

N. DEMİRBAŞ

DEVELOPING A DOMAIN ONTOLOGY FOR BUSINESS PROCESS
MANAGEMENT: A CASE STUDY ON BANKING

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY



NAGİHAN DEMİRBAŞ

A MASTER OF SCIENCE THESIS
IN
THE DEPARTMENT SOFTWARE ENGINEERING

ATILIM UNIVERSITY

2019

JUNE 2019

DEVELOPING A DOMAIN ONTOLOGY FOR BUSINESS PROCESS
MANAGEMENT: A CASE STUDY ON BANKING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

NAGİHAN DEMİRBAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF SOFTWARE ENGINEERING

JUNE 2019

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

Prof. Dr. Ali Kara
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Software Engineering, Atılım University.

Prof. Dr. Ali Yazıcı
Head of Department

This is to certify that we have read the thesis “Developing A Domain Ontology For Business Process Management: A Case Study On Banking” submitted by Nagihan Demirbaş and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Çiğdem Turhan
Supervisor

Asst. Prof. Dr. Erhan Gökçay
Software Eng. Department, Atılım University

Asst. Prof. Dr. Çiğdem Turhan
Software Eng. Department, Atılım University

Assoc. Prof. Dr. Erol Özçelik
Psychology Department, Çankaya University

Prof. Dr. Alok Mishra
Software Eng. Department, Atılım University

Asst. Prof. Dr. Güzin Tirkeş
Computer Eng. Department, Atılım University

Date: 26.06.2019

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.

Nagihan DEMİRBAŞ

ABSTRACT

DEVELOPING A DOMAIN ONTOLOGY FOR BUSINESS PROCESS MANAGEMENT: A CASE STUDY ON BANKING

Demirbaş, Nagihan

M.S., Department of Software Engineering

Supervisor: Asst. Prof. Dr. Çiğdem Turhan

June 2019, 94 pages

Business process management is a challenging task for organizations such as banks which have large collections of business process models. Organizations face problems which cannot be solved with current business process management technologies because of the limited degree of mechanization in BPM, creating insufficiency in the necessary evolution and dynamics of business processes. In other words, both querying and manipulating the process space regularly requires human labor, leading many times to slow, costly and imperfect situations. It also does not provide a uniform representation of an organization's process space on a semantic level, which would be accessible to intelligent queries.

Although, using formal languages, such as BPMN, Petri Net, UML activity diagram for modeling business processes, a missing semantic representation of process elements can prevent further interconnectivity and interoperability of business processes. Therefore, using ontologies can provide support to the business process management. By describing business process models in a machine readable and interpretable format which enables semantic annotations and computer reasoning, organizations like banks having hundreds of business process models, can achieve the desired effectiveness and agility

In this thesis, a domain ontology which provides machine readable semantic annotation of process models and efficient semantic analysis via logical queries is introduced for a selected bank sample.

Keywords: Business Process Management, Ontology, Semantic Analysis, Protégé



ÖZ

İŞ SÜRECİ YÖNETİMİ İÇİN ETKİ ALANI ONTOLOJİSİNİN GELİŞTİRİLMESİ: BANKACILIKLA İLGİLİ BİR ÖRNEK ÇALIŞMA

Demirbaş, Nagihan

Yüksek Lisans, Yazılım Mühendisliği Bölümü

Tez Yöneticisi: Dr. Öğr. Üyesi Çiğdem Turhan

Haziran 2019, 94 sayfa

İş süreci yönetimi, büyük iş süreci model koleksiyonuna sahip bankalar gibi kuruluşlar için zorlu bir iştir. Kuruluşlar, BPM'deki sınırlı bir mekanizasyon derecesi nedeniyle mevcut iş süreçleri yönetimi teknolojileriyle çözülemeyen problemlerle karşı karşıya kalmakta, gerekli evrim ve iş süreçlerinin dinamikleri konusunda yetersizlik yaratmaktadır. Başka bir deyişle, süreç alanını düzenli olarak sorgulamak ve değiştirmek, insan emeğini gerektirir ve çoğu zaman yavaş, maliyetli ve kusurlu durumlara yol açar. Ayrıca, bir kuruluşun süreç alanını akıllı sorgular için erişilebilir bir anlamsal düzeyde düzgün bir temsilini sağlamaz. Her ne kadar BPMN, Petri Net, iş süreçlerini modellemek için UML aktivite şeması gibi biçimsel diller kullanılsa da, süreç öğelerinin eksik bir anlamsal temsili, iş süreçlerinin daha fazla birbirine bağlanmasını ve birlikte çalışabilirliğini önleyebilir. Bu nedenle, ontolojileri kullanmak iş süreci yönetimine destek sağlayabilir. İş süreci modellerini, anlamsal ek açıklamalar ve bilgisayar muhakemesi sağlayan makinede okunabilir ve yorumlanabilir bir biçimde tanımlamak, yüzlerce iş süreci modeline sahip bankalar gibi kuruluşların, istenen etkinliği ve çevikliği elde etmelerinde yardımcı olur.

Bu tezde, seçili bir banka örneği için, makinenin okunabilir anlamsal süreç açıklamalarını ve süreç modellerinin mantıksal sorgularla anlamsal analiz ihtiyaçlarını destekleyen bir alan ontolojisini tanımlanmaktadır.

Anahtar Kelimeler: İş Süreçleri Yönetimi, Ontoloji, Anlamsal Analiz, Protégé



To My Dear Family,

ACKNOWLEDGEMENTS

I would like to express sincere appreciation to my supervisor Asst.Prof.Dr. ıgdem Turhan for her guidance, supportive and constructing approach throughout the research. I'm grateful for your faith in me, even when I'm being pessimistic.

I would like to thank to my thesis jury members Prof. Dr. Alok Mishra, Asst. Prof. Dr. Erhan Gökçay, Asst. Prof. Dr. Güzin Tirkeş, and Assoc. Prof. Dr. Erol Özçelik for their guidance in evaluation and review of my study.

Grateful thanks to my parents, Zeki Öztürk, and Melek and Cengiz Demirbaş and my husband Mustafa Fatih Demirbaş for their endless support, love and understanding. I appreciate your presence.

I am also grateful to my dear aunt, friend, and confidant Şirin Kara, for her continuous technical support, encouragement, and patience during this period. She is the best motivation for me to not to give up.

TABLE OF CONTENTS

ABSTRACT	iii
ÖZ	v
ACKNOWLEDGEMENTS	vii
TABLE OF CONTENTS.....	viii
LIST OF TABLES.....	x
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xiii
CHAPTER 1.....	1
INTRODUCTION	1
1.1 Overview	1
1.2 Motivation.....	3
1.3 Organization.....	4
CHAPTER 2.....	5
ONTOLOGIES.....	5
2.1 What Are Ontologies?	5
2.2 Why use ontologies?	7
2.3 Types and Categories of Ontologies	8
2.4 Ontology Development Methodology.....	13
2.5 Ontology Languages	16
2.6 Ontology Tools	22
2.7 Ontology Reasoners	26
CHAPTER 3.....	29
BUSINESS PROCESSES.....	29
3.1 What is Business Process?.....	29
3.2 What is Business Process Management?.....	30
3.3 Business Process Models.....	31
3.4 Business Process Languages.....	32

3.5 Business Process Modeling Tools.....	35
3.6 Business Process Management Life Cycle	36
3.7 Ontologies and Business Process Management	37
CHAPTER 4.....	40
ANALYSIS, DESIGN AND IMPLEMENTATION OF THE SYSTEM	40
4.1 Analysis of Domain.....	40
4.2 Implementation of the Ontology	45
CHAPTER 5	69
EVALUATION AND VERIFICATION OF THE ONTOLOGY	69
5.1 Ontology Verification.....	69
5.2 Evaluation of the Ontology.....	70
CHAPTER 6.....	81
CONCLUSION AND DISCUSSION.....	81
REFERENCES	84
APPENDIX A	91
RAW DATA TABLES	91
APPENDIX B.....	94
TURKISH VERSION OF QUESTION SET	94

LIST OF TABLES

Table 2.1 Classification of ontology languages	18
Table 2.2 A Comparison between Ontology Languages	22
Table 2.3 Comparison of ontology tools.....	25
Table 2.4 Comparison of ontology reasoners.....	28



LIST OF FIGURES

Figure 2.1 Graphical representation of ontology by Haller	6
Figure 2.2 The ontology spectrum that defined by Lassila and McGuinness	9
Figure 2.3 Uses for ontologies.....	11
Figure 2.4 Types of ontologies	12
Figure 2.5 Top-down approach	15
Figure 2.6 Bottom-up approach.....	15
Figure 2.7 Classification of ontology languages	17
Figure 2.8 The Semantic Web Stack from Tim Berners-Lee.....	19
Figure 2.9 The three OWL sublanguages	20
Figure 3.1 Basic BPMN notations	34
Figure 3.2 The BPM life cycle	36
Figure 4.1 Process overview	41
Figure 4.2 Process details.....	42
Figure 4.3 Process revision tracking.....	42
Figure 4.4 Process flow chart	43
Figure 4.5 Main processes.....	46
Figure 4.6 Subclasses of deposit main process	46
Figure 4.7 Subclasses of credit main process.....	47
Figure 4.8 Subclasses of accounting main process.....	47
Figure 4.9 Subclasses of alternative distribution channels main process	47
Figure 4.10 Subclasses of credit cards main process.....	48
Figure 4.11 Subclasses of payment systems main process	48
Figure 4.12 Subclasses of treasury / securities and fund management main process.....	48
Figure 4.13 Subclasses of support main process	49
Figure 4.14 Create disjoint classes	49
Figure 4.15 Subclasses of the Business Units Class	50
Figure 4.16 Subclasses of the Documents class	51
Figure 4.17 Subclasses of the Roles class	51
Figure 4.18 Subclasses of the Application Tools class.....	52
Figure 4.19 Create class tab	52
Figure 4.20 Object property characteristics	53
Figure 4.21 Object properties defined in the ontology	54
Figure 4.22 Business Processes class and its relation with other classes	55
Figure 4.23 Description of AnahtarKelimeler class	60
Figure 4.24 Description of individual Adk	60

Figure 4.25 Data properties defined in the ontology	62
Figure 4.26 Description of <i>Tanimlar</i> class.....	63
Figure 4.27 Description of individual <i>Atmok</i>	64
Figure 4.28 Individuals defined under 01.Mevduat main process	65
Figure 4.29 Individuals defined under <i>IsBirimleri</i> class	66
Figure 4.30 Individuals defined under <i>Dokumanlar</i> class	66
Figure 4.31 Individuals defined of <i>Roller</i> class.....	67
Figure 4.32 Individuals that give reference to the role <i>Opk</i>	67
Figure 4.33 Individual 01.02.11.SupheliCekUyariBilgisiGirisIslemleriSureci	68
Figure 4.34 Individual 01.01.01.MusteriNumarasiVermeSureci	68
Figure 5.1 Sample queries set.....	70
Figure 5.2 Result of the query: What are the processes that <i>give reference</i> to 01.01.01.MusteriNumarasiVermeSureci?	72
Figure 5.3 Result of the query: What are the business units that <i>own</i> the 01.01.01.MusteriNumarasiVermeSureci?	73
Figure 5.4 Result of the query: What are the processes that <i>belong</i> to the MusteriVeHesapOperasyonlariMudurlugu?	74
Figure 5.5 Result of the query: What are the processes, <i>in force</i> status, which involve role <i>Opk</i> ?	75
Figure 5.6 Result of the query: What are the processes, <i>in force</i> status, which involve application tool <i>Ets</i> ?.....	76
Figure 5.7 Result of the query: What are the processes, <i>in force</i> status, which involve 6415SayiliTerorizminFinansmanininOnlenmesiHakkindaKanun law?.....	77
Figure 5.8 Result of the query: What are the processes, <i>in force</i> status, which involve <i>Masak definition</i> ?.....	78
Figure 5.9 Result of the query: What are the processes that <i>give reference</i> to processes that <i>in canceled</i> status?.....	78
Figure 5.10 Result of the query: What are the individuals that give reference to <i>Merkezilesme</i> keyword?	79
Figure 5.11 Result of the query: What are the processes, <i>in force</i> status, with version number zero and <i>revision date</i> older than 2017?.....	80

LIST OF ABBREVIATIONS

BPD	-	Business Process Diagram
BPMN	-	Business Process Modeling Notation
BPO	-	Business Process Ontology
BRSA	-	Banking Regulation and Supervision Agency
CWA	-	Close World Assumption
DAML	-	DARPA Agent Markup Language
DL	-	Description Logics
EPC	-	Event Driven Process Chain
OIL	-	Ontology Inference Layer
OWA	-	Open World Assumption
OWL	-	Web Ontology Language
RDF	-	Resource Description Framework
RDFS	-	RDF schema
SWRL	-	Semantic Web Rule Language
UML	-	Unified Modeling Language
UNA	-	Unique Name Assumption

W3C	-	World Wide Web Consortium
WFM	-	Workflow Management
WFMS	-	Workflow Management System
WSML	-	Web Service Modeling Language
XML	-	Extensible Markup Language



CHAPTER 1

INTRODUCTION

1.1 Overview

Business processes are key aspects of modern companies. Generally speaking, a business process is modeled as a series of interconnected activities using data and control dependencies.

There are many business process modeling notations and languages. Business Process Modeling Notation (BPMN) [1], Petri Net [2], Activity Oriented Process Chain (EPC) [3] and Unified Modeling Language (UML) [4] are some of them. Each language provides different concepts, syntax, and complexity to model business processes.

Lately, more and more researchers have realized that traditional process modeling languages provide only a monolithic view of the real process of interest. The process is “flattened” to allow for a diagram that describes its lifecycle [5].

Process modeling mostly focuses on the graphical representation of processes by using modeling languages that cannot capture the domain semantics, e.g. BPMN. Consequently, process definitions do not supply machine readable semantics that can support business process representatives in the analysis and maintenance of processes [6]. Parts of languages such as BPMN, UML, BPEL and EPC have been formalized by various authors. These formalizations, however, typically include only selected, small parts of the languages at the theoretical level. Actually, the problems faced by companies using process models are not limited to this. Process models are usually

designed only to document the process flows in the company. This is often due to norms of quality management or because of the laws and regulations laid down by the competent authorities.

Also, process models in most companies are not executable. They can be used as a starting point for model transformation to obtain more platform-independent and even platform-specific models, but as described in most cases they are intended to be used only for documentation. As a result, because they are designed to document the process flow in the company and are not executable, process models cannot be automatically generated and they cannot be kept up-to-date.

Business Process Management (BPM) is a continuous effort; that is, processes need to be managed and BPM does not stop after completing the process model or system implementation.

BPM can be seen as an extension of Workflow Management (WFM). While WFM primarily focuses on the automated execution of business processes in a rather mechanistic manner without much attention for human factors and management support, Business Process Management (BPM) aims at supporting the whole life-cycle necessary to apply and maintain business processes in organizations from a managerial perspective.

Organizations are living systems that are constantly changing. Because of changes in organization departments, process activities, used systems, regulations, etc, the process space of an organization may also undergo changes. Changing circumstances may trigger process adaptations and generate new analysis questions. Therefore, it is important to continuously monitor process.

To keep process models up to date, a business process may undergo a series of modifications undertaken by business users. The series of modifications is called the change history of the business process model. In certain sectors, organizations must record all the change histories of their business process models for government auditing or for some legal requirements. Banks are examples to these organizations. According to the regulations made by the Banking Regulation and Supervision

Agency (BRSA), all banks are obliged to create and define their business processes and keep them up to date.

Organizations with hundreds of process models like banks often have difficulty in detecting needs of change and reflecting them to business process models. However, as the number of business processes increases, it is difficult for them to manage the process models by themselves and to find the required business process information effectively.

Several BPM tools to manage business processes allow people to design apply and monitor business processes. But, these tools usually put a strong emphasis on the graphical representation of processes. In other words, business processes are not part of the company knowledge but they are available only in the business process tool (if there are any) and/or in the documents [7].

The level of mechanization in BPM is still very limited, and does not provide a uniform representation of an organization's process space on a semantic level, that would be reachable by logical queries. In other words, organizations have incomplete knowledge of and delayed control over their process spaces [8].

In order to solve these problems, it is important to represent processes in a formal way. A body of formally represented knowledge is based on a conceptualization: the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold them [9]. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly. Ontology is an explicit specification of a conceptualization.

1.2 Motivation

Within the scope of the thesis, a bank sample was selected and presented in an ontological approach to produce the semantic and machine readable definition of the business processes of the bank.

The main problem is that the components and systems that make up the Bank's processes are not integrated. Systems are not "aware" of processes. Therefore, it is not possible to automatically determine the changes that caused the process update or what processes would be affected from these changes.

To solve this problem, a domain ontology was designed that incorporates extensive knowledge about the bank's processes, activities, departments, roles, application systems, relations between processes, inputs and outputs.

Contributions of this study to the bank's process domain are:

- Creating a machine-readable representation of processes,
- Allowing querying processes by logical expressions,
- Facilitating the determination of whether there is any process that will be affected by the changes via logical queries,
- Providing the opportunity to analyze and improve business processes during the implementation of the detected changes,
- Providing the management and orchestration of the processes,
- Facilitating tasks such as the exploration, documentation, and validation of new requirements.

1.3 Organization

The organization of the thesis in the following chapters is as follows:

Chapter 2 provides background information about ontologies. In addition, different perspectives, foundations of ontology development, and information about different ontology tools are provided.

Chapter 3 provides background information about Business Process Management, Business Process Modelling, Business Process Languages and Notations, Business Process Management Tools and Business Process Life-Cycle.

