processes



APPENDIX D

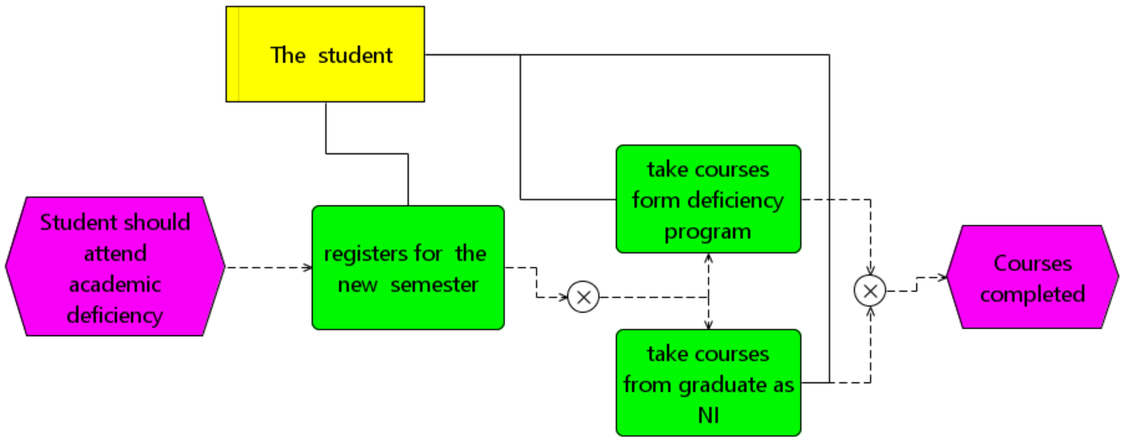
User Interface of PROMPTUM



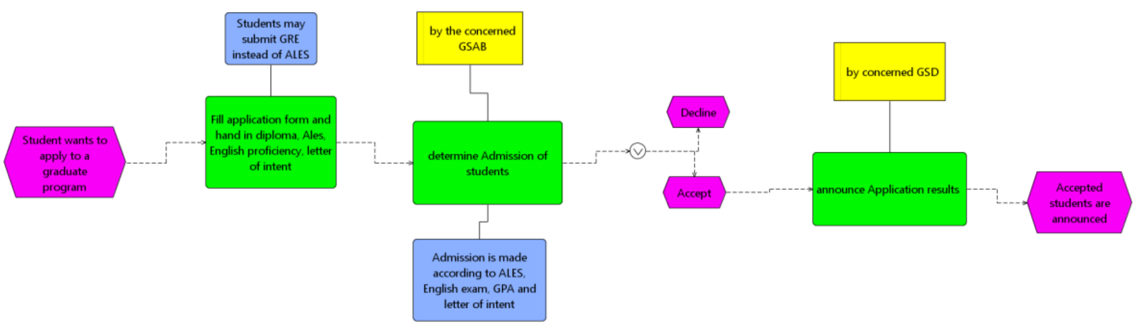
APPENDIX E

Process Models Generated as part of Exploratory Study 2

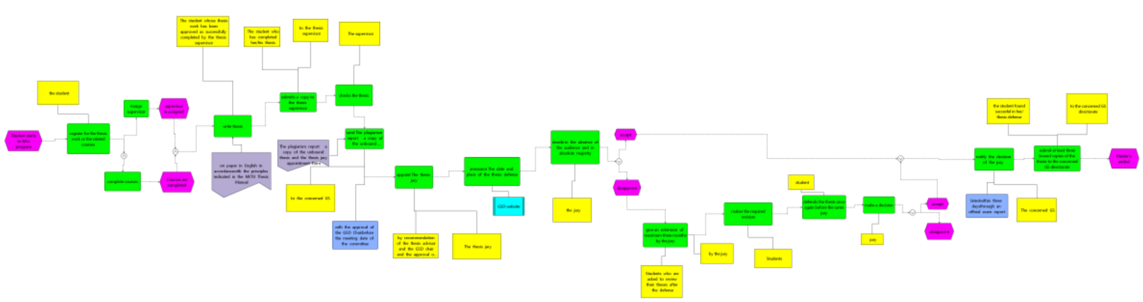
Academic Deficiency Process



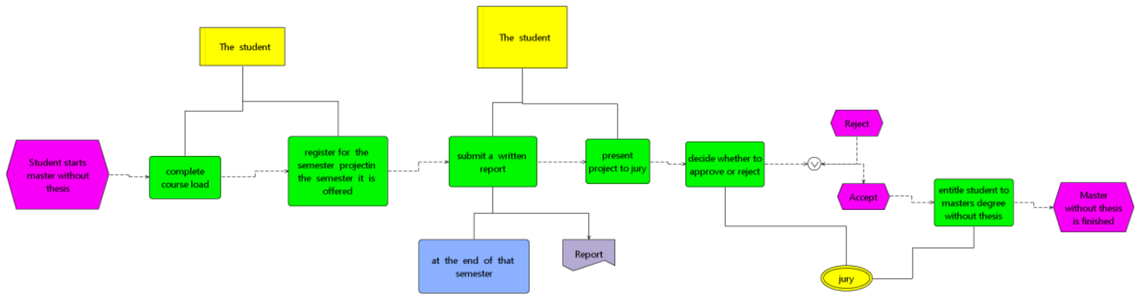
Application Process



Master Process

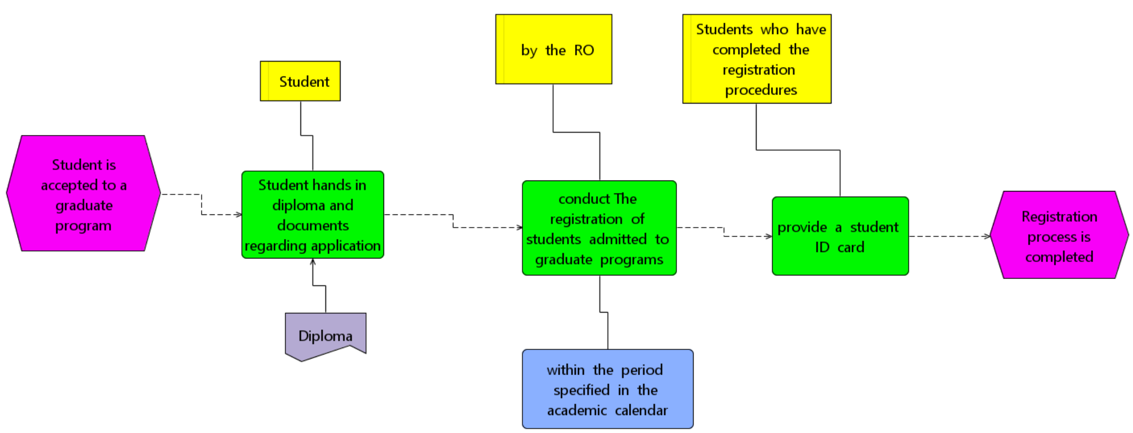