In Chapter 4, the implementation of the ontology is given and in chapter 5 it is evaluated by the competency questions. Finally, chapter 6 concludes the thesis and provides some possible future work.

CHAPTER 2

ONTOLOGIES

2.1 What Are Ontologies?

In philosophy, an ontology means an organized explanation of existence. In science literature, the most referenced definition of ontology is made by Grubber, in 1993, “An ontology is an explicit specification of a conceptualization” [10]. After the popularity of this definition, Borst [11] slightly altered Grubber's definition and in 1997 defined an ontology as a "formal specification of a shared conceptualization".

In 1998, Studer and his colleagues combined these two definitions as [12]: “An ontology is a formal, explicit specification of a shared conceptualization” and explained it as: “A *conceptualization* refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. *Explicit* means that the type of concepts used, and the constraints on their use are explicitly defined. *Formal* refers to the fact that the ontology should be machine readable. *Shared* reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.” [12]. In addition, the graphical representation of an ontology has been presented by Haller [13] as shown in Figure 2.1.

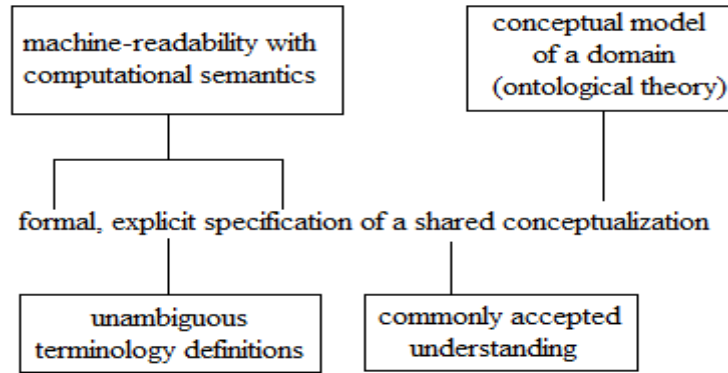


Figure 2.1 Graphical representation of ontology by Haller

Since Grubber, many definitions and studies have been conducted on what an ontology is. One of them was made by Guarino and Giaretta in 1995 [14] where various definitions, syntactic and semantic interpretations of ontology were compiled.

There are other definitions of ontologies as well, but these are based on the approach taken by authors to construct their ontologies. Reusability and redesign of ontologies are some of the approaches for ontology construction. As an example, the SENSUS project has defined ontologies as: “hierarchically structured set of terms, for describing a domain that can be used as a skeletal foundation for a knowledge base” [15].

One other example is the KACTUS project that defined an ontology as: “the means for describing explicitly the conceptualization behind the knowledge represented in a knowledge base” [16].

Except those, there are quite a number of ontology construction approaches which have been proposed. Some of these approaches are TOVE[17], Enterprise Ontology[18], METHONTOLOGY[19], ONTOLINGUA[20], ONIONS[21], etc. These approaches have been examined in a number of studies.[22] [23].

As presented above in this section, the literature provides a series of definitions of an ontology and these different definitions provide a different and complemented perspective of the same reality.

2.2 Why use ontologies?

Ontology explicitly expresses the concepts within a given area or domain, the characteristics of these concepts and their relations with each other, through a common understanding. Some of the reasons to develop an ontology are [24]:

- **Sharing common understanding:** For example, if different websites operating in the same service sector share the same ontology of the terms they use, computer systems can understand and process the information from these web sites to answer users' questions or use them as input data for other utilizations.
- **Reusing of domain knowledge:** If we develop a well-defined ontology, e.g. time ontology, people can reuse it for their own field instead of building from scratch. Also, if we want to construct a large scale ontology, we can combine other present ontologies that define parts of our large scale ontology.
- **Making domain assumptions explicit:** If we explicitly define our assumptions within the domain, these assumptions are easy to update in the situation of a knowledge change about the domain. Also, clear definitions of domain knowledge make it easy for new users to understand concepts and properties of the domain.
- **Separating the domain knowledge from the operational knowledge:** Domain knowledge is the knowledge people have in a specific area or subject. It can be about different areas or subjects. Operational knowledge is known as the knowledge used by people performing their daily activities. It is mainly based on the competences and experiences that people learn by doing.
- **Analyzing domain knowledge:** The main purpose of creating an ontology is not to create an ontology of the domain itself. Usually the aim is reuse, extend or update the existing ontologies. To be able to achieve this aim, analyzing the domain knowledge is very important. Once a declarative specification of the terms is available, other applications can use these ontologies and knowledge bases as input data for themselves.

2.3 Types and Categories of Ontologies

There are a number of methods used to build an ontology. Which method will be used depends on our specific objectives in building the ontology. There are three basic dimensions defined by Uschold [25] which indirectly lead to the type of ontology, which are:

- Formality
- Purpose
- Subject Matter

2.3.1 Formality

An ontology can be in a variety of forms, but formality level of the applied ontology depends on the formality to be used to encode terms and meanings. Uschold and Gruninger [26] defined four formality degrees: highly informal, semi-informal, semi-formal or rigorously formal ontologies which are defined below:

Highly-informal: Ontologies defined by loose expressions in natural language. e.g. Glossaries.

Semi-informal: Ontologies defined by limited and planned form of natural language. e.g. Enterprise Ontology [18].

Semi-formal: Ontologies defined by a formally defined language. e.g. ONTOLINGUA [20].

Rigorously-formal: Ontologies defined by well-defined terms, semantics, and properties, e.g. TOVE [17].

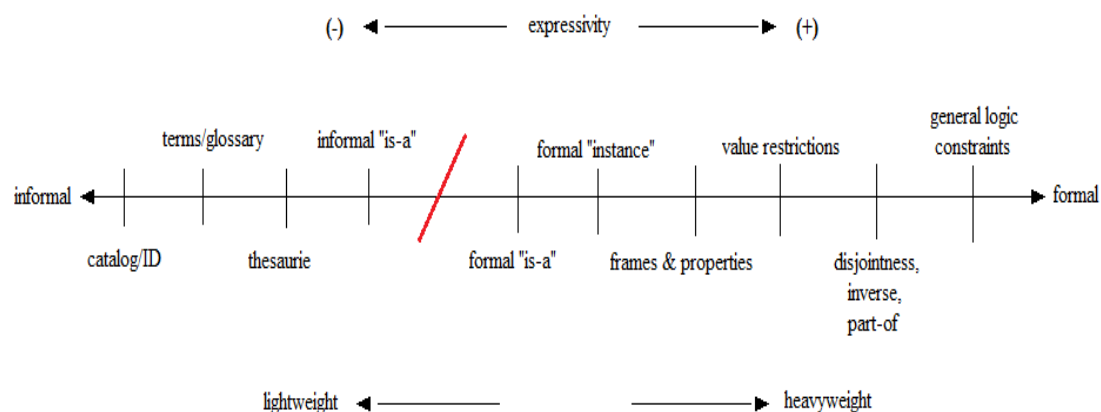


Figure 2.2 The ontology spectrum that defined by Lassila and McGuinness

Ontologies can also be classified according to their expressivity level. In Figure 2.2, the spectrum of ontologies according to expressivity is presented. Below are the classifications described in literature [27] [28].

The most primitive and simple form of ontologies are controlled vocabularies. A controlled vocabulary has a limited list of terms with a unique identifier. The most lightweight examples of controlled vocabularies are Catalogues. In *Catalog*, a term can be interpreted without any ambiguous interpretation by using the same identifier each time.

Other example of controlled vocabularies is Glossaries. A *glossary* consists of a list of terms and their meanings. Meanings of terms are defined by natural language statements. Users can learn more information by reading these natural language statements. However, these statements are not explicit and adequate for machines so enough inferences cannot be made by machines.

One other example of a controlled vocabulary is Thesauri. *Thesaurus* increase expressiveness of ontology by adding extra linguistic relations between terms (e.g. homonyms and synonyms) But, like glossaries, thesaurus do not define adequate explicit term hierarchy, either (e.g. sub-class, super-class).

Very few people see controlled vocabularies (Catalogs, Glossary, and Thesaurus) as ontologies. In order to view something as an ontology, many people think that it should be created in an explicit hierarchy.

Informal "is-a" defines an explicit subclass-super class hierarchy, but is not a strict hierarchy.

After this point on the ontology spectrum, more strictly defined ontologies start to appear. They are created by strict specialization–generalization relations. In hierarchy, more specific terms inherit their meaning from more general ones. First sample of this group is *formal "is-a"* relationships that include strict superclass-subclass relationships. If class B is a subclass of class A and class C is subclass of class B then class C is subclass of class A at the same time. Beyond one step of formal "is-a" relationships, there is a *formal instance* hierarchy. Similarly, in formal instance relationships, if class B is subclass of class A, instances of class B are instances of class A at the same time.

One step further on the spectrum, we encounter *frames* that are light heavyweight. Here, property information are added to the classes. They are inherited by subclasses and instances according to generalization and specification (super class-subclass hierarchy) relations of classes, or the decomposition of classes into subclasses.

An axiom is a set of statements, each of which must be true in the situation described. One example of axioms is value restrictions that are used to add restrictions on properties. *Value restrictions* define what values a property can take (e.g. exact numbers or defined range). One other example of axioms is *disjoint classes*. If two classes are defined as disjoint to each other, e.g. Male and Female, an instance of one of them cannot be an instance of the other one at the same time. The last point on the spectrum is *general logic constructions* that are other examples of axioms.

2.3.2 Purpose

In this section, the motivations of the use of ontologies are reviewed. Presently, there are rich definitions and uses for ontologies in literature. The most common source of motivation for ontology use is the reusability. Another important motivation for ontologies is to integrate different field models into a consistent framework, engineering and design. Some consider their ontologies as a means of structuring a

knowledge base. Based on these motivations, Uschold and Gruninger [26] identified three main categories of uses for ontologies, as seen Figure 2.3:

1. *Communication*: among people with different needs and perspectives. Ontologies that are informal but not ambiguous belong to this group.
2. *Inter-Operability (Cooperation)*: between systems that are obtained by translating between different modeling methods, paradigms, languages and software tools. Here, the ontology is used as root of data exchange.
3. *Systems engineering benefits*: by offering a conceptual description, the ontology can help in software systems' analysis and design.
 - Re-usability: The purpose of reusability is to reveal an ontology, which is preferably adapted to a large number of applications.
 - Knowledge acquisition: ontologies can be used for analysis of domains or knowledge base developments, so speed and reliability of knowledge acquisition can be increased.
 - Reliability: A formal representation can increase reliability of an application and systems by supplying automatic verification of consistency.
 - Specification: in the phase of defining and analysis of information systems' specification, ontologies can help.

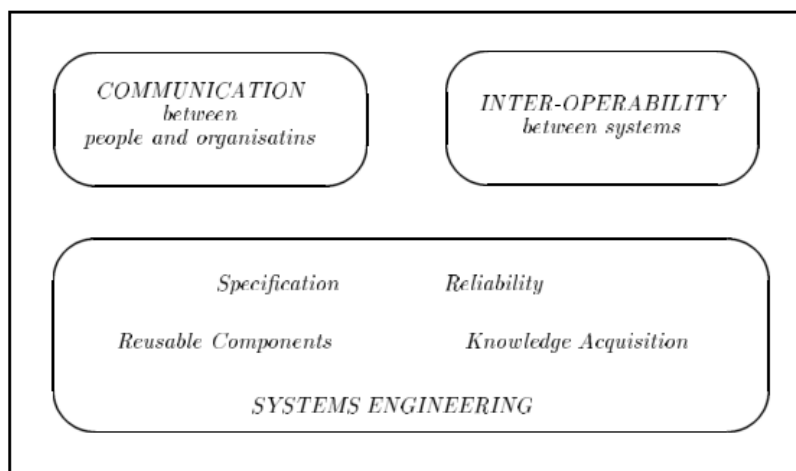


Figure 2.3 Uses for ontologies

2.3.3. Subject Matter

The subject matter that an ontology represents can be classified according to their level of context dependency.

Types of ontologies, according to their level of dependence on a particular task or point of view are given in Figure 2.4 [29]. Thick arrows represent specialization relationships.

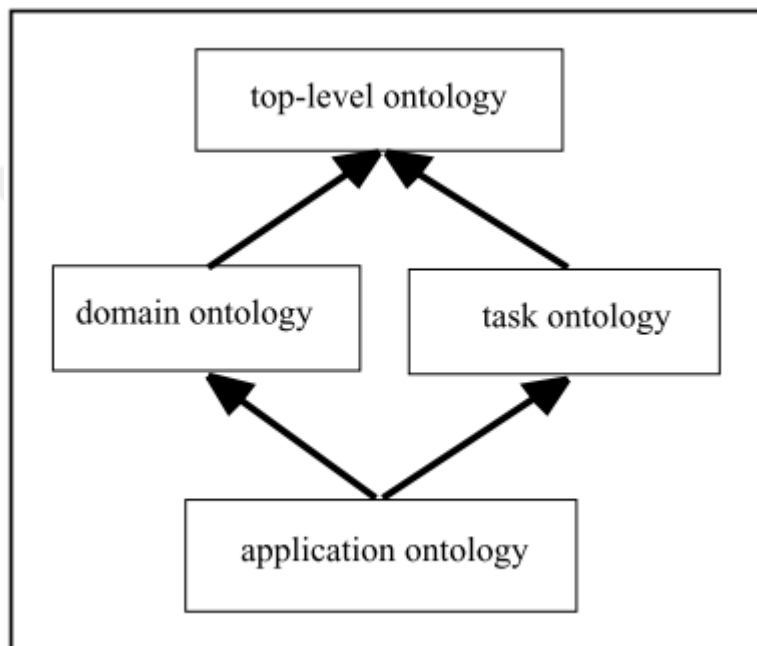


Figure 2.4 Types of ontologies

Top-level ontologies also known as *foundational ontologies* represent very general concepts which are independent of a special problem or domain. (e.g. space, time, matter, object, case, fact) Having integrated top-level ontologies can be suitable for a large group of users.

Domain and *task ontology* concepts are specializations of top-level ontology concepts that are used in a special domain (e.g. banking, medicine) or task/activity (e.g. accounting, diagnosing).

Application ontology concepts are specializations of both domain and task ontology concepts. (e.g. accounting in the banking industry)

When building a new ontology, there are a few fundamental questions to be clarified about the methodologies, tools and languages to be used in the development process [30]:

- What approaches and methodologies can be used to create an ontology? Will existing ontologies be used again or the ontology will be built from scratch? What activities are carried out when building ontologies with a methodology?
- What tools can be used when developing an ontology? Does the tool have an extraction motor? Does it have translator for different ontology languages or formats? Can the ontology work together with different applications?
- Which language(s) can be used to build an ontology and which tools support the language? Is it appropriate to exchange information between applications?

2.4 Ontology Development Methodology

To develop ontologies, there is no single right way or methodology. There are always appropriate alternatives to domain of discourse. The best solution relies on the application domain and the extensions that are foreseen. Development of ontology is a repetitious work. After the first model of ontology is created, it can be evaluated by using it in applications or by consulting with experts in the field. Ontology should consist of the concepts that are close to the objects and relationships in your domain of discourse. Most likely, these are nouns (objects) and verbs (relationships) in the sentences that define the domain of discourse [24]. As noted, there is no single correct way to develop an ontology. Different researchers have presented different approaches. Among them, seven basic steps have been suggested by McGuinness and Noy's [24] presented briefly as follows:

Step 1: Determine the domain and scope of the ontology

The basic questions that need to be answered when determining the boundaries of an ontology are as follows:

- What is the scope of the ontology?
- For what purpose will the ontology be used?

- What kind of questions should the ontology answer to?
- Who is going to use and keep the ontology?

Also, to determine the scope of the ontology, a list of questions known as competency questions that a knowledge base based on the ontology can be used. A detailed list of the competence questions of the ontology that has been prepared is included in the following sections.

Step 2: Reuse existing ontologies

As has been mentioned in the previous sections, the reuse of existing ontologies is one of the most important driving forces behind ontology studies. Many ontologies are currently available electronically and can be transferred to an ontology development environment. If there are no suitable ontologies, an ontology can be developed from scratch.

Step 3: List of the important terms

It is useful to write a list of terms to be included in the ontology as well as their properties. For example, some of the most important terms in the ontology that we have been developed in this study are: main process, sub-process, business unit, ownership, giving reference, responsible roles, etc.

Step 4: The classes and the class hierarchy

The next step is arranging the classes in a taxonomic hierarchy with super- and sub-relationships. There are a few possible approaches to develop the class hierarchy: top-down, bottom-up, middle-out.

Top-down:

In this approach, reusable and shareable top-level ontologies are created manually by domain experts. The process begins with identification of the most general concepts in the domain and then continues with specialization of the concepts as shown in Figure 2.5 [31]. As with all approaches, this approach has its pros and cons. It may

lead to the selection of arbitrary high-level categories that may cause risk on the stability of the model, while helping to better control the level of detail [26].

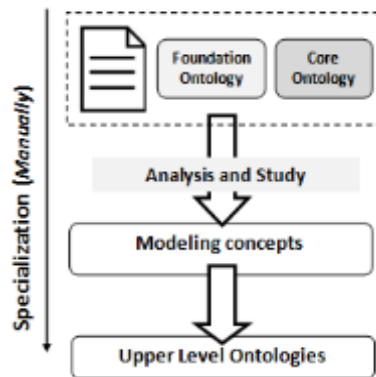


Figure 2.5 Top-down approach

Bottom-up:

In this approach, ontologies with high levels of detail are created by the semi-automatic analysis of domain related documents. The process begins with the identification of the most specific concepts in the domain and then continues with generalization of the concepts as shown in Figure 2.6 [31]. The disadvantage of this approach is that it is limited to the non-reusable domain or application ontologies, but it can help with the purification or expansion of existing ontologies by including new information emerging from the documents [26].

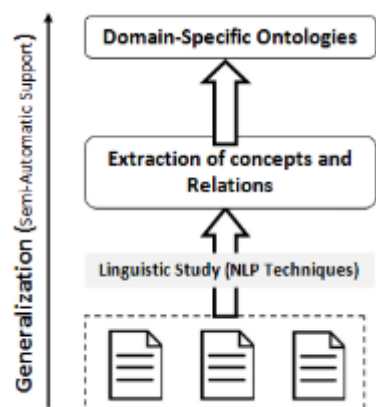


Figure 2.6 Bottom-up approach

Middle-out:

This approach is a combination of the top-down and bottom-up approaches. The process begins with the identification of the more prominent concepts first and then continues with generalizing and specializing them appropriately. [26]

There is no definite correct approach. Which of these three approaches to choose, depends to a large extent on the person's view of the domain.

Step 5: Define the properties of classes - Slots

Only the classes are not sufficient to answer the competency questions determined in the first step. After defining the classes, the internal structure of the concepts need to be defined. When the classes are subtracted from the list of terms defined in the second step, the rest are usually properties. Then, the classes these properties will define need to be specified. These properties become slots connected to classes.

Step 6: Define the facets of the slots

There may be some restrictions on properties that define slots. These restrictions can be about value type, allowed values, cardinality, domain-range, etc.

Step 7: Create instances

In the final step, the individuals belonging to the classes are created and their slots values are filled.

2.5 Ontology Languages

Ontology languages are formal languages used to create domain ontologies. A number of ontology languages have been developed and some of these have become ontology languages that are accepted in the context of the Semantic Web. Cocho and Gomez-Perez [32] classified the ontology languages into two main groups as shown

in Figure 2.7. The first group, called **traditional ontology languages**, consists of *first-order prediction logic* based languages, *frame-based* languages, and *description logic (DL)* based languages. The second group, called **web-based ontology languages**, is based on a combination of a particular Web standard and a specific traditional ontology language. The languages in the second group are used to help in the exchange of information between the Internet and ontology languages.

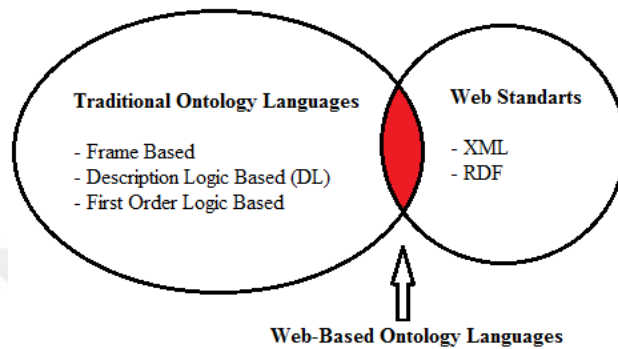


Figure 2.7 Classification of ontology languages

Frames are one of the knowledge representation approaches. They show similarity with class hierarchies in object-oriented languages but the difference is that frames focus on explicit and intuitive representation of knowledge.

Description logics (DL) are a family of formal knowledge representation languages. They model concepts, roles, individuals, and their relationships. The key difference between frames and DLs is axioms. While a frame notifies and completely defines a class, DLs use axioms, logical statements, to relate roles and/or concepts. Table 2.1 presents the classification of ontology languages according to structure or syntax [33].

Table 2.1 Classification of ontology languages

	Classification by structure			Classification by syntaxes	
	Frame based	Description logic-based	1st order logic-based	Traditional syntax ontology languages	Markup ontology languages
KIF			√	√	
LOOM				√	
OCML				√	
F-Logic	√			√	
SHOE					√
OML				√	
RDF(s)					√
OIL					√
DAML+OIL					√
OWL		√			√
CycL			√	√	

2.5.1. Traditional Ontology Languages

Knowledge Interchange Format (KIF): first order logic-based language which serves as a change format for different software.

Frame Logic (F-Logic): merges frames and first order logic by representing concepts, concept taxonomies, functions, and axioms.

Loom: is designed for general KBs, not for implementing ontologies. It includes DLs and production rules, and supply automatic classifications of concepts.

In Table 2.2 [33], comparison between ontology languages is given.

2.5.2. Web Standards and Recommendations

Extensible Markup Language (XML): is a World Wide Web Consortium (W3C) recommendation [34]. It is human readable, its syntax is well defined and easy to parse, but is not suitable for making semantic inference by machines.

Resource Description Framework (RDF): was developed by the W3C. It is written in Extensible Markup Language (XML) and provides a simple semantics for the data models [35]. It is developed to exchange machine-understandable information between applications not displayed to people. RDF statements occur as a triplet: resources (subjects), properties (predicates), statements (objects).

2.5.3. Web-Based Ontology Specification Languages

RDF schema (RDFS): has been built as a layer on the top of the basic RDF model. The RDFS can be seen as a set of ontological modelling primitives. There are three additional languages that have been developed as extensions to RDF(S) - i.e. OIL, DAML OIL and OWL.

Web Ontology Language (OWL): is a knowledge representation language standardized by the World Wide Web Consortium (W3C) which is an extension of RDF Schema as shown in Figure 2.8.

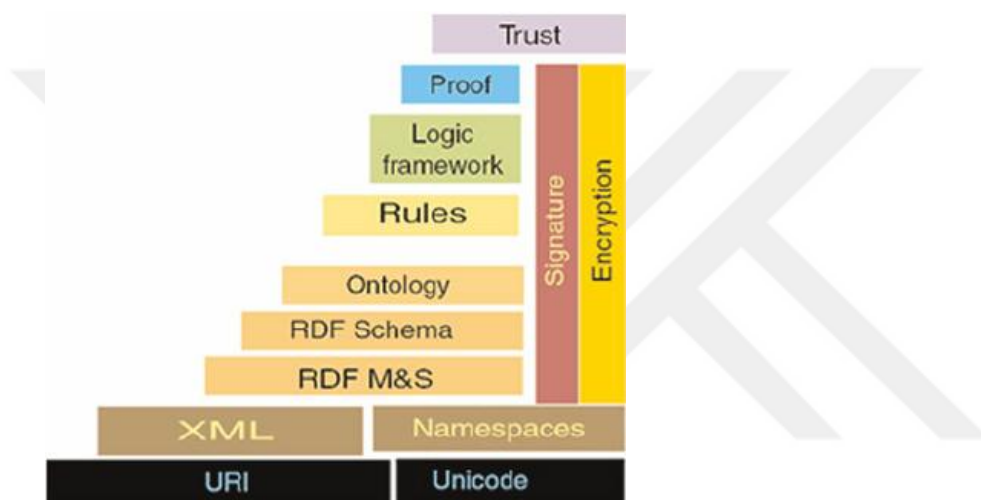


Figure 2.8 The Semantic Web Stack from Tim Berners-Lee

RDFS' expressivity that is limited to subclass and sub property hierarchy, domain and range restrictions and instances of classes is insufficient to describe resources to detail. The most important features to describe semantic knowledge of resources are missing [36]. They are:

- Placed range and domain constraints
- Disjointness of classes
- Cardinality constraints
- Boolean combinations of classes
- Special characteristics of properties: inverse, transitive, unique, symmetric.

Therefore, a stronger and richer ontology language was needed compared to the RDF scheme. OWL is the most important application of description logics (DL). The main building blocks of OWL and DLs are very similar. The basic difference is that concepts are named as classes and roles are named as properties [37].

OWL [38] provides a definition of three OWL variables with different expression levels as seen Figure 2.9.

- OWL Lite: is designed for users who need a classification hierarchy and simple constraints. It is a subset of OWL DL.
- OWL DL: is subset of OWL Full that designed to provide the highest possible expression by preserving the integrity of the calculation, decision-making and the availability of practical reasoning algorithms. It is a type of OWL language optimized for reasoning and information modeling.
- OWL Full: is a union of OWL syntax and RDF that is designed to use RDF's with maximum effectiveness and syntactic freedom. The biggest disadvantage of OWL Full is its inability to provide complete and effective reasoning.

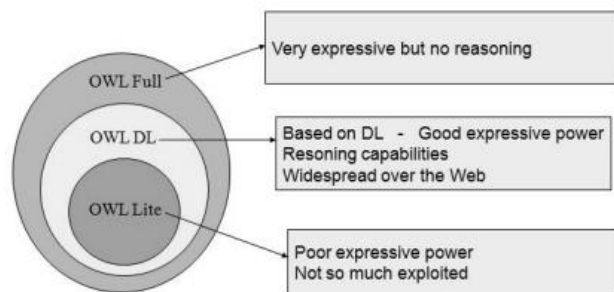


Figure 2.9 The three OWL sublanguages

OWL2 [39] on the other hand, adds new functionality with respect to OWL1 as presented below:

- property chains
- richer data types, data ranges
- qualified cardinality restrictions

- asymmetric, reflexive, and disjoint properties
- enhanced annotation capabilities

OWL2 also specifies three new profiles [40]: OWL2 EL, OWL2 QL, and OWL2 RL. Each of them lists different aspects of the expressive power of the OWL for distinct operational and / or application advantages.

- OWL2 EL: is especially helpful in applications where ontologies with a large number of properties and / or classes are used.
- OWL2 QL: is suitable for applications where a large instance of data is used and where query answering is the most important reasoning task.
- OWL2 RL: is intended for applications that take scalable reasoning without compromising too much expressive power.

The OWL2 is not a programming language, is a declarative, meaning it describes the domain of discourse logically. With the help of suitable tools (reasoners), we can learn information about this domain of discourse. Moreover, although OWL2 does store information and make databases, the OWL2 documentation is not a database framework [41].

Ontologies, i.e. OWL2, and databases share some features: for example, they both form a description of a specific part or section of the real world. However, there are some differences between these two systems. For example, while databases use the CWA (Close World Assumption), ontologies use the OWA (Open World Assumption). While databases use UNA (Unique Name Assumption), ontologies do not. While databases concentrate on data storage, ontologies are focused on adding meaning. While databases are being created from scratch, ontologies take attempt to advantage of using existing ontologies [42].

The components of ontologies include:

- Individuals: instances or objects of classes.
- Classes: concepts(terms) in the domain.
- Attributes: properties, features, characteristics, or parameters that objects (and classes) can have.

- Relations: relations between classes or between individuals.
- Restrictions: officially define that some claims should be true in the way they are accepted as input.
- Rules: if-then statements that express logical implications in a particular way.
- Axioms: logical claims (including rules) that constitute the general theory defined in the field of ontology.

Table 2.2 A Comparison between Ontology Languages

	LOOM	OCML	FLogic	SHOE	OML	XOL	RDF(s)	OIL	DAML + OIL	OWL	OWL2
General issues											
Concept partitions	√	X	X	X	√	X	X	√	√	X	√
documentation	√	√	X	√	√	√	√	√	√	√	√
Attributes											
Instance attributes	√	√	√	√	√	√	√	√	√	√	√
Class attributes	√	√	√	X	√	√	X	√	√	X	√
Local scope	√	√	√	√	√	√	√	√	√	√	√
Global scope	√	X	X	X	√	√	√	√	√	√	√
Facets											
Default slot value	√	√	√	X	X	√	X	X	X		
Type constraints	√	√	√	√	√	√	√	√	√	√	√
Cardinality constraints	√	√	X	X	X	X	X	√	√	√	√
Slot documentation	√	√	X	√	√	√	√	√	√	√	√
Taxonomies											
Subclass of	√	√	√	√	√	√	√	√	√	√	√
Disjoint Decomposition	√	√	X	X	X	X	X	√	√	X	√
Relations											
n-ary relations	√	√	X	√	√	X	X	X	X	X	√
Operational definition	√	√	√	X	X	X	X	X	X	√	√
Axioms											
1st order logic	√	√	√	X	√	X	X	X	X	√	√
2nd order logic	X	X	X	X	X	X	X	X	X	√	√
Named axioms	X	√	X	X	X	X	X	X	X	X	X
Embedded axioms	√	√	X	X	√	X	X	X	X	√	√
Instances											
Concept instances	√	√	√	√	√	√	√	√	√	√	√
Facts	X	X	X	X	√	X	X	X	X	√	√
Other											
Rules	√	√	X	√	√	X	X	X	X	√	√

2.6 Ontology Tools

In recent years, the number of environments and tools developed to create ontologies has increased considerably. Ontology tools allow users to visually design, build, examine, modify, and manage ontologies. Which tool is going to be select can be decided using different comparison criteria such as:

- Easiness

- Generality
- Expressiveness
- Complexity
- Documentation
- Scalability

The most preferred and widely used editors are Apollo, Protégé, TopBraid Composer, and Swoop. These tools are compared according to the following specifications and Table 2.3 shows comparison results of ontology tools [43].

- General description of the tools
- Software architecture and tool evolution
- Interoperability with other ontology development tools and languages
- Knowledge representation
- Inference services attached to the tool
- Usability

Apollo: has frame based knowledge representation model. The users can model the ontology with basic principles, e.g. Classes, instances, functions, relationships. It offers a strong type of consistency control. Although it is a user-friendly information modeling application, it does not support graphical view, web, information extraction, and multiuser capabilities or collaborative processing.

OntoStudio: is a knowledge model related to frame-based languages. It provides graphical tools for developing ontologies. With the help of client / server architecture where ontologies are managed on a central server, different users can reach and change these ontologies. Ontologies can be developed collaboratively. It supports various ontology languages for import and export.

TopBraid: Composer (FE): is ontology creation and transmission modeling tool based on Eclipse platform and Jena API which provides usability and extensibility. Its knowledge representation model is based on RDF, OWL and SWRL. It imports and exports from a range of data format including RDF(S), XML, Excel, etc. However, it does not support graphical view for users.

Protégé: is a free, open source ontology development tool and a knowledge management system developed at Stanford University. Protégé offers a plug-and-play environment that is Java-based, extensible and flexible, making it a flexible foundation for rapid prototyping and application development. Protégé fully supports the latest OWL2 Web Ontology Language and RDF features. It provides a graphical user interface to develop ontologies and offers a rapid prototyping environment where ontology developers can instantly create their own ontologies and try semantic constraints. It also contains deductive classifiers to verify that the models are consistent as well as to extract new information. It is the most popular and widely used ontology tool. It has more than 300,000 registered users and a large community support. There is even a forum dedicated to Protégé users. Additionally, there are also tutorial examples that provides guidance on ontology development [24][44].

Table 2.3 Comparison of ontology tools

Feature	Apollo	OntoStudio	Protégé	Swoop	TopBraid
<i>General description of the tools</i>					
Feature	Apollo	OntoStudio	Protégé	Swoop	TopBraid
Developers	KMI (Open University)	Ontoprise	SMI (Stanford University)	MND (University of Maryland)	Top Quadrant
Availability	Open source	Software license (Freeware)	Open source	Open source	Software license
<i>Software architecture and tool evolution</i>					
Feature	Apollo	OntoStudio	Protégé	Swoop	TopBraid
Semantic web Architecture	Standalone	Eclipse Client-server	Standalone and Client-server	Web-based and Client-server	Standalone Eclipse plug-in
Extensibility	Plug-ins	Plug-ins	Plug-ins	Plug-ins	Plug-ins
Backup management	No	No	No	No	No
Ontology storage	Files	DBMS	File and DBMS (JDBC)	As HTML Models	DBMS
<i>Knowledge representation and methodological support</i>					
Feature	Apollo	OntoStudio	Protégé	Swoop	TopBraid
KR paradigm of knowledge model	Frames (OKBC)	Frames, First Order logic	Frames, First Order logic, SWRL, Metaclasses	OWL	RDF, OWL, SWRL
Axiom language	Unrestricted	Yes (F-Logic)	Yes (PAL)	OWL-DL	OWL-DL
Methodological support	NO	Yes (Onto-Knowledge)	NO	NO	NO
<i>Interoperability</i>					
Feature	Apollo	OntoStudio	Protégé	Swoop	TopBraid
With other ontology tools	No	OntoAnnotate, OntoBroker, OntoMat, Semantic and Miner	PROMPT, OKBC, JESS, FaCT and Jena	No	Sesame, Jena and Allegro Graph
Imports from languages	Apollo Meta language	XML(S), OWL,RDF(S), UML Diagram, Database schemas (Oracle,MS -SQL, DB2,MySQL), Outlook, File system metadata, Remote OntoBroker	XML(S), RDF(S), OWL, (RDF,UML, XML)backend , Excel, BioPortal, DataMaster	OWL, XML, RDF, Text formats	RDFa, WOL, RDF(s), XH TML, SPIN, News Feed, RDF Files into a new TDB, Email, Excel

Table 2.3 Comparison of ontology tools (continued)

<i>Interoperability</i>					
Feature	Apollo	OntoStudio	Protégé	Swoop	TopBraid
Exports to languages	OCML, CLOS	XML(S), OWL, RDF(S), UML, OXML	XML(S), RDF(S), OWL, Clips, SWRL-IQ, Instance Selection, MetaAnalysis, OWLDoc, (RDF,UML, XML)backend	RDF(S), OIL, DAML	Merge/Convert RDF Graphs, RDF(S), WOL
<i>Inference services</i>					
Feature	Apollo	OntoStudio	Protégé	Swoop	TopBraid
Built-in inference engine	NO	Yes OntoBroker	Yes (PAL)	NO	WOL, SPARQL, Rule
Other attached inference engine	NO	NO	RACER, FACT, FACT++, F-logic, Pellet, Hermit	Pellet, RDF- like	OWLIM, Pellet, SPARQL Rules
Constraint / Consistency checking	Yes	Yes	Yes	Only with reasoner plug-in	Yes
Automatic classifications	No	No	No	No	No
Exception Handling	No	No	No	No	Yes
<i>Usability of tools</i>					
Feature	Apollo	OntoStudio	Protégé	Swoop	TopBraid
Graphical taxonomy	No	Yes	Yes	Yes	No
Graphical prunes (views)	No	Yes	Yes	No	No
Zooms	No	Yes	Yes	No	No
Collaborative working	No	Yes	Yes	Yes	Yes
Ontology libraries	Yes	Yes	Yes	No	Yes

2.7 Ontology Reasoners

A reasoner is a key component for working with OWL ontologies. A reasoner can be used to derive new facts from ontologies, because the knowledge in an ontology may not be explicit, and a reasoner will be required to extract implicit information in order to obtain accurate query results. There are many reasoners developed for this purpose (Pellet, RACER, FACT++, Snorocket, Hermit, CEL, ELK, SWRL-IQ, TrOWL, etc.).