Master with Non Thesis Process

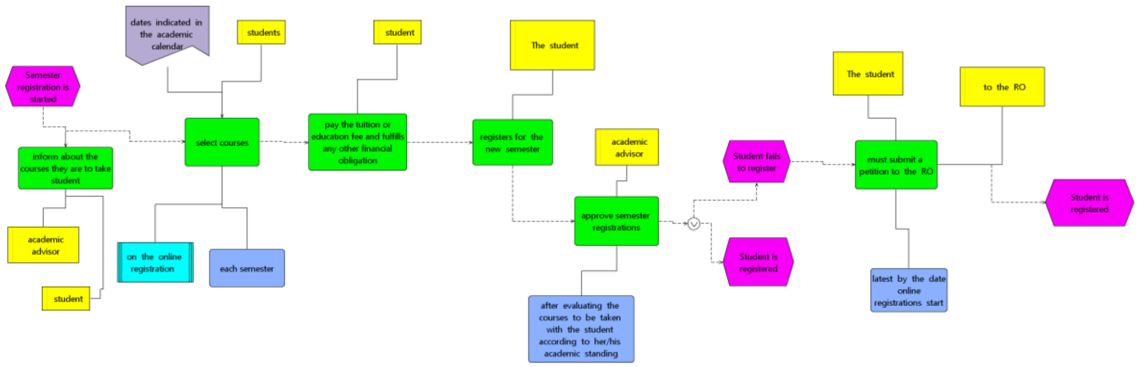


PhD Process

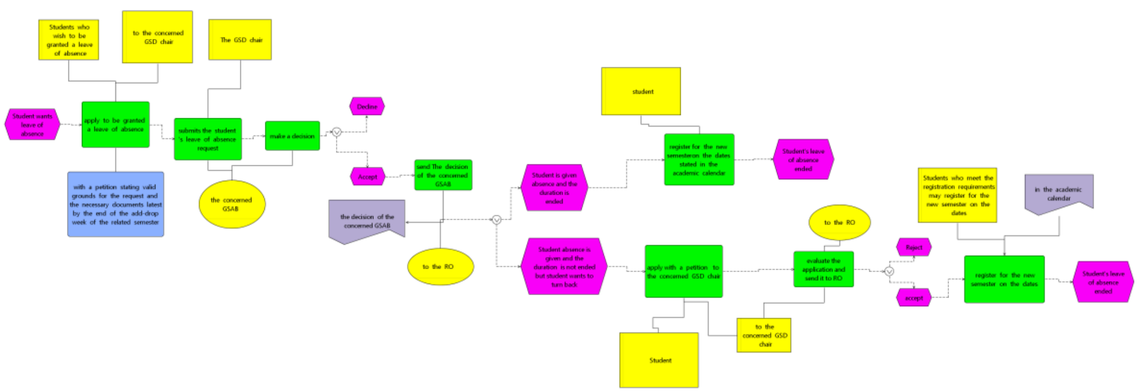
Registration Process



Semester Registration Process



Student's Absence of Leave Process

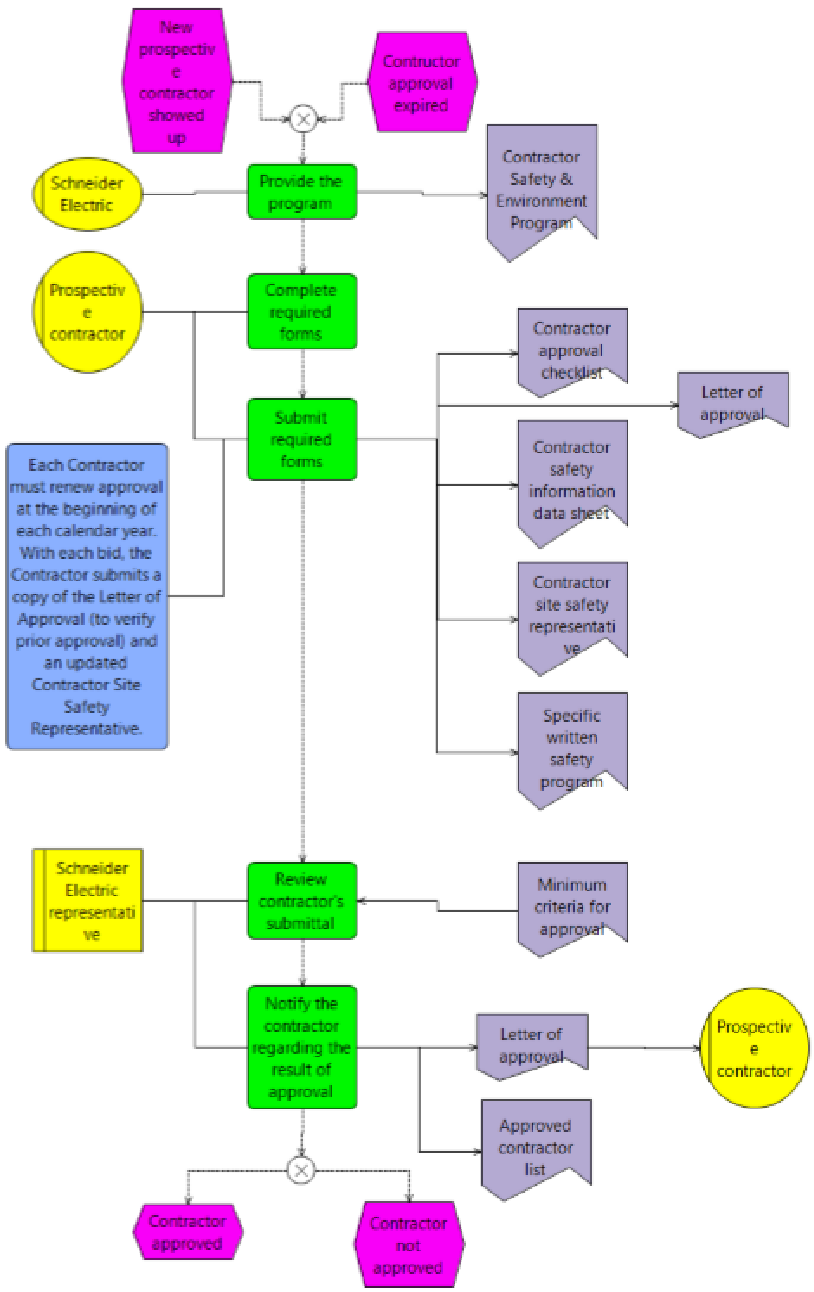


APPENDIX F

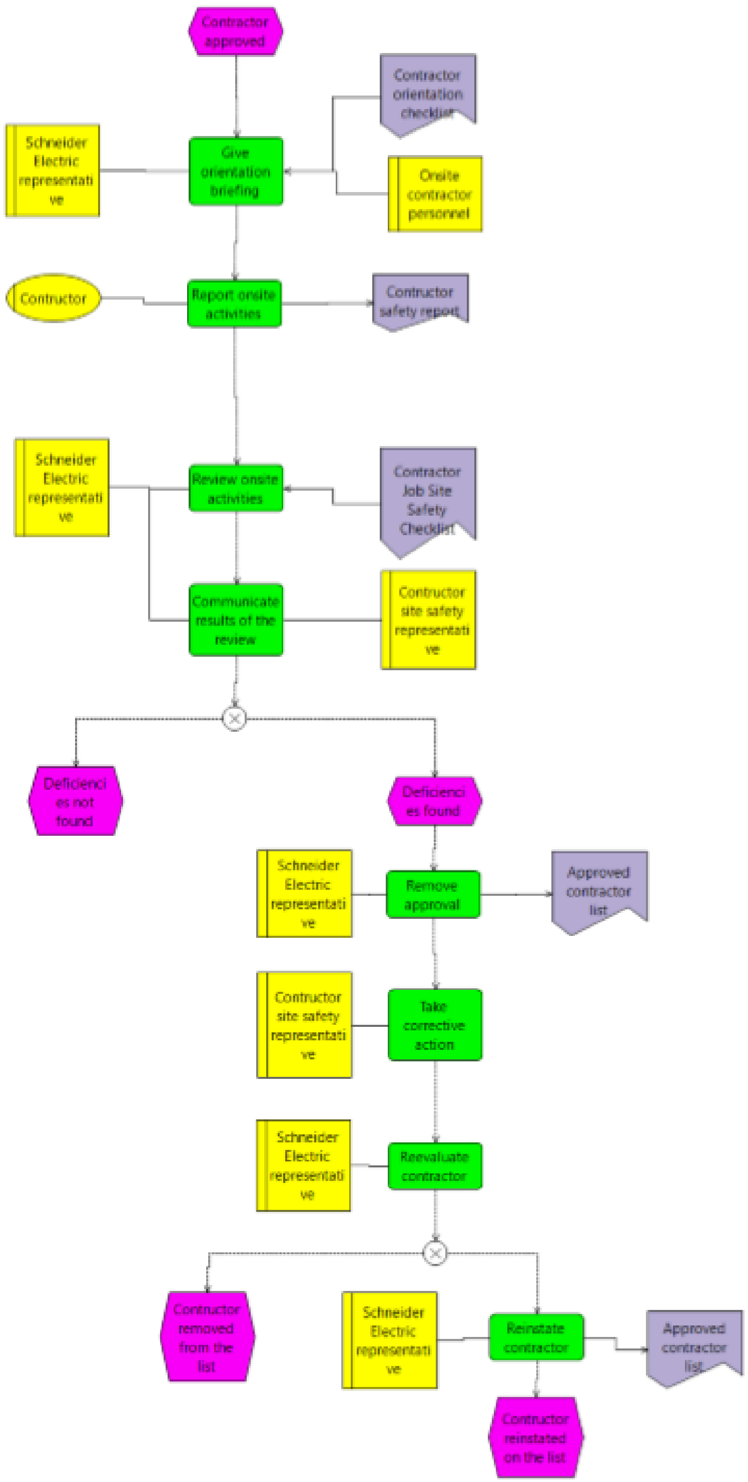
Existing Process Models of Manufacturing and Registration Processes