Most of them can be used as a plug-in for Protégé. Cornet et al [45] examined the characteristics of the reasoners in three groups and Table 2.4 shows comparison results.

1. Reasoning characteristics: gives information about the basic properties of ontology reasoners. For example, which algorithm (*methodology*) is used by the reasoner, is it able to make all potential inferences (*soundness and completeness*), does it allow different types of axioms (*expressivity*), does it reuse the previous axioms with the updated ones (*incremental classification*), does it enable a combination of ontologies with rules (*rule support*), does it provide explanations for existing inconsistencies (*justification*), does it reasoning with individuals and contain instance checking (*ABox reasoning*) [46].
2. Practical usability characteristics: gives information about *availability* and *license* of the reasoners, and conformity of reasoners with OWL API, OWL Link API, and Jena Support, Protégé plug-in.
3. Performance indicators: performance of ontology reasoner is measured by computing the whole subsumption hierarchy of the ontology, classification, etc.

Table 2.4 Comparison of ontology reasoners

		Pellet	RACER	FACT++	Snorocket	SWRL-IO	HermiT	CEL	TrOWL	ELK
Methodology		Tableau based	Tableaux based	tableau based	Completion rules	SWRL rules	Hypertableu based	Completion rules	Completion rules	Consequence based
Soundness		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Completeness		Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Expressivity		SROIQ(D)	SHIQ	SROIQ(D)	EL+	-	SROIQ(D)	EL+	SROIQ	EL
Native Profile		DL, EL	DL	DL	EL	-	DL	EL	DL, EL	EL
Incremental Classification	Addition	Yes	No	No	Yes	Y/N	No	Yes	No	Yes
	Removal	Yes	No	No	No	Y/N	No	No	No	Yes
Rule Support		Yes (SWRL)	Yes (SWRL)	No	No	Yes (SWRL)	Yes (SWRL)	No	No	Yes (Own rule format)
Platforms		all	all	all	all	all	all	Linux	all	all
Justifications		Yes	Yes	No	No	Yes	No	Yes	No	No
ABOX Reasoning		Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No
OWL API		Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
OWL Link API		Yes	Yes	Yes	No	No	Yes	Yes	No	Y/N
Protégé Support		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
License		DULI: AGPL	own	GLGPL	own	Y/N	GLGPL	Apache License 2.0	DULI: AGPL	Apache License 2.0
Jena Support		Yes	No	No	No	No	No	No	Yes	Y/N
Impl. Language		Java	LISP	C++	Java	Prolog	Java	LISP	Java	Java
Availability		Open source	Commercial	Open Source	Commercial	Y/N	Open source	Open source	Commercial	Open source

Within the scope of this study, short explanations for Pellet, Fact ++ and Hermit are presented.

Pellet is an OWL2 reasoner which is Java based supporting OWL2 profiles. It is open-source and supports Jena and OWL API libraries. It is a tableaux-based reasoner.

FaCT++ is an OWL DL Reasoner developed in C++ at the University of Manchester. It supports OWL and OWL2 DL based ontology languages. It is open source and can be used with OWL API application. It implements tableaux based algorithms. FaCT++ is one of the default reasoners in the Protégé OWL editor.

HermiT is developed in University of Oxford, based on OWL2. Its key feature is that it is based on hyper tableau calculus. It can determine whether a given ontology is consistent or not by identifying relations between concept and other features much faster than other reasoners.

CHAPTER 3

BUSINESS PROCESSES

3.1 What is Business Process?

Business processes are main elements in modern companies. A business process is a combination of interdependent tasks that aim to solve a particular problem. It can be represented by a workflow model which is a graphical representation of the correct task sequence that leads to the achievement of the goal. A workflow model consists of the basic notations that are processes, actors, tasks, control flow and data flow. The executions of tasks is typically limited by the rules of dependence between tasks, such as sequence constraints, branching and merge rules, pre and post conditions, event management points, etc.

A process can consist of several sub-processes that have their own properties, or it can be a sub-process that contributes to achieving the purpose of the super-process. [47]

Business processes can be examined under three groups [48]:

- *Person-to-Person (P2P)*: Processes which mainly consist of activities requiring human intervention are in this group. Generally in these processes, e-mail, video conferencing tools, project tracking servers, social networks are used as software tools.
- *Application-to-Application (A2A)*: processes which only consist of activities requiring software systems intervention are in this group. Generally in these processes, transaction processing systems, enterprise application integration.

(EAI) platforms, and Web-based integration servers are used as software tools.

- *Person-to-Application (P2A)*: processes which consist of activities requiring both human intervention and applications which act without human intervention are in this group. Most of the BPM/WFM systems are in this group. The purpose of most information systems is to bring people and practices together to work as a whole. The boundaries between P2P, P2A and A2A are not sharp and there is a wide scale from P2P to A2A.

3.2 What is Business Process Management?

BPM stems from the restrictions originated by the Workflow Management System (WFMS), that generally focus on the realization of processes by automated systems that ignore the sustained adaptation and development of existing processes [49]. BPM accepts and supports the entire life cycle of business processes, including the realization of transactions with automated systems.

Van Der Aalst [5] has gathered the domain of the BPM activities around six main items: process modeling languages, process enactment infrastructures, process model analysis, process mining, process flexibility, and process reuse.

Process Modeling Languages: In business process management, modeling and analysis of processes play key roles. Once the processes are modeled, they become ready for analysis. This is why the choice of language to represent the processes of an organization is very important and there are many studies on which language should be used.

Process Enactment Infrastructures: Process models obtained by modeling can be used for enactment, for example, to execute a process model through a BPM system for control the workflow.

Process Model Analysis: Two main ways to model-based analysis are verification and performance analysis. While performance analysis aims to improve processes in terms of time, cost or quality, verification relates to the logical accuracy of a system or process, or to the comparison of two models. Both types of analysis seriously trust

on the existence of high-quality models. However, there is a problem of a lack of harmony between these handmade models and reality.

Process Mining: Problem of lack of harmony between handmade models and reality is aimed to solve by process mining. Process mining not only uses process models but also event data about the process. Thus, a direct link between the models and the actual low-level event data about the process are created.

Process Flexibility: There are many factors that require updating and adaptation of business processes. New laws, regulations and business rules, new systems, changes in business strategy are some examples to these factors. Achieving coverage of such changes is called process flexibility and is one of the main concerns of BPM.

Process Reuse: Organizations that have a large number of business process models face difficulties in the maintenance and management of these processes. Some models may be older, some may be combined due to their similarities, and some may need to be divided. The most common process reuse activities are about repositories, activity search, activity merge, activity convert, activity unify, and model similarity search.

3.3 Business Process Models

Business Process Management (BPM) aims to support “business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information” [50].

The concept of a process model is the basis for BPM. Process models can be expressed in different ways. There are many notation languages to model business processes. E.g. Petri nets, BPMN, UML and EPCs.

What is required to create a comprehensive process model? In response to this question, Van Der Aalst has defined four necessary perspectives [5].

- *The control-flow perspective:* order of activities is modelled.
- *The data perspective:* decisions, data creation, forms, etc. are modelled.

- *The time perspective:* durations, deadlines, etc. are modelled.
- *The function perspective:* activities and related applications are modelled.

Although the control-flow perspective is the basis of the process modeling, all perspectives must be included in the design of a comprehensive process model.

3.4 Business Process Languages

Much research and discussion have been conducted on how to model data. Even though there are diverse languages and Database Management (DBM) systems, the basic concepts of the information-centered view of information systems have been decided long ago.

On the other hand, there is no complete consensus on the fundamental concepts of process-centric view on information systems. Consequently, nowadays there are many systems and languages to model business processes such as BPMN, BPEL, UML, EPCs.

Organization's processes and the purpose of modeling these processes are very important in determining the language to be used. Van Der Aalst [5] identified three classes of languages:

- *Formal languages:* are based on theoretical formalities that provide clear semantics to allow for the identification and analysis of business process models (e.g. Petri Nets).
- *Conceptual languages:* don not have solid semantics as much as formal languages. They do not have well-defined semantics; as such, they are informal. Due to some blurring in modeling, they are not suitable for analysis or direct execution, but provide powerful graphical notations; which is the reason why business analysts prefer them in the first phase of process modeling. BPMN, EPC, and UML are in this class.
- *Execution languages:* contain more technical languages related to business process execution such as BPEL which are not within the scope of this study.

A Petri Net: [51] is a specific two-sided oriented graph type that is populated by three object types. These objects are places, transitions, and directed arcs. Directed arcs connect places to transitions or vice versa. No arc can connect a place to another place or a transition to another transition. Places that are designed by circles symbolize the possible conditions, states of the system. Transitions designated by rectangles symbolize the events, actions that may occur or change states. Places can include tokens that can be carried to other places by performing their operations. If there is a token in a place, it means that the corresponding condition is met or a resource exists. Specific features that are shown by an event-oriented dynamic system, like concurrency, decision making, synchronization and priorities, can be modeled by Petri nets in a satisfactory way.

Event Driven Process Chain (EPC): Simplicity and easy-to-understand notation is the major strength of EPC. Sequence of steps of a business process are represented by a connected graph that has events, functions and logical connectors (AND/ OR/ XOR). Events are passive elements, describing the conditions under which a function or a process works, or which function or process results. Functions are active elements. They model tasks or activities in a business process. Functions describe the transformations from the beginning state to the final state. Functions and events are connected through links (arrows) and logical connectors. Description of the process is completed by extra information, such as document, role, process owner, and organization unit [51].

UML Activity Diagram (AD): is a graphical representation of workflows of step-by-step activities and actions that support choice, repetition, and concurrency. Activity diagrams are composed of a limited number of figures connected to the arrows [51]. Arrows run from start to finish and indicate the order in which the activities occur. Such diagrams have the advantage that they can be understood by the software engineers liable for applying the business support components. Yet, its proximity to computational languages makes it hard to be used by business analysts with no knowledge in software design.

Business Process Modeling Notation (BPMN): is the state-of-the-art graphic language used to create business process schemes based on a flowchart technique

that is adapted to compose graphical models of business process operations [52]. Supplying a simple and intuitive notation that is easily understandable by business users is the main purpose of BPMN [1]. Initial drafts of processes can easily be drawn by business analysts and, these processes can be easily managed and monitored by business people. It is a combination of the precision of a well-defined notation system and its ease of use, which is responsible for the tremendous success of BPMN and makes it an international standard for modeling business processes. BPMN language is preferred and used in modeling of business processes because of being the latest technology language, wide-use, and ease of use. Therefore, while the business processes of the Bank in our study are modeled, BPMN notations are taken as a basis.

A business process is designed in a Business Process Diagram (BPD) in BPMN with a flowchart-based representation that provides a clear demonstration of activities, control flows, data, and helpful information about the process. The BPMN graphic elements that make up a BPD can be arranged in two groups: BPD Core Element Set and BPD Extended Set. The BPD Core Element Set consists of the key elements of the language that are appropriate to sufficiently represent the key features of most business processes. The BPD Extended Set consists of supplementary graphical elements and non-graphical attributes that allow more complex and detailed modeling to empower the core elements set.

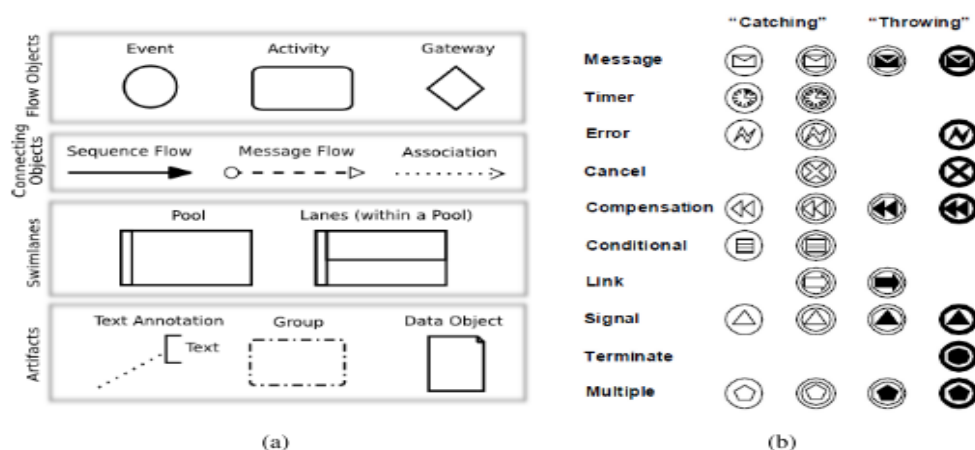


Figure 3.1 Basic BPMN notations

Figure 3.1 (a) shows BPD core element set flow objects:

Event: something that happens

Activity: work to be performed

Gateway: control flow elements e.g. branching, merging, etc.

1. Connecting Objects:

Sequence Flow: the order in which activities are performed

Message Flow: the flow of messages between businesses

Association: additional information attached to flow objects

2. Swimlanes:

Pool: describing participants in a process

Lane: organizing and categorizing activities

3. Artifacts:

Data Object: representing data processed or produced by activities

Group: informal grouping of activities within the same category

Text Annotation: illustrative textual notes

Figure 3.1 (b) shows an excerpt of the BPD extended set (the sub-types of Event element) available in BPMN.

3.5 Business Process Modeling Tools

In the design phase, to graphically represent the business process models, business analysts need a business process modeling tool. Actually, not only in the design phase, but also in modeling and analysis phases, a process modeling tool can be utilized. In light of this information, process modeling tools can be compiled into three groups:

- Tools which have a graphical interface to show sequence of activities.
- Tools that aim to describe an organizational model to specify resources such as roles, groups, etc.
- Tools that supply simulation or verification of workflows for analysis.

There are many tools that serve these purposes. For example, Eclipse, BPMN2 Modeler, Camunda, Activiti, Maestro, and Adonis CE. All these tools support BPMN 2.0 notations as well.

3.6 Business Process Management Life Cycle

Organizations design, implement and use business processes to achieve their goals and objectives. BPM manages not only automated but also non-automated business processes to obtain results that are suitable to an organization's strategic goals. To identify, implement, execute, monitor, measure and document processes, BPM describes a systematic approach. In other words, BPM is a systematic approach to improve business processes, make workflows more efficient and advance the ability to adapt to a constantly changing business environment.

Four key activities of BPM are:

- Model: creating a process model to be used for analysis or enactment.
- Enact: exercising a process model to check and verify the concrete event.
- Analyze: analyzing a process model using event records.
- Manage: all other activities, for instance, regulating the process, reallocating resources, or administering large collections of related process models, documenting.

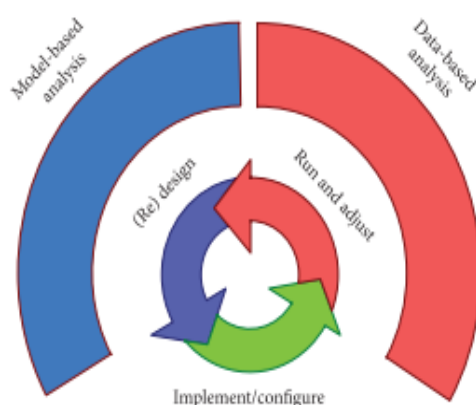


Figure 3.2 The BPM life cycle

Figure 3.2 shows Van Der Aalst's BPM life cycle [5] consisting of three phases: (re)design, implement/configure, and run and adjust.

After a process model is designed in the *(re)design phase*, it is passed to *implementation/configuration phase* where it is converted into a working system. When a BPM system already exists and the model is in executable form, this phase may take a very short time, otherwise it takes quite a long time. The final phase is *the run and adjust phase* where processes are enacted and adjusted if necessary. The process is not redesigned at this phase, only pre-defined controls are used. There are two types of analysis.

- In *model-based analysis*, the simulations of process models or “what-if” scenarios are used to test correctness of the model.
- In *data-based analysis*, event data are gathered through the running system to discover if there is congestion, waste, and deflection.

After the last phase is finished, the cycle can be repeated since BPM is an iterative approach.

3.7 Ontologies and Business Process Management

Many Business Process Management (BPM) research was done that resulted in methods, techniques and tools to support the design, implementation, management and analysis of operational business processes.

The common study subject of BPM researchers are process modeling languages, process enactment infrastructures, process model analysis, process mining, process flexibility, and process reuse [5]. The most common process reuse activities are about repositories, activity search, activity merge, activity convert, activity unify, and model similarity search.

One of the general problems affecting all BPM activities is the use of informal text. For example, the same context may be tagged differently from one process to another such as "life insurance" and "life assurance". The best way to solve these and similar problems is the use of a common ontology [8].

Adding semantic annotations to business process models is not a new idea. There are studies that have been suggested by a few authors for different process languages and objectives. These studies are divided into two groups. In the first group, there are studies that add semantic annotations to indicate the dynamic behavior of a business process [53],[54]. The second group includes studies that add semantic annotations to indicate the meaning of process elements [55] [56].

Pereira and Markovic [53] suggest an effective formalism to define business process models to backup the reuse of existing process pieces during modeling. They support not only dynamic but also static aspects of a process definition. To support dynamic aspects of a process definition they use process control flow algebra, the π -calculus, thus catching the behavior of the process. Furthermore, they support static aspects of a process definition by defining organizational and informational perspectives. E.g. input/output data, organizational roles, etc. To support both aspects they suggest Business Process Ontology (BPO) described using WSML (Web Service Modeling Language) ontology representation language.

Smith and Proietti [54] suggest a rule-based framework for representing and reasoning about business processes from both the behavioral and semantic point of views. For this purpose, they use standards for business process modeling like BPMN and ontology definition such as OWL-RL rule-based ontology language. They use an approach similar to Fluent Calculus for modeling the behavior of individual process elements in terms of preconditions under which a flow element can be executed and its effects after execution.

Ehring et al. [55] offer a semi-automatic approach for measuring the similarity between semantic business process models. To achieve this, they have built an OWL-DL based description of Petri nets. They use three ways to measure similarities. In *syntactic measurement*, the numbers of common characters in element names are compared. In *linguistic measurement*, the synonyms of the element names are determined based on the WordNet dictionary. In *structural measure*, they use concept instances to detect homonyms. In the last step, the composite similarity measure is calculated by taking the average of the three similarity measurements.

Born et al. [56] suggest a tool for adding semantic annotation to business process models. The tool is an extension of business process modeling tool Maestro that supports BPMN notations. They intend to match elements of a graphical business process model with concepts of domain ontologies. To achieve this, they use;

- Knowledge about domain concepts, objects, actions, states from the ontology
- Structural knowledge from the process graphs
- Matching methods considering synonyms and homonyms

Finally, there is the Financial Industry Business Ontology (FIBO) [57] that has been developed by Petrova et al. Here, they consider an approach to a formalized description and meaning harmonization for financial terms and means of semantic modeling. Ontologies for the semantic models are presented with special languages developed for the Semantic Web such as RDFS, OWL.

CHAPTER 4

ANALYSIS, DESIGN AND IMPLEMENTATION OF THE SYSTEM

As mentioned in Section 2.4, there is no single right way or method to develop an ontology, There are always appropriate alternatives for the domain of discourse. The best solution relies on the application that you have in mind and the extensions that you foresee. Development of an ontology is repetitious work. After the first model of the ontology is created, it can be evaluated by using different techniques. The main point is to make a decision by focusing on the purpose of the ontology by taking into account the domain data for a descriptive and meaningful ontology structure.

In this thesis, to decide on the ontology design, firstly the domain data has been analyzed. Then, according to the findings of the analysis; entities, relations, data properties are decided and the ontology is developed. Finally, the resulting ontology is tested with the help of competency questions that are presented in the evaluation chapter.

4.1 Analysis of Domain

According to the regulations of the Banking Regulation and Supervision Agency (BRSA), all banks in the Turkish banking system are obliged to create work flow charts showing the controls and work steps of the banks' business processes.

As per the Regulation on Internal Systems and Internal Capital Adequacy Assessment Process of Banks dated 11.07.2014 and numbered 29057, all banks are obliged to create and keep up to date their business process flow.

As per the Regulation on the Supervision of Banking Information Systems and Banking Processes to be held by the Independent Audit Firms dated 13.01.2010 and numbered 27461, the banks' business processes are audited once a year.

Within the scope of the thesis, a bank sample was selected based on the business processes of Vakıflar Bankası T.A.O. Pertaining to the aforementioned regulations, flow charts of all business processes of the selected bank were created and documented in 2012. There are a total of 621 business processes that are in the process status, cancelled or in force. Two process documents are created for each of business processes.

The first one is the Excel document with all the details of the process. The document consists of 3 worksheets where the first worksheet contains general information about the process, as shown in Figure 4.1. This information is as follows: name and index number of the process, the purpose of the process, the owner business unit of the process (directorate and presidency), process authority person, the date of creation, the date of revision, the date of cancellation, the version number, list of reference documents, definitions and abbreviations.

VakıfBank		SÜREÇ GENEL BİLGİLERİ	
Süreç Adı		Süreçin Amacı	
Toplu Mevduat Hesabı Açma Süreci		Toplu mevduat hesabı açma işlemlerinin yürütülmesi amacıyla süreç oluşturulmuştur.	
Süreç Sahibi İş Birimi Onay		Süreç Yetkilisi	
Kredi Müsteri ve Hesap Operasyonları Başkanlığı Erol TUNCAY		Tanımlar ve Kısaltmalar	
Müşteri ve Hesap Operasyonları Müdürlüğü Ayça MELİKOĞLU		ETS: Evrak Takip Sistemi	
Referans Dokümanlar		Süreç Oluşturma Tarihi	
Belge Adı	Belge Tarihi	27.01.2011	
		Süreç Revizyon Tarihi	
		7.03.2017	
		Revizyon No	
		003	
		Yönetim Kurulu Onay Bilgileri	
		Toplantı No:	
		Tarih:	
		Karar No:	

Figure 4.1 Process overview

The second worksheet contains more detailed information about the process, as shown in Figure 4.2. This information is as follows: activity number, decision step, activity, activity owner business units, responsible roles, application tools used,

reference documents (laws, notifications, regulations, contracts, forms, etc.), checkpoints, and extra explanations.

01.Mevduat Süreci 01.01.Hesap İşlemleri Süreci 01.01.12.Toplu Mevduat Hesabı Açma Süreci										Revizyon No:003		
No	Karar Adımı	Aktivite	Karar	Aktivite Sahibi	Sorumlu	Aktivite Sıklığı	Kullanılan Uygulama / İletişim Aracı	Referans Doküman	İlgili Belge/ Doküman	Kontrol Tipi (K)	Kontrol Açıklaması	Açıklama
1	Başlangıç	Kurumdan toplu mevduat hesabı açma talebi ve varsa ürün talebi alınır.		Şube	Satış Grubu Personeli	Günde 12 İşlem	E-posta	2016/504 Sayılı Tebliğ		Manuel	Beyazıt Ürün ve Hizmet Paket Sözleşmesinin talep edilen ürüne göre ilgili sayfaları müşteriye verilir.	
2		Kurum talepli Müşteri ve Hesap Operasyonları Müdürlüğüne iletilir.	3. ve 4. İşlem Adımı	Şube	Satış Grubu Personeli Operasyon Grubu Personeli		ETS		ETS'de Yer Alan Toplu Hesap Açma ve Kredilendirme Şablonu, Ürün Tanımlaması Yapılacak Kişilere Ait Datamın Bulunduğu Dosya			Kurum çalışanlarına hesap açılacak talep edilen diğer ürünlerin de belirtildiği ürün talep yazısı ve kurumdan alınan veri ETS üzerinden gönderilir. Şube tarafından istenirse müşteri numaraları tek tek verilerek Müşteri ve Hesap Operasyonları Müdürlüğüne bildirir ya da müşteri numaraları toplu olarak Müşteri ve Hesap Operasyonları Müdürlüğü tarafından verilir.
3		Talep incelenerek başka ürün talep edilip edilmediğine bakılır.	5. veya 10. İşlem Adımı	Müşteri ve Hesap Operasyonları Müdürlüğü	İlgili Personel				Talep Yazısı			
4		Şubeden gelen data uygun formata getirilir ve varsa diğer ürün talepleri gelecek sisteme aktarılır.		Müşteri ve Hesap Operasyonları Müdürlüğü	İlgili Personel		VIT-CRM-Bayvurular-Liste Sorgu-Toplu Data Talebi Dosyası		Müşterinin Ait Bilgilerin Yer aldığı Data Dosyası	Manuel	İşlem bir başka kullanıcı tarafından kontrol edilerek onaylanır.	Şubeden gönderilen dosya ile hazırlanan data hesap açma yapısına uygun olacak formata getirilir.
5		Sistem tarafından müşteri bilgileri kontrol edilir.	6. veya 7. İşlem Adımı	Müşteri ve Hesap Operasyonları Müdürlüğü						Sistem	Müşterinin mevduat hesabı açılmasına engel bir durumu olup olmadığına bakılır.	Ürün servislerinden gelen sistemsel hatalara ve müşterinin mevduat hesabı açılmasına engel bir durumu olup olmadığına bakılır.
6	Mevduat Hesabı Açılmasına Engel Durum Var İse	Sistem tarafından hesap açılmaz.	8. İşlem Adımı	Müşteri ve Hesap Operasyonları Müdürlüğü			VIT-CRM-Bayvurular-Liste Sorgu-Toplu Data Talebi Dosyası					Sistemden kaynaklı olarak ürün servislerinden yanıt alınmaması, müşterinin kimliğinin kayıp patenti listesinde yer alması veya müşterinin düzgün durumunda olması gibi hesap açılmasına engel olan nedenlere bakılarak işlem müşteri için sonlandırılır ya da devam ettirilir.
7	Mevduat Hesabı Açılmasına Engel Durum Yok İse	Sistem tarafından hesaplar açılır.		Müşteri ve Hesap Operasyonları Müdürlüğü						Sistem	Mevduat hesap numaraları manuel müdahale olmaksızın ilgili pozisyondan iletilir.	Sistem data pozisyonu ile ilgili müşteri numaralarına bağlı olarak mevduat hesaplarını açar.
8		Açılan ve açılmayan hesap numaralarına ilişkin bilgiler ekranda görüntülenir ve dosya PC'ye kaydedilerek Şubeye e-posta ile gönderilir.	Son	Müşteri ve Hesap Operasyonları Müdürlüğü	İlgili Personel		VIT-CRM-Bayvurular-Liste Sorgu-Toplu Data Talebi Dosyası					VIT grid ekranına bilgilerin kimlik, müşteri, hesap numarası ve verimleşme diğer için bilgileri girilerek Oku komutuyla ekrandaki bu bilgiler excel formatı seçilerek PC'ye kaydedilir.
9	Başka Ürün Talep Ediliyor İse	Bkz.01.01.13 Toplu Kredi Kartı Verme, Kredi Kartı Baskı, Yatırım Hesabı, Havuz Hesabı, Döviz Hesabı Açma Süreci	Son	Müşteri ve Hesap Operasyonları Müdürlüğü	İlgili Personel							
10	Başka Ürün Talep Edilmiyor İse	Gerçekleştirilen ve gerçekleştirilmeyen işlemlere ilişkin Şubeye bilgi verilir.	Son	Müşteri ve Hesap Operasyonları Müdürlüğü	İlgili Personel		E-posta					

Figure 4.2 Process details

The final worksheet contains detailed revision information about the process, as shown in Figure 4.3. This information is as follows: revision number, revision date, the changes that have been made, reference document.

REVİZYON TAKİPİ			
Revizyon No	Revizyon Tarihi	Yapılan Değişiklikler	Referans Doküman*
001	9.11.2012	27.01.2011 tarihli "01.01. Toplu Hesap Açma ve Müşteri Numarası Verme Süreci" indeks numarası ve adı değiştirilerek güncellenmiştir.	2012/2711 sayılı tebliğ
002	27.05.2013	Toplu müşteri numarası verme adımları süreç içerisinde çıkarılarak "01.01.01.Müşteri Numarası Verme Süreci"ne eklenmiştir.	
003	7.03.2017	02.05.2016 tarih 14 sayılı "Bankacılık Operasyonları Başkanlığı Yapılanması" Genelgesi kapsamında süreç revize edilerek güncellenmiştir.	

*Süreçte yer alan revizyonun Banka teşkilatına duyurulduğu Referans Doküman'ı (Genelge/Tebliğ/Duyuru) ifade eder.

Figure 4.3 Process revision tracking

The second document created is the process flow chart. To draw the process flow charts, Microsoft Office Visio which supports BPMN notations is used as a graphical interface. Detailed information about BPMN is given in Section 3.4 and the explanation of the BPMN notations is shown in Figure 3.1. An example of the process flow chart drawn with Microsoft Office Visio is shown in Figure 4.4.

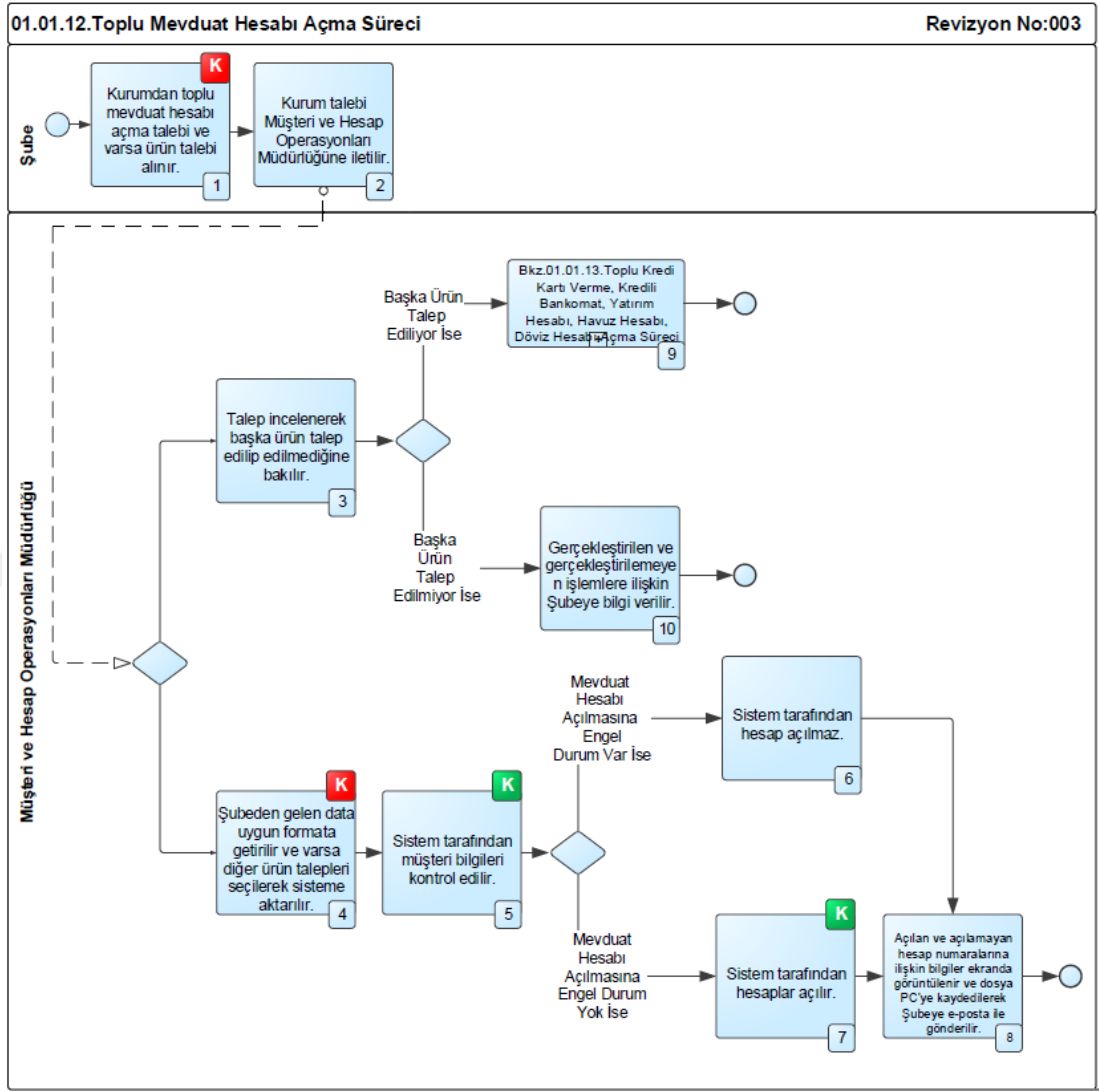


Figure 4.4 Process flow chart

Design of flow charts is kept simple to be easy to follow and understand the process. The most basic information such as activity owner, control points and the next activity step are shown on the flow chart, but information about responsible roles, reference processes, reference documents, and used application tools are not presented. As mentioned before, these details are kept in the Excel documents. This situation makes it difficult to perform automatic translation of the flow charts to an ontology that covers all situations which cause process update.

By its very nature, the banking system is one of the sectors where change and development are most frequent. In a sector where change is that fast, keeping all business processes up-to-date is not an easy task.

The main reasons that can cause change in business processes are:

- Changes, in organizational structure,
- Changes, in laws, regulations and notifications,
- Changes, in technology and systems used,
- Modifications and improvements in the operation of processes,
- Newly created or cancelled processes.

For example, changes in organizational structure lead to changes in process ownership, or changes in laws and regulations may result in updating of reference documents of the processes. Also, technological changes may require updating of the application tools used in the processes. In another example, if there are processes that give reference to a process that has been canceled, these processes need to be updated.

In the current structure, whether or not there is a change in the processes is tracked through two different channels. The first of these is the process owners that have the authority to put rules on the processes. Process owners are the main unit responsible for reporting changes in processes. The second is the process coordinator, who is responsible for reflecting the changes requested by the process owners to the process documentation and reporting the updated processes to the bank organization.

The main problem is that the components and systems that make up the processes are not integrated. Therefore, it is not possible to automatically determine the changes that caused the process update or what processes would be affected from these changes.

To solve this problem, a new ontology was designed by analyzing all Excel documents that belongs to processes and the resources that could cause changes in operations. With the help of the queries that can be made on the ontology, it is aimed to facilitate to determine whether there is any process that will be affected by the changes.