Manufacturing Processes

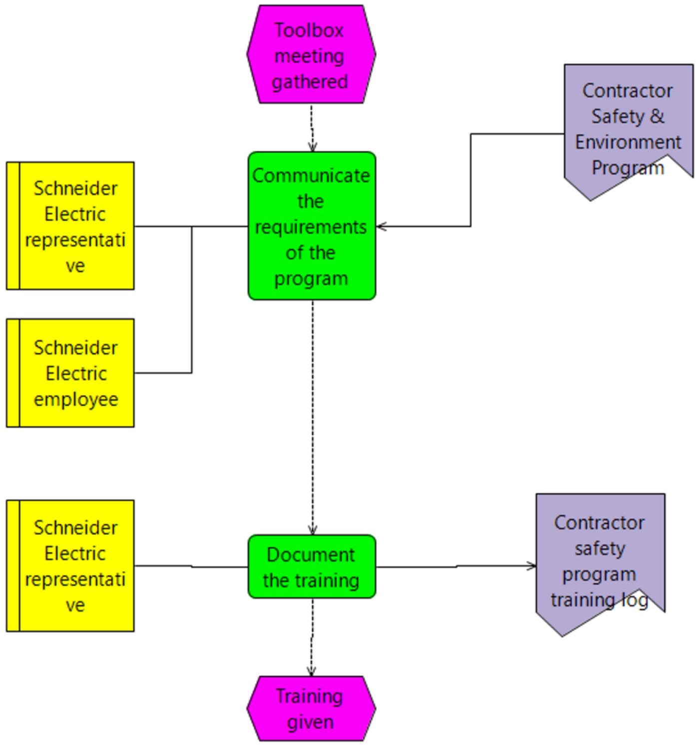
Contractor Approval Process



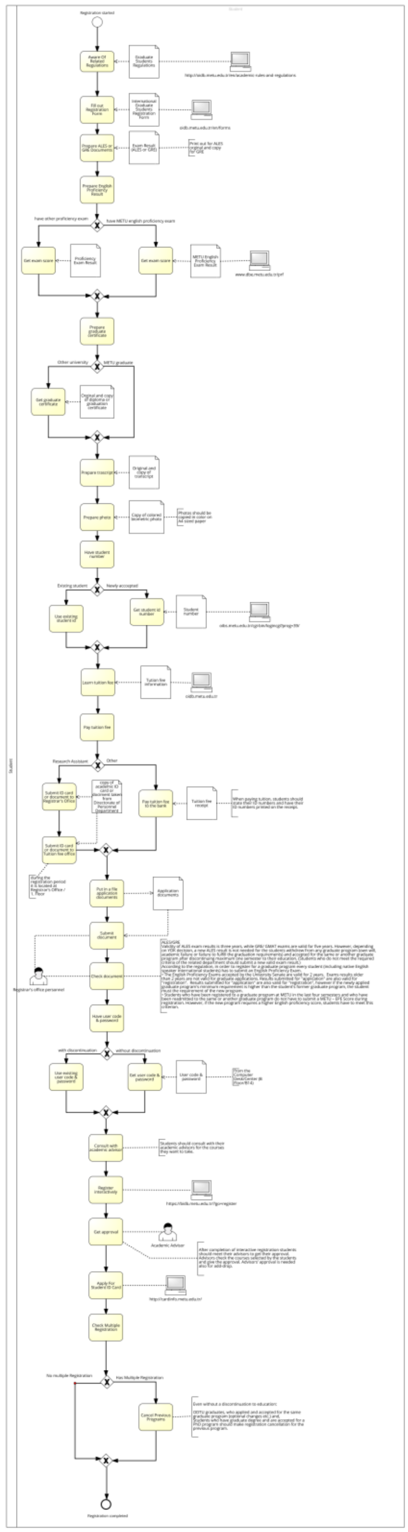
Contractor Orientation Process



Contractor Training Process



Registration Process

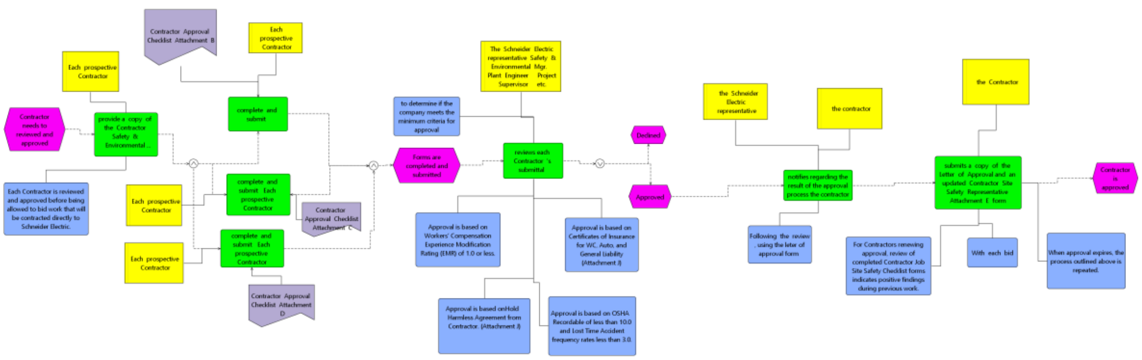


APPENDIX G

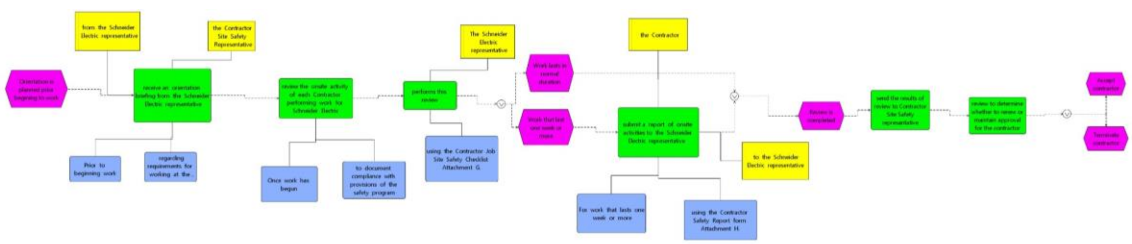
Process Models Generated as part of Multiple Case Study