4.2 Implementation of the Ontology

In implementation phase, Protégé was chosen as the ontology development tool. Protégé provides a graphical user interface to develop ontologies and offers a rapid prototyping environment where ontology developers can instantly create their own ontologies and try semantic constraints.

Protégé also contains deductive classifiers to verify that the models are consistent and to extract new information. Detailed information about this tool was given in Section 3.5

Ontologies are used to capture information about some domain of interest. An ontology explains the concepts in the domain and also the relationships between these concepts. The latest development in standard ontology languages is the OWL from the World Wide Web Consortium (W3C). Three components of OWL Ontologies are: *individuals*, *properties*, *classes*.

Individuals also known as instances represent objects in the domain of interest. Individuals can be referred to as being ‘instances of classes’ [44].

Properties are binary relations on individuals - i.e. properties that link two individuals together.

Classes are interpreted as sets that contain individuals. Classes may be organized into a superclass-subclass hierarchy, which is also known as taxonomy. Subclasses specialize their super classes.

4.2.1. Classes

As shown Figure 4.5, the processes in the bank system were classified under nine main processes that are subclass of *IsSurecleri* class and each of them are defined as disjoint classes:

- 1- Deposit process
- 2- Credit process
- 3- Accounting process
- 4- Alternative distribution channels process
- 5- Credit cards process

- 6- Financial reporting process
- 7- Payment systems process
- 8- Treasury / securities and fund management process
- 9- Support process

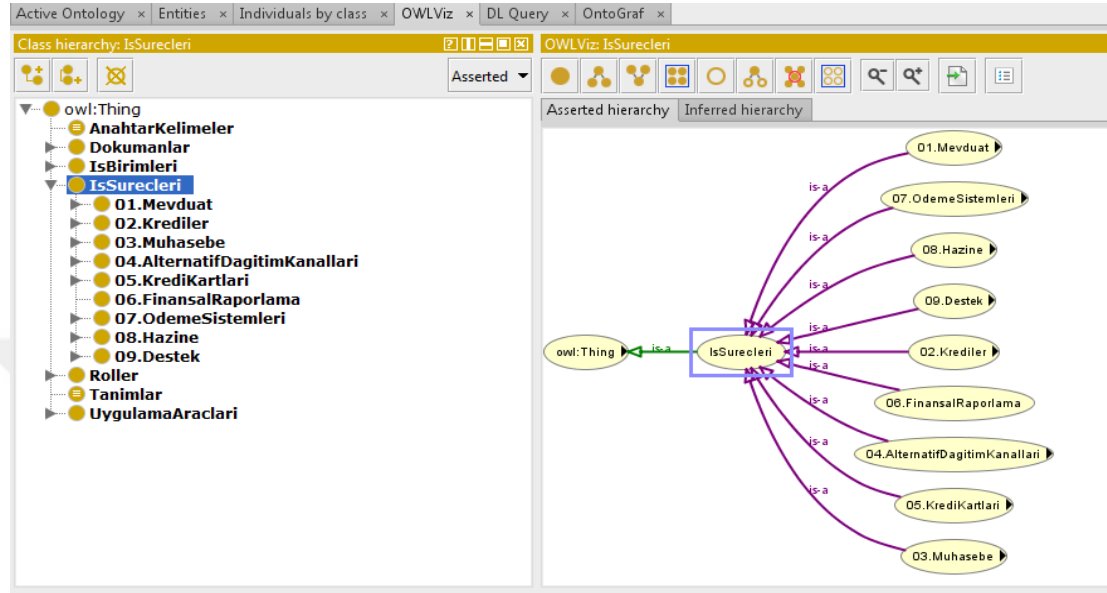


Figure 4.5 Main processes

As shown in Figures 4.6-4.13, each of main processes are subdivided into subprocesses that form the main processes and express the interrelated process groups.

Figure 4.6 shows four subprocesses of *01.Mevduat* main process. Each of subprocess is subclass of *01.Mevduat* class and each of them are disjoint classes.

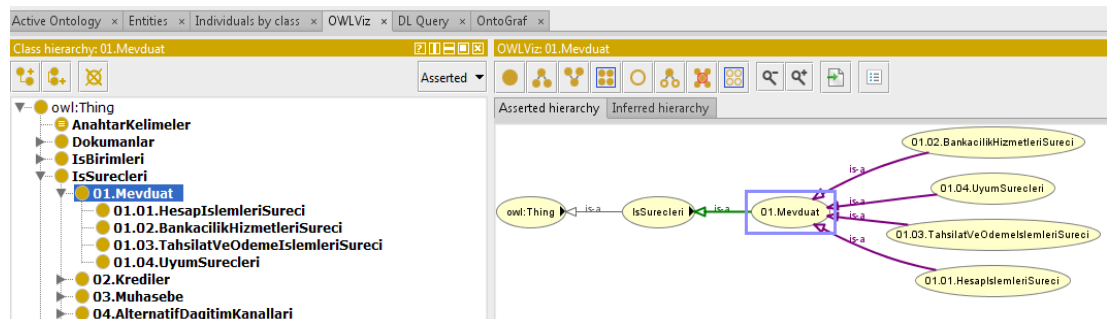


Figure 4.6 Subclasses of deposit main process

Figure 4.7 shows nine subprocesses of *02.Krediler* main process.

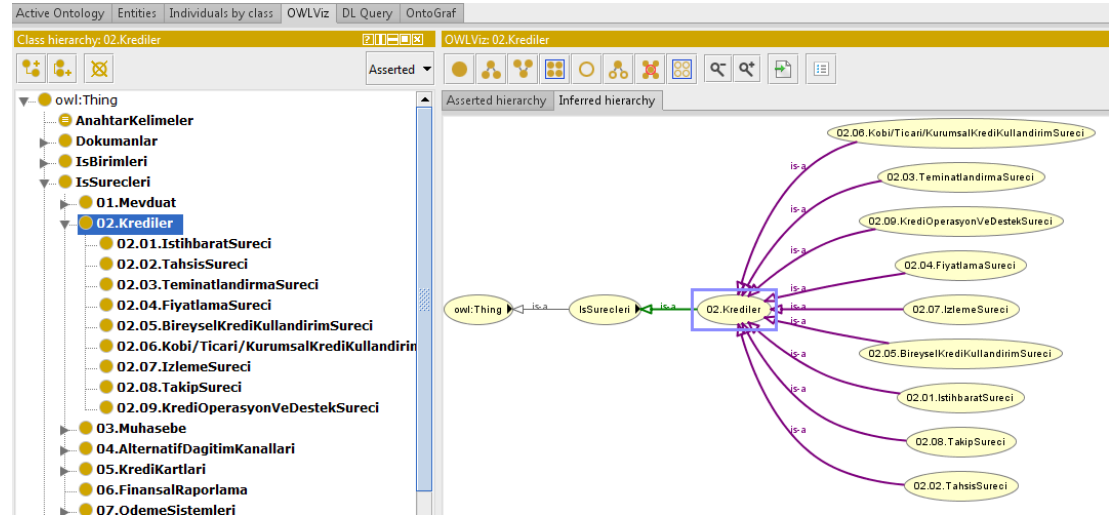


Figure 4.7 Subclasses of credit main process

Figure 4.8 shows four subprocesses of *03.Muhasebe* main process.

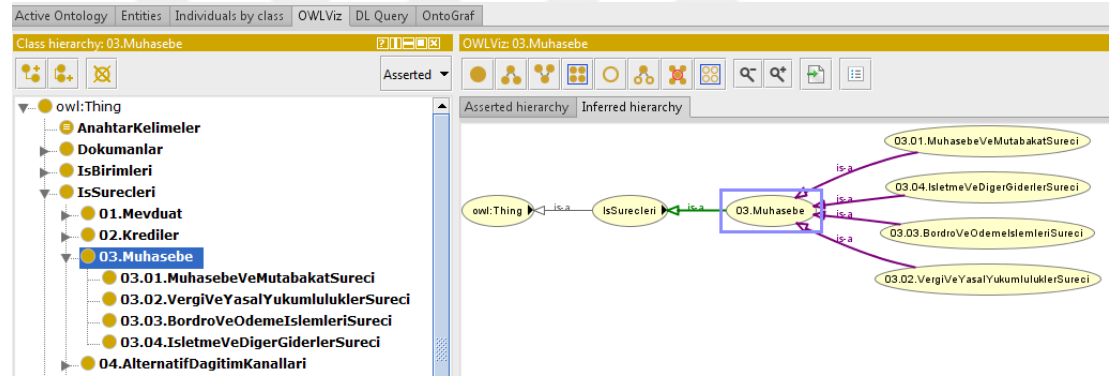


Figure 4.8 Subclasses of accounting main process

Figure 4.9 shows four subprocesses of *04.AlternatifDagitimKanallari* main process.

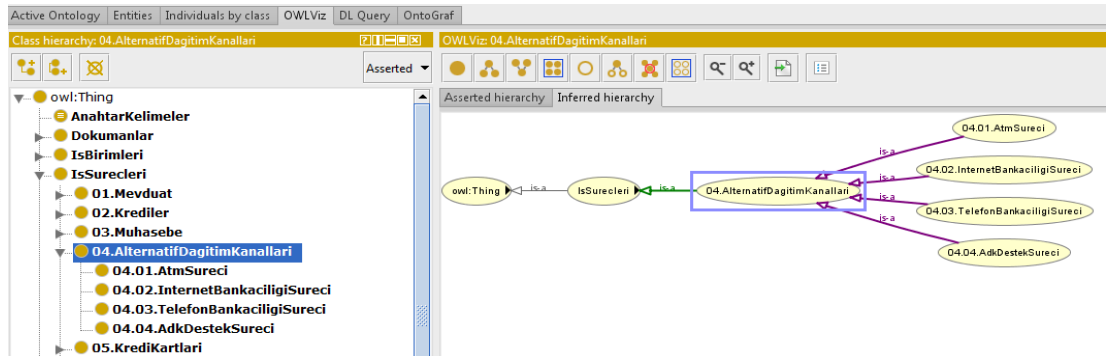


Figure 4.9 Subclasses of alternative distribution channels main process

Figure 4.10 shows two subprocesses of *05.KrediKartlari* main process.

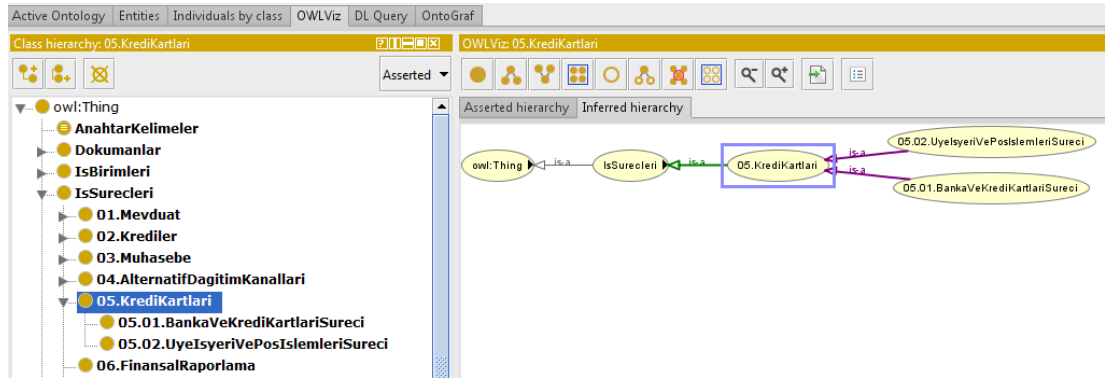


Figure 4.10 Subclasses of credit cards main process

Figure 4.11 shows three subprocesses of *07.OdemeSistemleri* main process.

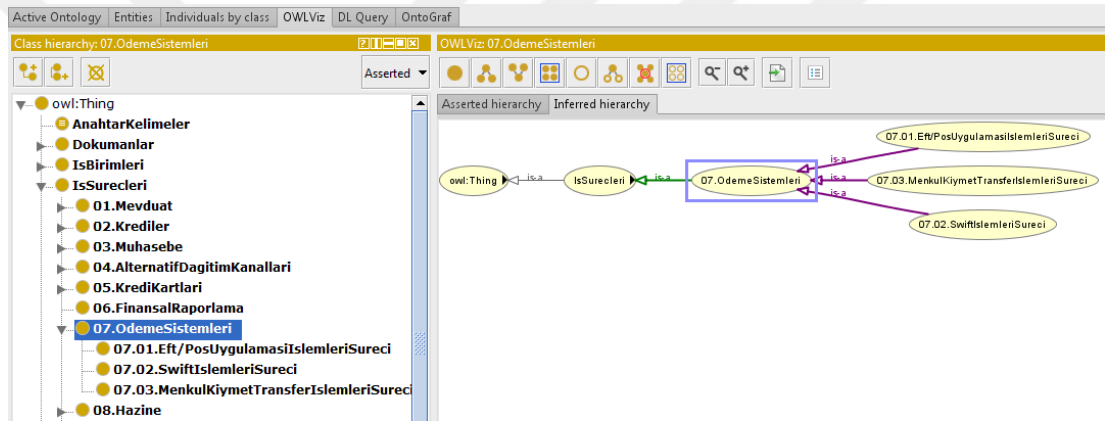


Figure 4.11 Subclasses of payment systems main process

Figure 4.12 shows six subprocesses of *08.Hazine* main process.

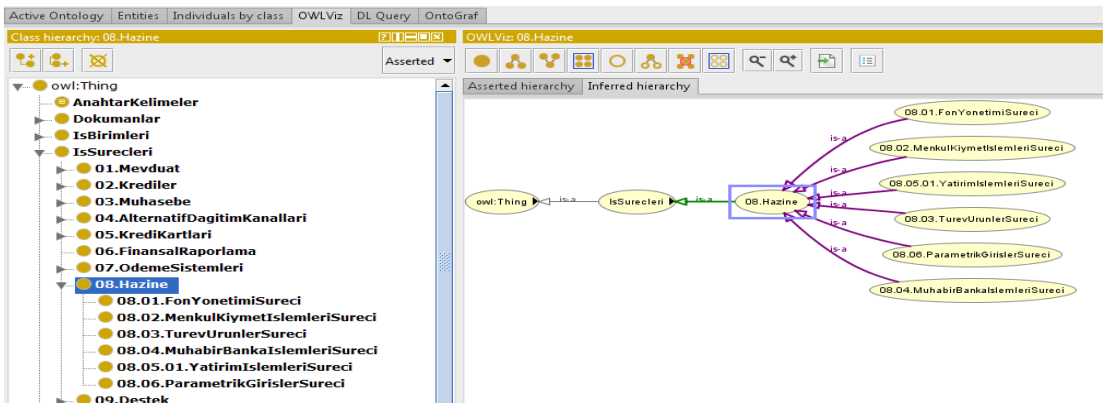


Figure 4.12 Subclasses of treasury / securities and fund management main process

Figure 4.13 shows thirteen subprocesses of *09.Destek* main process.

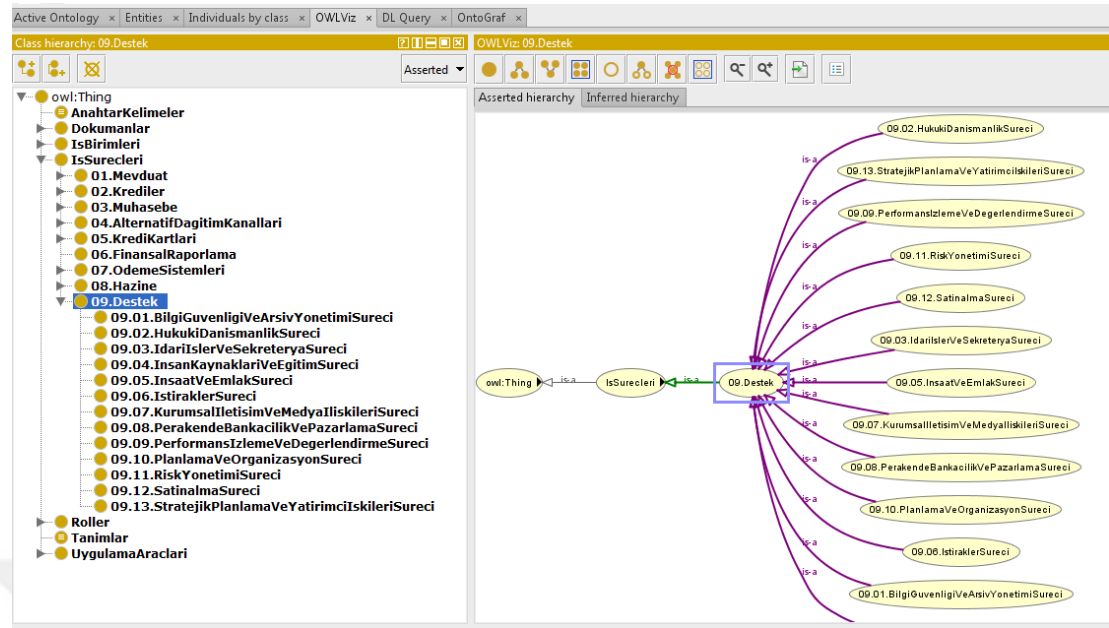


Figure 4.13 Subclasses of support main process

After adding the main process classes (total of nine) and their subprocess classes (total of forty five) to the ontology, these classes are defined as disjoint, so that an individual (process) cannot be an example of more than one of these forty five. Figure 4.14 shows that the subprocesses of *04.AlternatifDagitimKanallari* main process are defined as disjoint. The same procedure is repeated for all main process classes and subprocess classes.

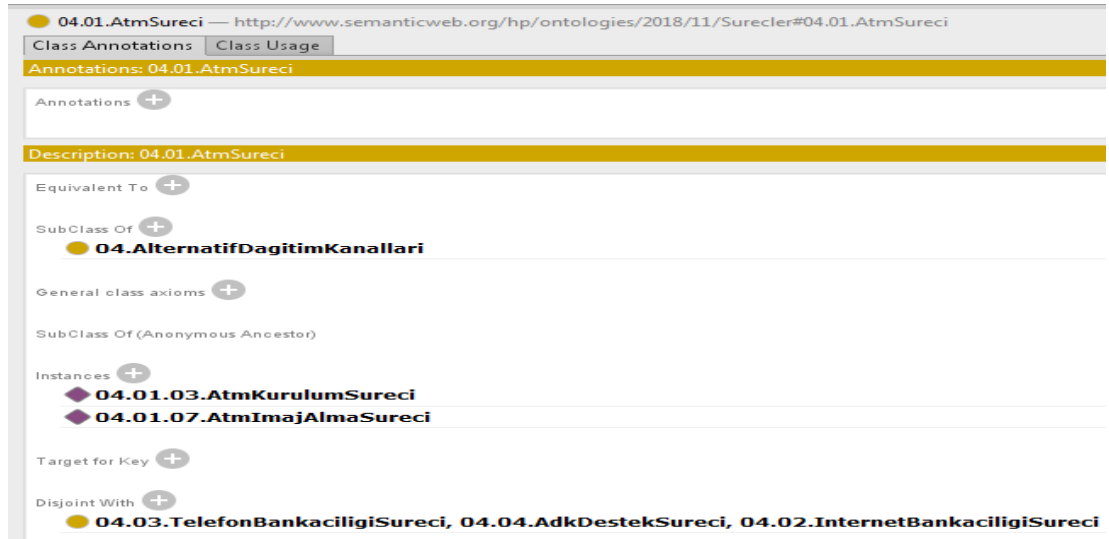


Figure 4.14 Create disjoint classes

In the next step, the business units that are the owners of or responsible of processes are defined. The business unit class and the subclasses below it are created. *IsBirimleri* class is created to show all business units. *IsBirimleri* class is divided into three subclasses: branch (*Sube*), region (*Bolge*), unit (*Birim*). *Birim* class consists of presidencies (subclasses), and directorates (individuals) that are under the presidencies. A total of thirty five presidencies (subclasses) are defined under the *Birim* class and Figure 4.15 shows some of these. Finally, similar to the example in Figure 4.14, all subclasses of the *IsBirimleri* class are defined as disjoint. This means that an individual belonging to the *Sube* class does not belong to a *Bolge* or *Birim* class, or an individual belonging to the *Birim* class cannot belong to a *Sube* or *Bolge* class.

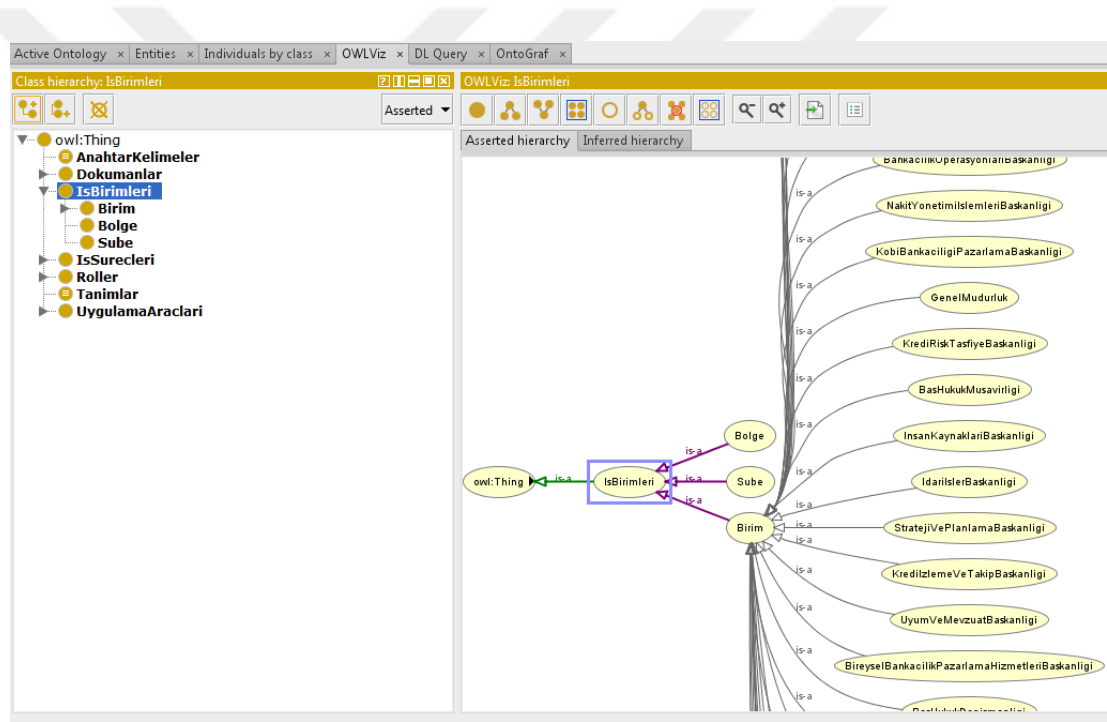


Figure 4.15 Subclasses of the Business Units Class

Another reason that may lead to a change in business processes is the change in the documents they refer to. For this reason, *Dokumanlar* class is defined and divided into twelve disjoint subclasses. As shown in Figure 4.16, the subclasses of the *Document* class are announcements (*Duyurular*), forms (*Formlar*), circulars (*Genelgeler*), prospectuses (*Izahnameler*), laws (*Kanunlar*), authority approvals (*MakamOnaylari*), letters (*Mektuplar*), contracts (*Sozlesmeler*), notices (*Tebliğler*),

articles (*Yazilar*), board decisions (*YonetimKuruluKararlari*), and regulations (*Yonetmelikler*).

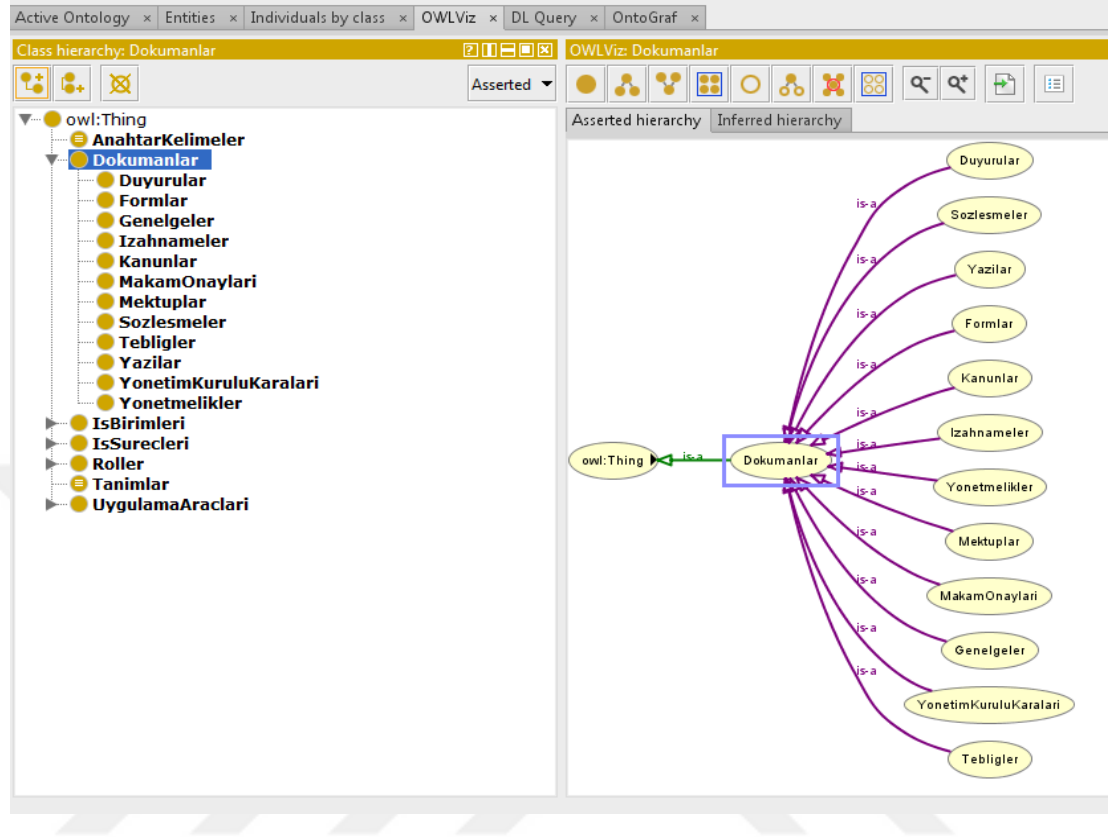


Figure 4.16 Subclasses of the Documents class

The changes in the roles referenced in the process may cause the process to be updated. For this reason, as shown in Figure 4.17, the *Roller* class and its two disjoint subclasses that are *Operasyon* class and *Satis* class are defined.

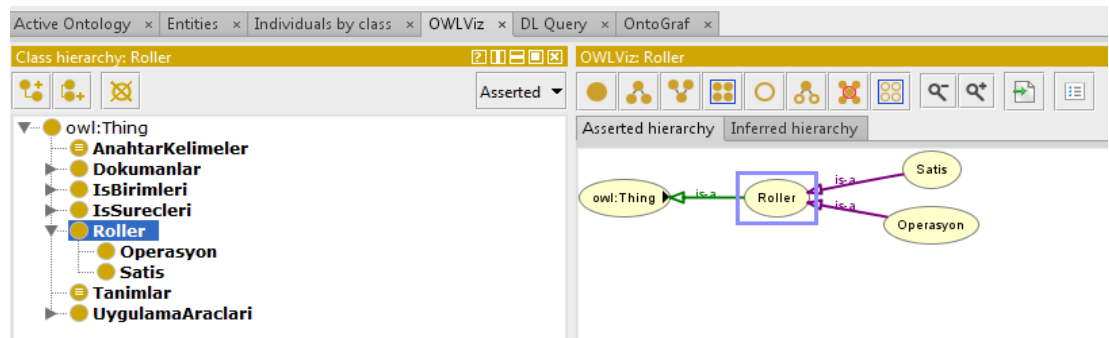


Figure 4.17 Subclasses of the Roles class

Finally, *UygulamaAraclari* class has been defined in order to follow the changes in the application tools referenced in the processes. Figure 4.18 shows the *UygulamaAraclari* class and its two disjoint subclasses that are *DisYazilim* and *IcYazilim*.

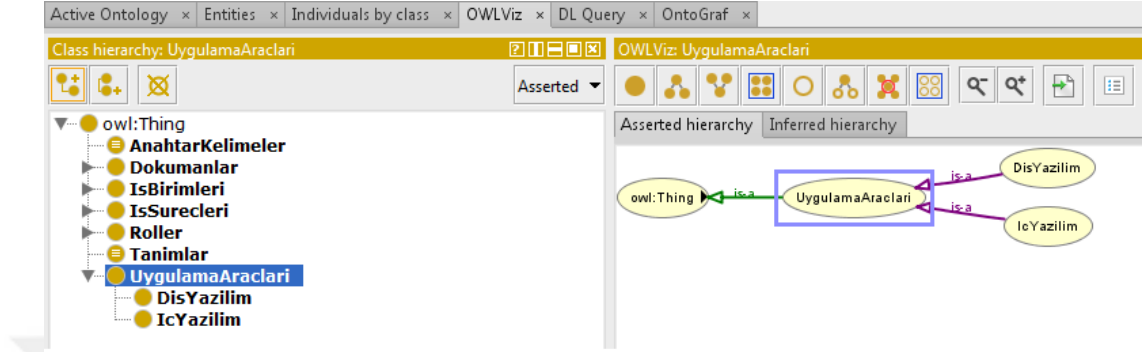


Figure 4.18 Subclasses of the Application Tools class

All classes created so far are referred to as “Named Classes”. *Named Classes* are created by using the "Classes Tab" shown in Figure 4.19.

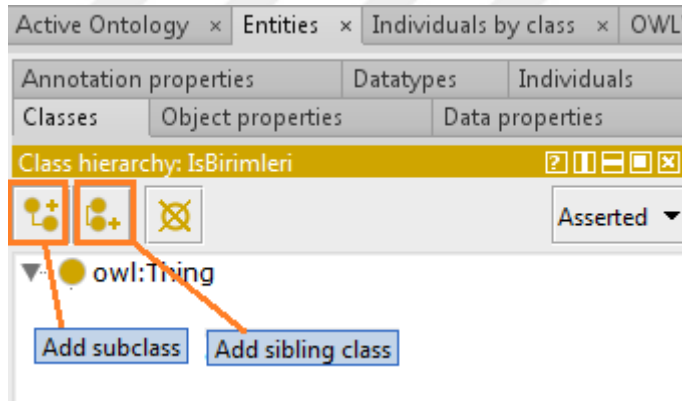


Figure 4.19 Create class tab

There is also another way to create classes: *property restrictions*. A *restriction* describes a class of individuals based on the relationships that members of the class participate in. In other words, a restriction is a kind of class, in the same way that a named class is a kind of class. These kind of classes are known as *Defined Class* or *Equivalent Classes*. Keywords (*AnahtarKelimeler*) and definitions (*Tanimlar*)

classes are examples of this type of class and will be presented once the properties of ontology are described.

4.2.2. Properties

OWL Properties represent relationships. There are two main types of properties, object properties and datatype properties. The *object properties* define the relationships between two individuals, while the *datatype properties* define the relationships between individuals and data values.

OWL allows the enrichment of the meaning of properties through the use of property characteristics. Object property characteristics that OWL allows are shown in Figure 4.20.

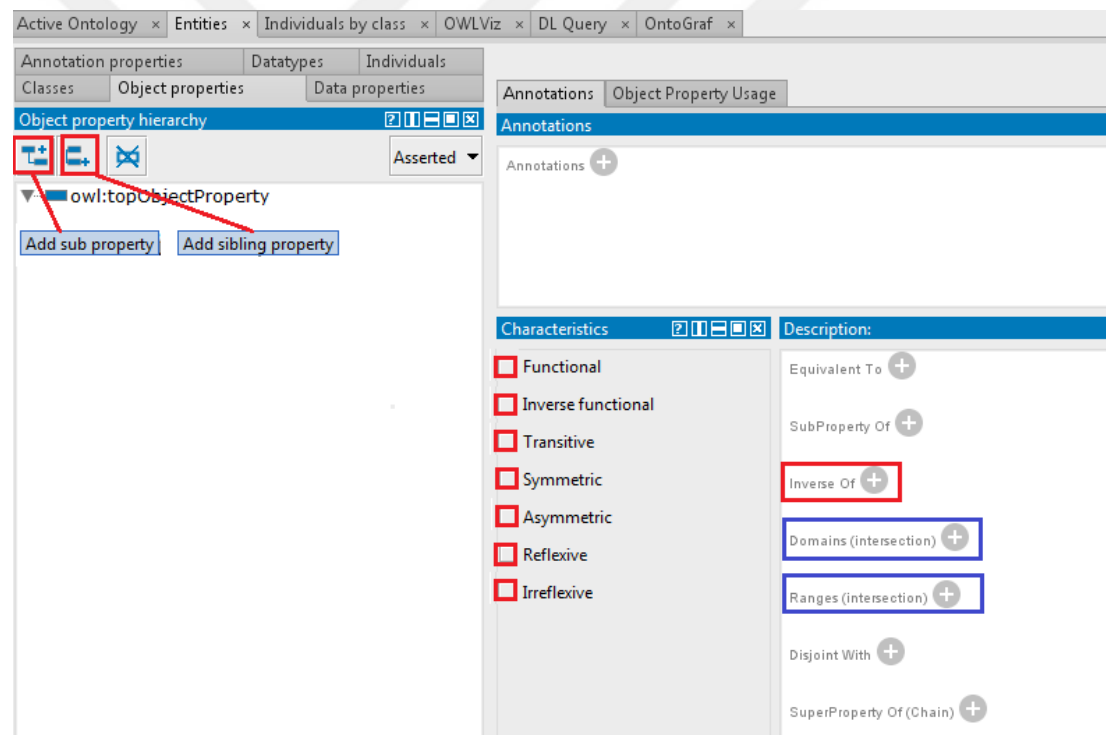


Figure 4.20 Object property characteristics

Each object property may have a corresponding *inverse property*. If some property links individual **A** to individual **B** then its inverse property will link individual **B** to individual **A**.

Properties link individuals from the domain to individuals from the range, thus properties may have a domain and a range specified as seen in Figure 4.20. Domains and ranges should not be viewed as constraints to be checked. They are used as ‘axioms’ in reasoning. Therefore, care must be taken when using them, otherwise unexpected classification results can lead to problems and unexpected side effects can be particularly difficult to track down in large ontologies.

Figure 4.21 shows the defined object properties and Figure 4.22 shows the relationships established between the *IsSurecleri* class and the other classes through these properties.