Manufacturing Processes

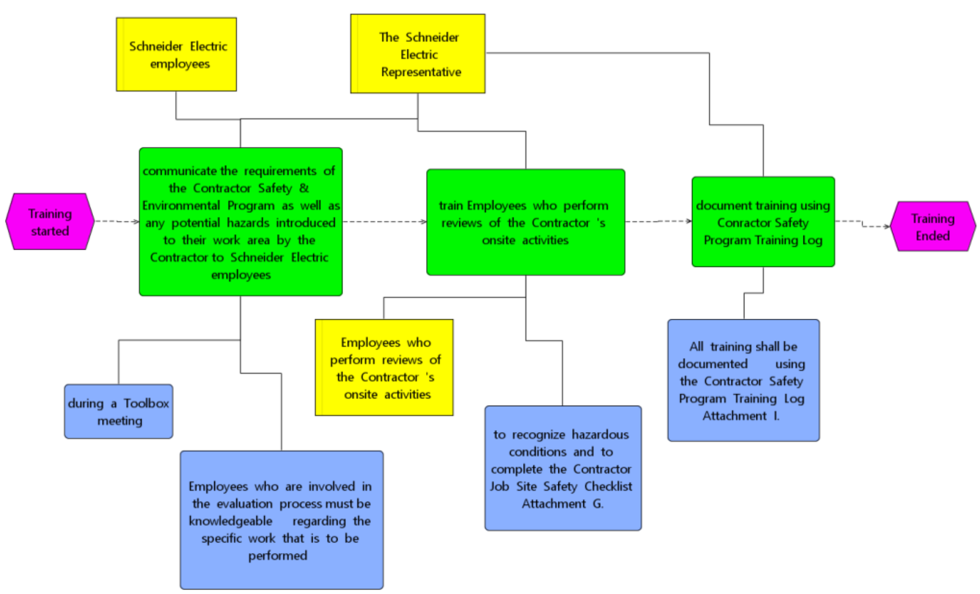
Contractor Approval Process



Contractor Orientation Process

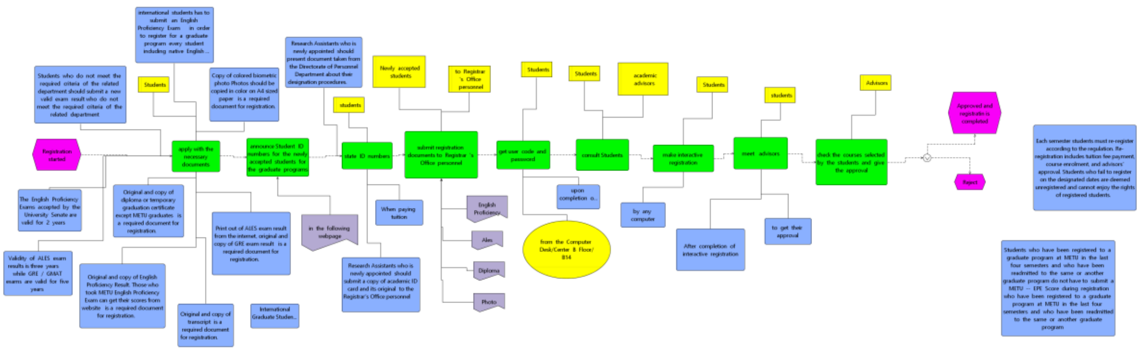


Contractor Training Process

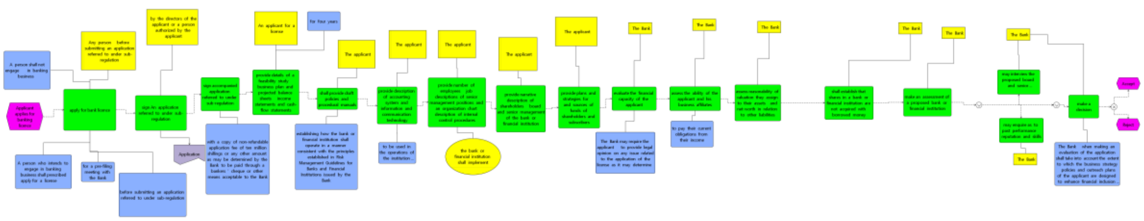


University Process

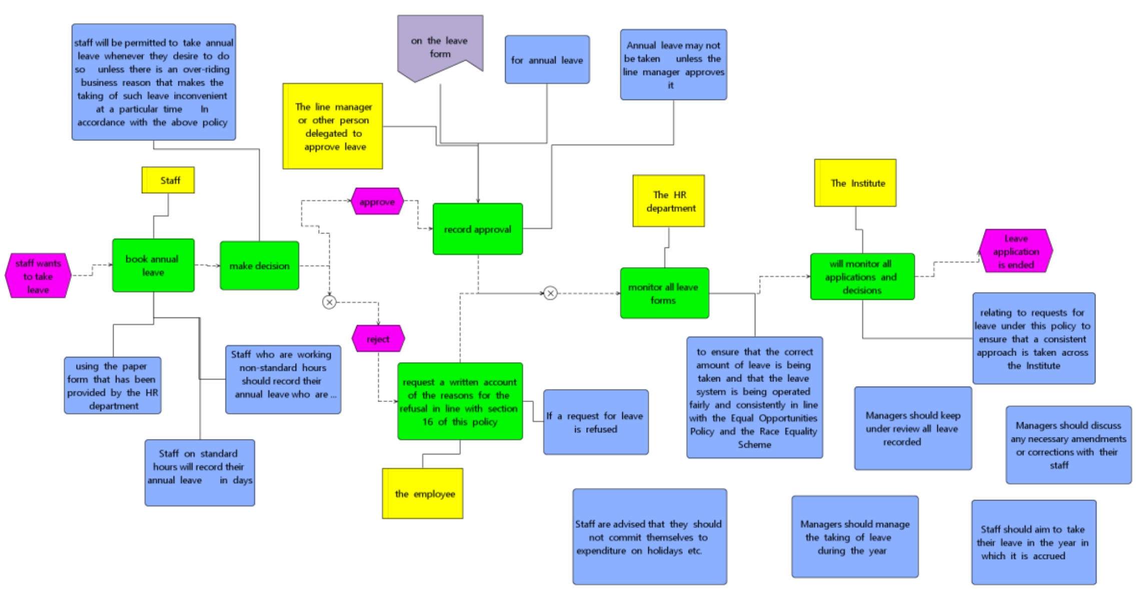
Registration Process



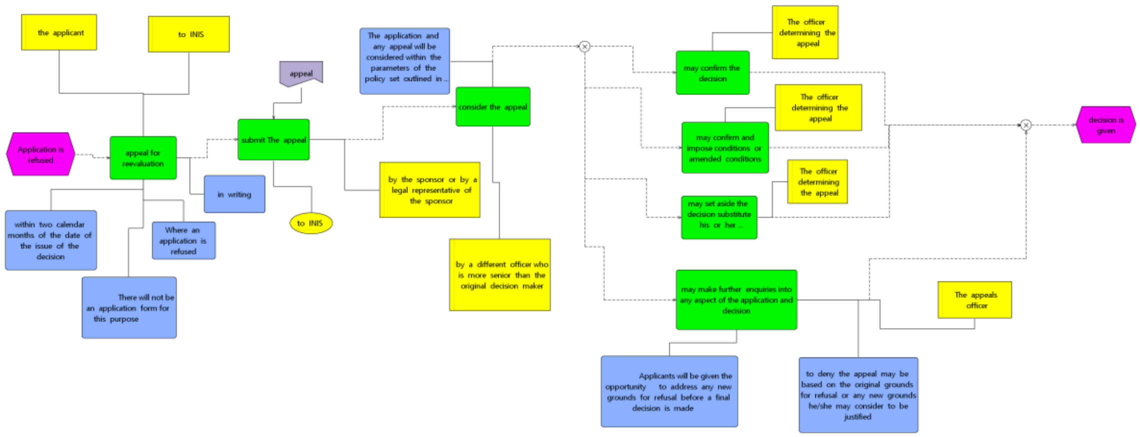
Finance Process



Human Resources Process



Visa Approval Process



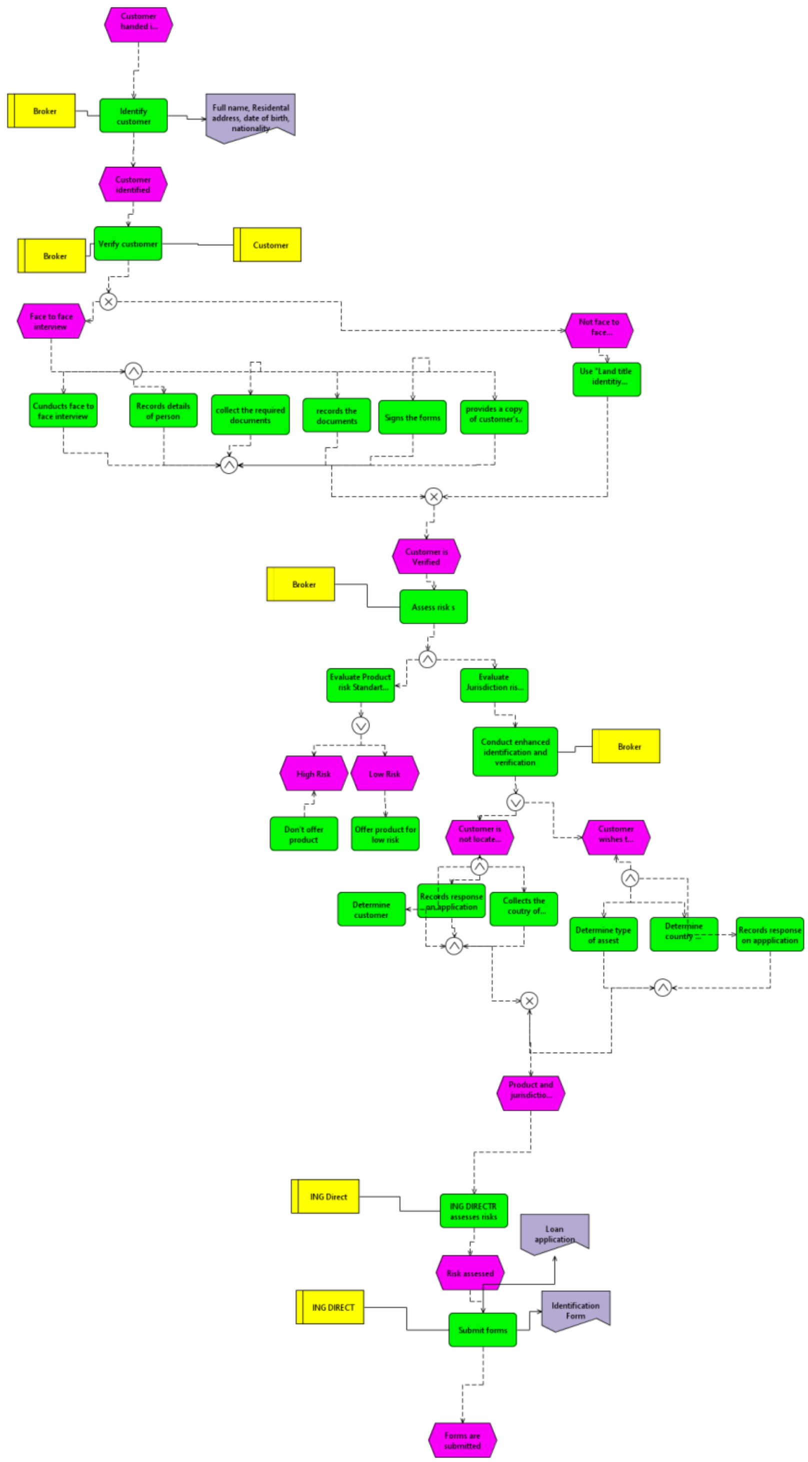
Appendix H

Activity Checklist for Customer Identification Process

Activity	M1	M2	M3	A1	A2	A3	T1	T2	T3
Conduct face to face interview	X	X	X	X	X	X	X	X	X
Use Australia's Post if face to face interview is unavailable	X	X	X	X	X	X	X	X	X
Assess product and jurisdiction risk	X	X	X	X	X	X	X	X	X
Conduct enhanced identification and verification if customer not located in Australia	X	X	X	X	X	X	X	X	X
Conduct enhanced identification and verification if asset not located in Australia	X	X	X	X	X	X	X	X	X
IngDirect assesses customer risk	X	X	X	X	X	X	X	X	-
Submit loan application, identification form and collected documents	X	X	X	X	X	X	X	X	X

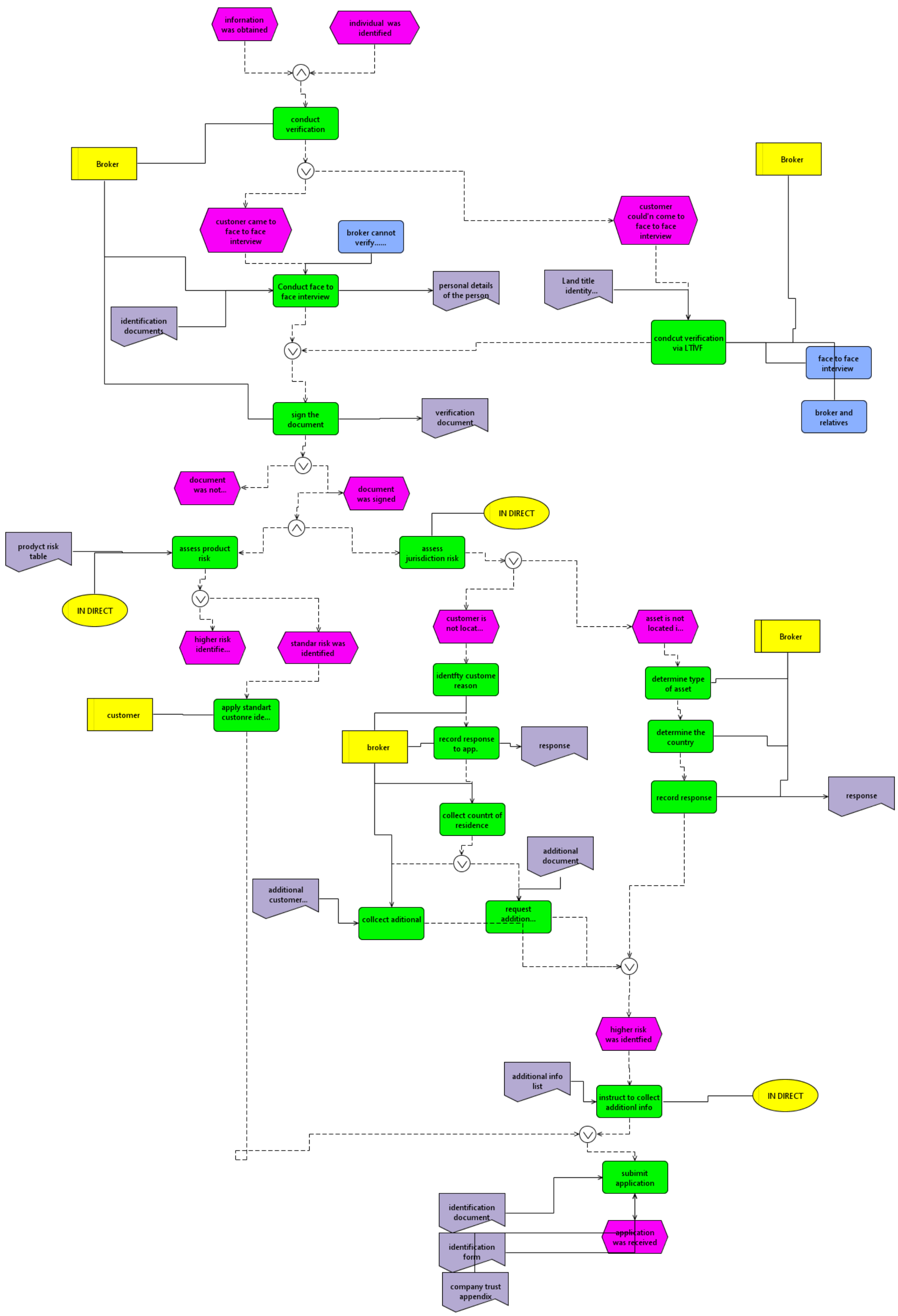
APPENDIX I

Traditional Group's Process Models

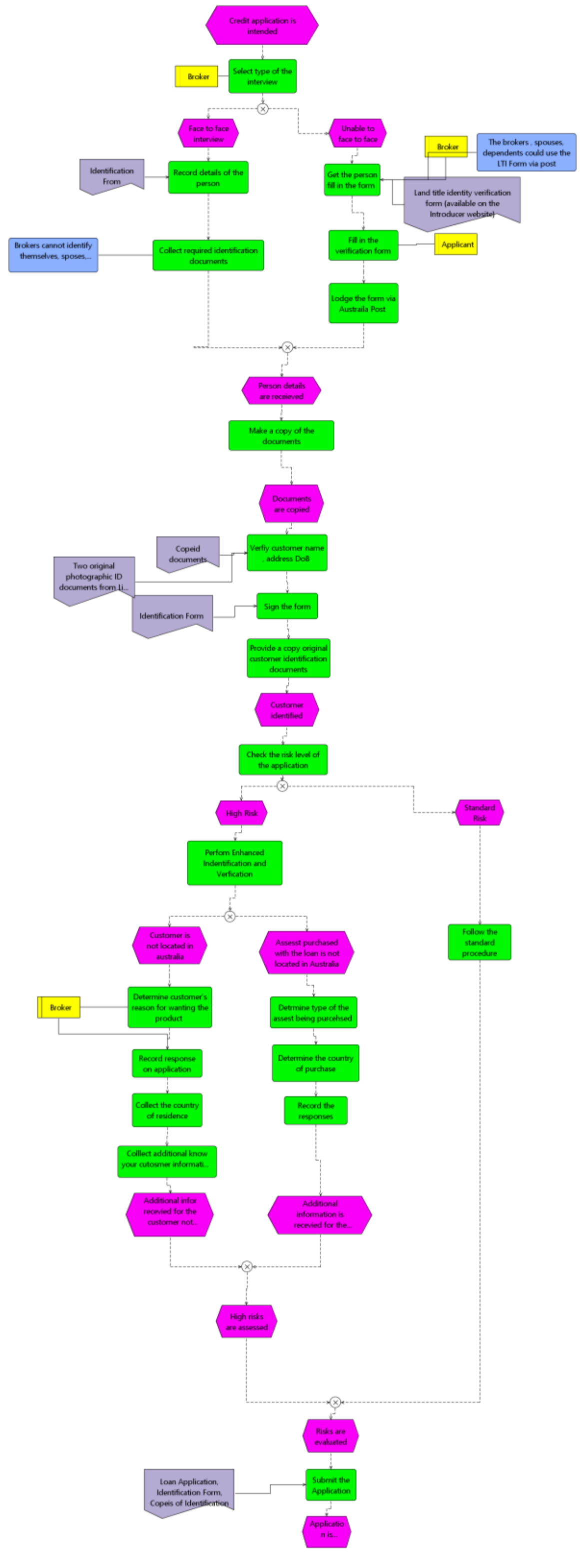


Traditional 1:

Traditional 2:



Traditional 3:

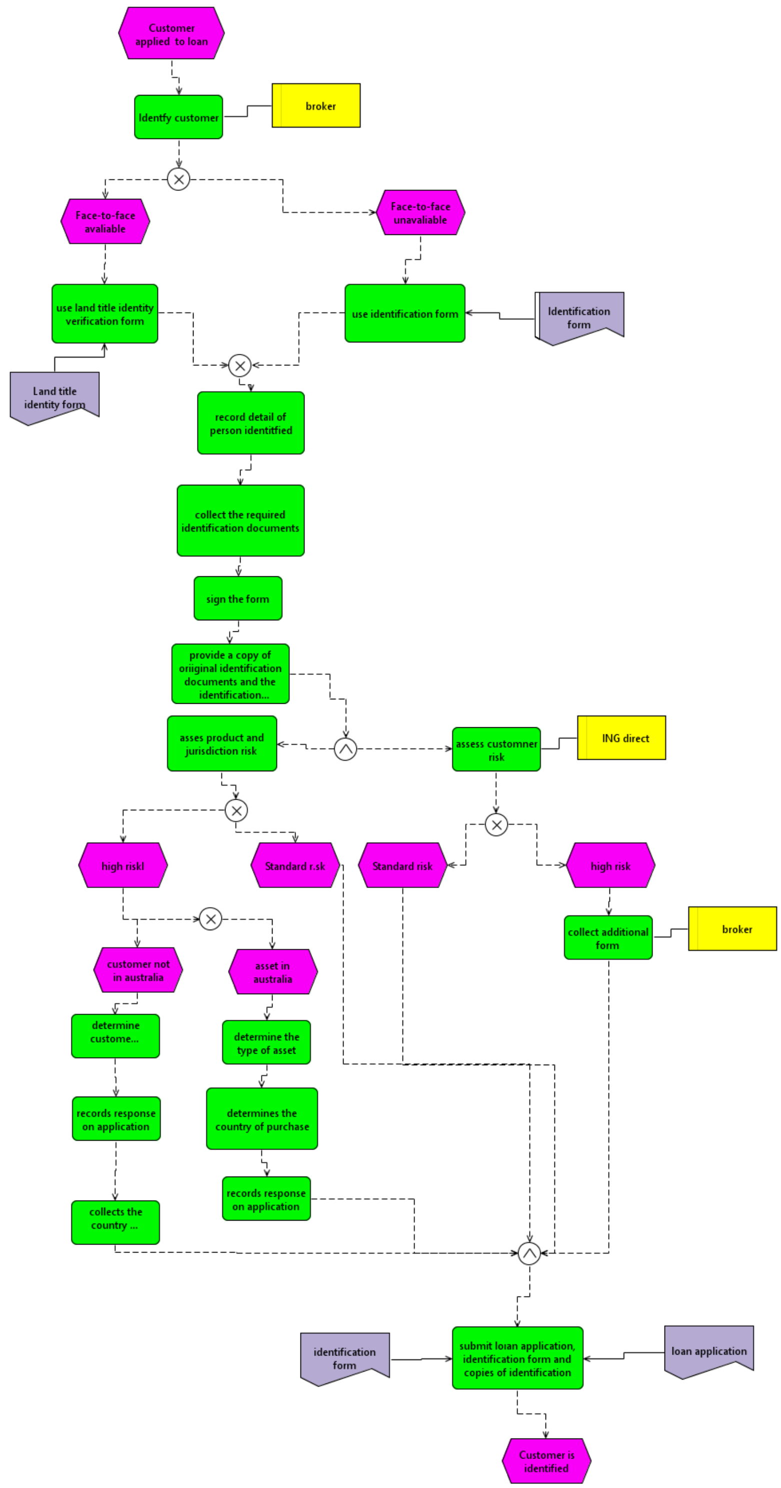


APPENDIX J

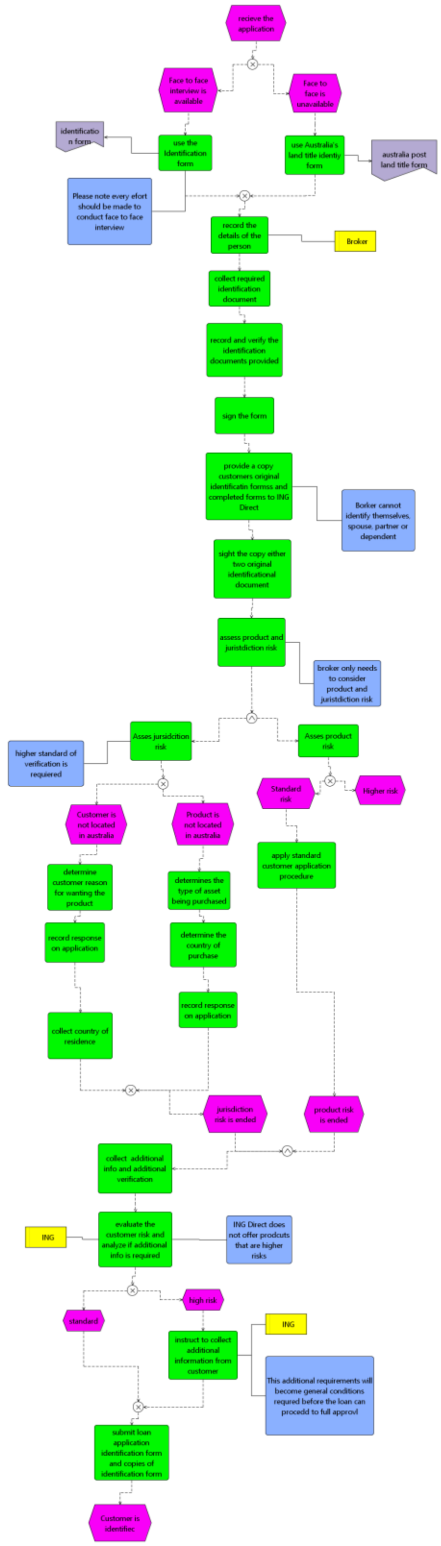
Manual Group's Process Models

Manual 1:

Manual 2:


Customer applied to loan
Identfy customer
broker
Face-to-face avaliable
Face-to-face unavaliable
use land title identity verification form
use identification form
Identification form
Land title identity form
record detail of person identitfied
collect the required identification documents
sign the form
provide a copy of original identification documents and the identification...
asses product and jurisdiction risk
assess customner risk
ING direct
high riskl
Standard risk
Standard risk
high risk
collect additional form
broker
customer not in australia
asset in australia
determine custome...
determine the type of asset
records response on application
determines the country of purchase
records response on application
collects the country ...
identification form
submit loan application, identification form and copies of identification
loan application
Customer is identified

Manual 3:

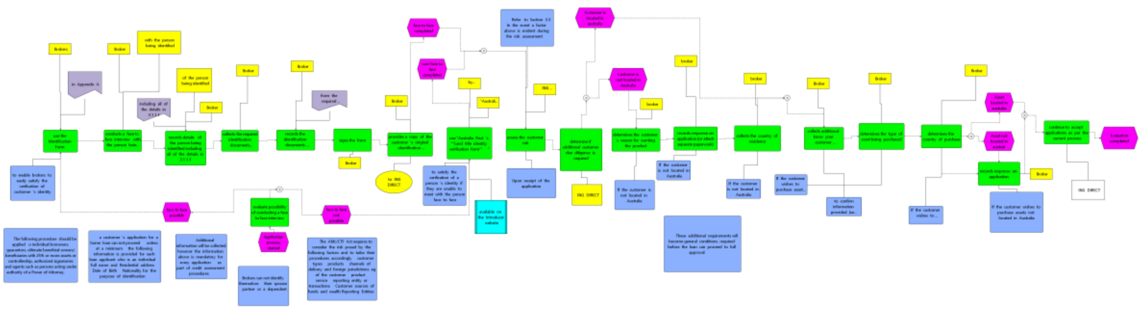


recieve the application
Face to face interview is available
Face to face is unavailable
identification form
use the Identification form
use Australia's land title identiy form
australia post land title form
Please note every efort should be made to conduct face to face interview
record the details of the person
Broker
collect required identification document
record and verify the identification documents provided
sign the form
provide a copy customers original identificatin formss and completed forms to ING Direct
Borker cannot identify themselves, spouse, partner or dependent
sight the copy either two original identificational document
assess product and juristdiction risk
broker only needs to consider product and juristdiction risk
Asses jursidcition risk
higher standard of verification is requiered
Asses product risk
Standard risk
Higher risk
Customer is not located in australia
Product is not located in australia
apply standard customer application procedure
determine customer reason for wanting the product
determines the type of asset being purchased
record response on application
determine the country of purchase
record response on application
collect country of residence
jurisdiction risk is ended
product risk is ended
collect additional info and additional verification
ING
evaluate the customer risk and analyze if additional info is required
ING Direct does not offer prodcuts that are higher risks
standard
high risk
ING
instruct to collect additional information from customer
This additional requirements will become general conditions requred before the loan can procedd to full approvl
submit loan application identification form and copies of identification form
Customer is identifiec

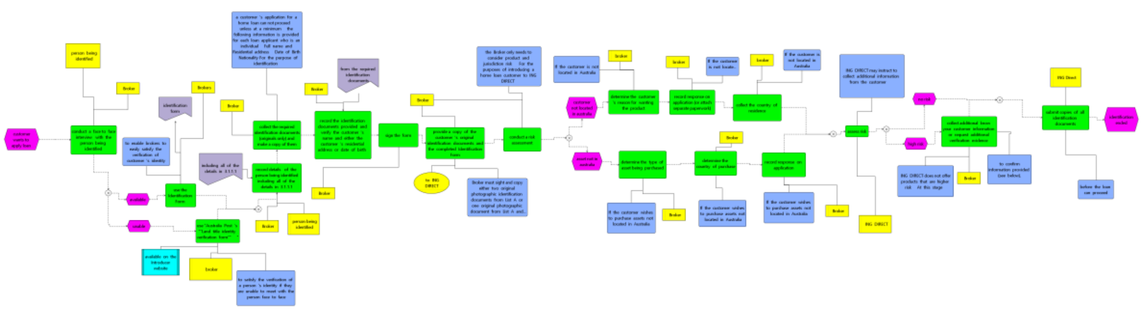
Appendix K

Auto Group's Process Models

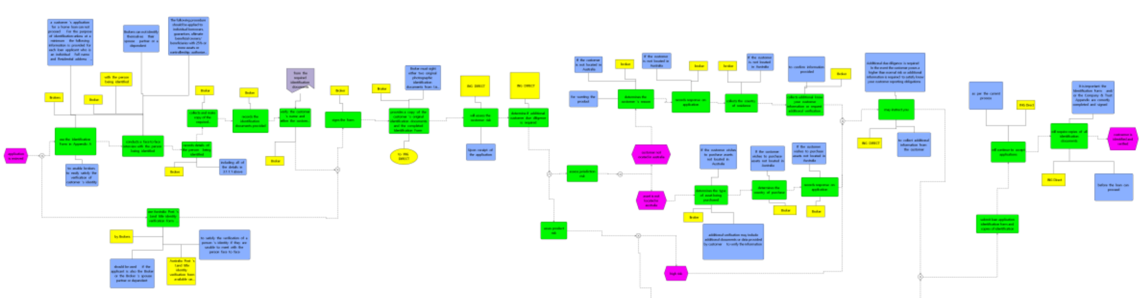
Auto 1:



Auto 2:



Auto 3:



CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Gürbüz, Özge
Nationality: Turkey
Date of Birth: January 30th 1988
Place of Birth: Karaman, Turkey
e-mail: gurbuzozge@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
PhD	METU, Information Systems	2017
MS	METU, Information Systems	2011
BS	Çankaya University, Computer Engineering	2009

WORK EXPERIENCE

Year	Place	Enrollment
2010-Present	METU, Informatics Institute, Ankara Turkey	Research Assistant
2012-2016	Bilgi Grubu Ltd., Ankara Turkey	Researcher, consultant, software engineer

FOREIGN LANGUAGE

English Fluent

PUBLICATIONS

1. Gürbüz O., Güceğlioğlu S., Demirörs O. : Uygulama Öncesinde Süreç Kalitesini Ölçmek için Tasarlanan Modelin İnsan Kaynakları Yönetimi Süreçleri Üzerinde Uygulanması, 5. National Software Engineering Symposium, Ankara, Türkiye, 26-28 September, 2011.
2. Gurbuz O., Guceglioglu S., Demirors O. : Application of Process Quality Measurement Frameworks for Human Resource Management Processes, IEEE International Conference on Quality and Reliability, 14-17 September, Bangkok, Thailand, 2011

3. Peker, S., Gürbüz Ö., Taşkaya Temizel, T., “A SOAP-based Social Traffic Network System”, IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA’14), Alberobello, Italy, June, 2014
4. Gürbüz, Ö., Güceğlioğlu, A. S., & Demirörs, O. “Otomatik Kalite Ölçümü: Uygulama Öncesinde Süreç Kalitesi Ölçüm Seti ile Kamu Kurumunda Durum Çalışması.” National Software Engineering Symposium (UYMS) 2014.
5. Coskuncay A., Gurbuz O., Demirors O. (2014). İş Süreç Modellerinden Ontolojiye Dönüşüm: Bir Durum Çalışması. 8. Ulusal Yazilim Muhendisligi Sempozyumu (UYMS) 2014, Güzelyurt/KKTC
6. Gurbuz, O., Demirors O. “Süreç Ontolojisi Oluşturma Yöntemi: Durum Çalışması”, Proceedings of the 10th Turkish National Software Engineering Symposium, Canakkale, October, 2016
7. Coskuncay A., Gurbuz O., Demirors O., Ekinci E.E. (2017). PROMPTUM Toolset: Tool Support for Integrated Ontologies and Process Models. In: Dumas M., Fantinato M. (eds) Business Process Management Workshops. BPM 2016. Lecture Notes in Business Information Processing, vol 281, pp 93-105. Springer
8. Gurbuz, O., Demirors O. “From Organizational Guidelines to Business Process Models: Exploratory Case for an Ontology Based Methodology”, 19th IEEE Conference on Business Informatics, Thessaloniki, July, 2017 (Accepted)
9. Gurbuz, O., Demirors O. "A Comparison of Process Ontology Discovery from Organizational Guidelines in Two Different Languages", 43rd Conference on Software Engineering and Advanced Applications (SEAA), Vienna, August 2017 (Accepted)

CERTIFICATES

1. CCNA Fundamentals of Network, Hame Polytechnic, Forssa/Finland, 26.08.2008
2. COSMIC Functional Size Measurer, Certificate, Common Software Measurement International Consortium, February 2011.
3. Practical Experiments on Software Engineering, Summer School Certificate, Free University of Bolzano, Italy, 2012
4. Verification of Systems and Applications, Summer School Certificate, Nancy/France, 2013
5. Linked Data Engineering, Open Hasso Plattner Institute, 2016
6. Business Process and Decision Modeling, Open Hasso Plattner Institute, 2016