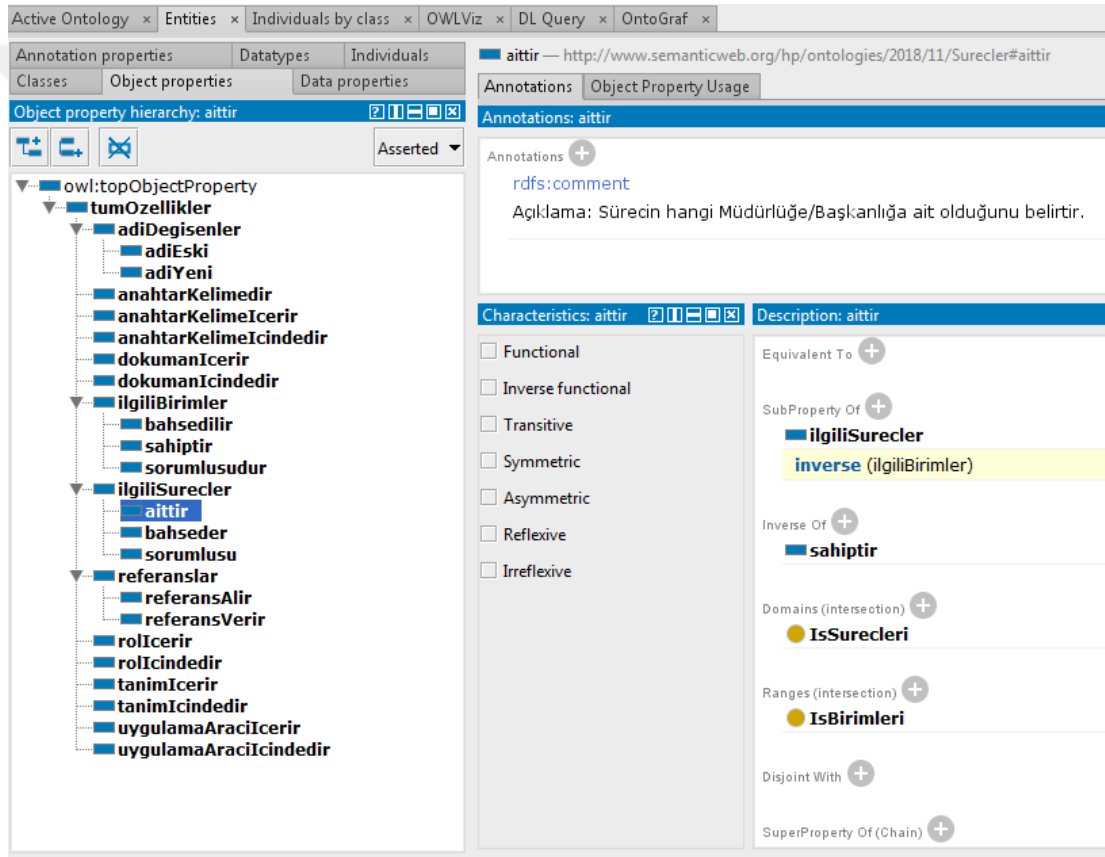


Figure 4.21 Object properties defined in the ontology

With the help of defined properties and established relations, see Figure 4.21-4.22, the ontology can be queried by logical expressions. Consequently, change needs in processes can be detected, and while reflecting them to process documents, improvement opportunity can be caught by analyzing the processes, or deficiencies can be identified.

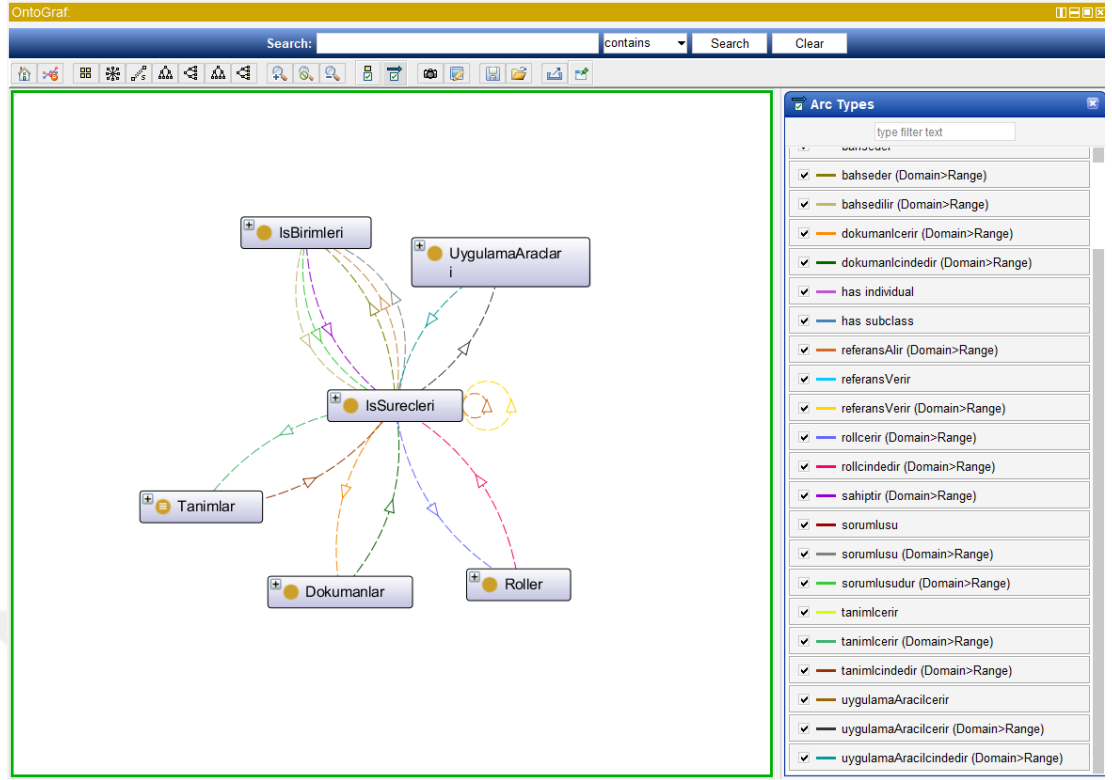


Figure 4.22 Business Processes class and its relation with other classes

Object properties briefly serve the following purposes:

Each process has at least one process owner. Process owner is a business unit that has the authority to put rules on the processes and they are the main units that are responsible for updating the processes they have. To create this relation between process classes and business unit classes, **aittir** object property and its inverse property **sahiptir** are defined as seen in Figure 4.22. **aittir** property links individuals from the domain *IsSurecleri* to individuals from the range *IsBirimleri*, while **sahiptir** property link individuals from the domain *IsBirimleri* to individuals from the range *IsSurecleri*, see Figure 4.22

Processes can also have process responsible business units. During the process update, these units submit their opinions on the processing steps they perform within the process. To create this relation between the process classes and business unit classes, **sorumlusu** object property and its inverse property **sorumlusudur** are defined. **sorumlusu** property link individuals from the domain *IsSurecleri* to individuals from the range *IsBirimleri* while **sorumlusudur** property link individuals

from the domain *IsBirimleri* to individuals from the range *IsSurecleri*, see Figure 4.22.

In the Excel document of the processes, reference can be made to the business unit names in the explanation column of the process detail page, see Figure 4.2. During the process update, if changes in these unit names are detected, or if it is deemed appropriate to remove them, necessary changes are made by process owner or process coordinator. These units have no responsibility in updating the processes. To create this relation between process classes and business unit classes, **baseder** object property and its inverse property **bahsedilir** are defined. **bahseder** property link individuals from the domain *IsSurecleri* to individuals from the range *IsBirimleri* while **bahsedilir** property link individuals from the domain *IsBirimleri* to individuals from the range *IsSurecleri*, see Figure 4.22.

These six properties, i.e. **aittir**, **sorumlusu**, **bahseder** and their inverse properties, have vital importance. Changes in the organizational structure are announced in the bank system. If there is a change in the organizational structure, it means that there will be changes in the processes ownerships. And most likely these changes will change the process owners as well as the functioning of the processes. Therefore, the processes affected by the changes need to be analyzed.

If there is a change in the name of business units and processes, **adiYeni** object property is going to be used. Its inverse property is **adiEski**. Both properties link individuals from the domain *IsSurecleri* to individuals from the range *IsSurecleri* or link individuals from the domain *IsBirimleri* to individuals from the range *IsBirimleri*.

It is possible to specify multiple classes as the range for a property. If multiple classes are specified as range, the range of the property is interpreted to be the intersection of the classes. It means that if the range of **adiYeni** property is specified as *IsSurecleri* and *IsBirimleri* Classes, the range of the property will be interpreted as intersection of *IsSurecleri* and *IsBirimleri* Classes. So, in order to avoid unexpected classification results, domain and range of **adiYeni** and **adiEski** properties are not defined.

The name of an existing directorate may change for three reasons. With the changes in the organizational structure, business units can be divided or combined, or only the existing unit can be renamed. These changes need to be reflected in the process documents. However, updating processes is time-consuming, difficult and requires continuous monitoring. It is impossible to complete all changes at once. Therefore, it is necessary to query the process domain at intervals. **adiYeni** and its inverse property **adiEski** properties serve this purpose.

Business processes can refer to other business processes. To show this relation, **referansVerir** object property and its inverse property **referansAlir** are defined. **referansVerir** property link individuals from the domain *IsSurecleri* to individuals from the range *IsSurecleri* while **referansAlir** property link individuals from the domain *IsSurecleri* to individuals from the range *IsSurecleri*, see Figure 4.22

Just like business units, business processes can be divided, merged, renamed, canceled or created. The processes associated with these changed processes need to be determined, analyzed and updated. **referansVerir** and its inverse property **referansAlir** properties serve this purpose.

Business processes can be subject to laws, regulations, etc., and may refer to these documents. In case of changes or updates to the referenced documents, the process should be reviewed, too and updated if necessary. With this aim, **dokumanIcerir** object property and its inverse property **dokumanIcindedir** are defined. **dokumanIcerir** property link individuals from the domain *IsSurecleri* to individuals from the range *Dokumanlar* while **dokumanIcindedir** property link individuals from the domain *Dokumanlar* to individuals from the range *IsSurecleri*, see Figure 4.22.

The **dokumanIcerir** property and its inverse property **dokumanIcindedir** can also serve more detailed and logical queries. The processes referring to the same document may be related to each other, and this relationship may be uninvolved or overlooked. Deficiencies can be identified and remedied through such logical queries.

The roles that have authority in performing the process can be specified in the process. In the event of a change in the roles and their authority, they may need to be

reflected in the processes. Also, there may be a need to query the roles and the processes they are responsible for. With this aim, **rollIcerir** object property and its inverse property **rollIcindedir** are defined. **rollIcerir** property link individuals from the domain *IsSurecleri* to individuals from the range *Roller* while **rollIcindedir** property link individuals from the domain *Roller* to individuals from the range *IsSurecleri*, see Figure 4.22.

Role based logical queries can help in project developments. For example, there is a project that is planned to combine the roles of *Bpy* and *Gise* responsible for branch operations. The processes in which these roles take place are not fully known. It is tried to determine the transactions they perform with database queries. Yet, this study shows that the level of complexity of existing business process documents is insufficient for such logical inquiries. While some processes have sufficient level of detail, many do not. With the help of logical queries to be made in the ontology, deficiencies can be identified and eliminated.

Application or communication tools are used to carry through processes which are indicated in the process. If the systems and tools used are changed or new ones are developed, they should be reflected in the processes, as well. With this aim, **uygulamaAraciIcerir** object property and its inverse property **uygulamaAraciIcindedir** are defined. **uygulamaAraciIcerir** property link individuals from the domain *IsSurecleri* to individuals from the range *UygulamaAraclari* while **uygulamaAraciIcindedir** property link individuals from the domain *UygulamaAraclari* to individuals from the range *IsSurecleri*, see Figure 4.22.

Like role based logical queries, application tool based queries can help in project development. For example, there is a project that plans to develop a system similar to the *Ets* system on the *Vit* system. *Ets* is an application written by an external company that allows the sharing of official and digitally signed texts and documents between business units. *Vit* is actually a very comprehensive project that covers the entire bank and is the acronym for "*Vakıfbank Innovative Transformation*". The aim of the project is to develop an internal software system called *Vit* where all banking works and services are gathered, managed and analyzed together. All discrete systems and software are gradually transferred to the *Vit* system. In this context, any

new changes implemented should be reflected in the relevant process documents. Thus, logical queries can be made on the process domain, such as what are the processes that include the *Ets* application tool. Or what are the processes carried out according to the *letter* of the *external institution*?

Process Excel documents can include definitions and abbreviations in the process overview sheet, see Figure 4.1. By compiling all abbreviations and definitions, a *definitions dictionary* is created, so that users can query on the dictionary and have information about the processes. Also, process updates may be made through definition based logical queries. For example, *Masak* is an abbreviation of “*Mali Suçları araştırma Kurumu*”. If there is a change or update in *Masak* transaction limits, it may be necessary to examine the processes in which *Masak* is mentioned.

With this aim, ***tanımIcerir*** object property and its inverse property ***tanımIcindedir*** are defined. ***tanımIcerir*** property link individuals from the domain *IsSurecleri* to individuals from the range *Tanimlar* while ***tanımIcindedir*** property link individuals from the domain *Tanimlar* to individuals from the range *IsSurecleri*, see Figure 4.22.

To be used in logical querying, the *AnahtarKelimeler* class is defined. As mentioned before, *property restrictions* describe a class of individuals based on the relationships that members of the class participate in. *AnahtarKelimeler* is an equivalent class defined by using ***anahtarKelimelidir*** object property, see Figure 4.23.

AnahtarKelimeler class is defined using *necessary and sufficient* condition that if any (random) individual has at least one ***anahtarKelimelidir*** property, then this condition is *sufficient* to determine that the individual must be a member of the class *AnahtarKelimeler*.

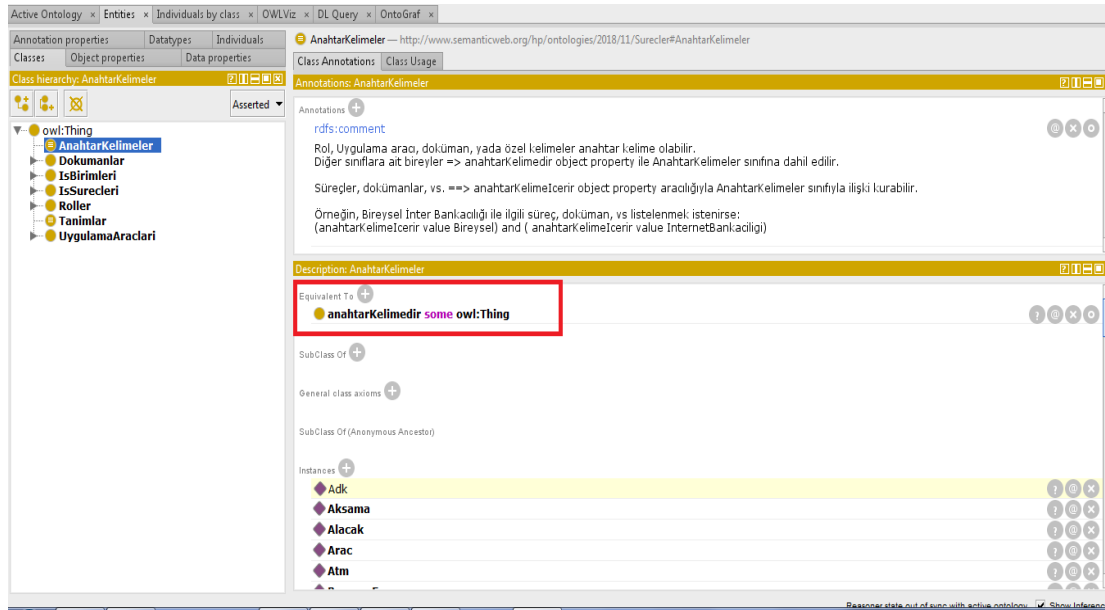


Figure 4.23 Description of AnahtarKelimeler class

For example, as seen in Figure 4.24, individual *Adk* that is an acronym of “*Alternatif Dağıtım Kanalları*”, is a member of *Tanimlar* class and it has a relation with itself via *anahtarKelimedir* property. It means that *Adk* that is a member of *Tanimlar* class is a member of *AnahtarKelimeler* class, too at the same time.

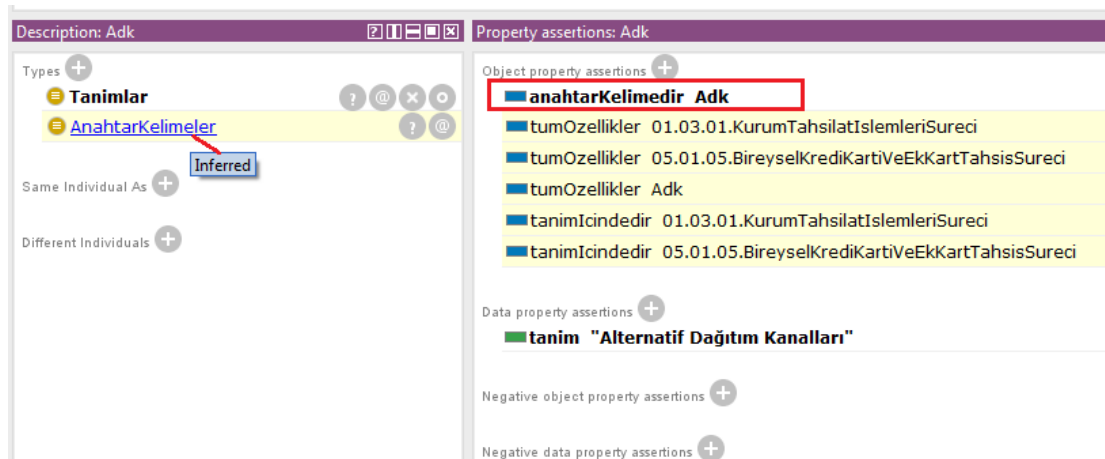


Figure 4.24 Description of individual Adk

After defining *AnahtarKelimeler* class, two further object properties are defined as *anahtarKelimeIcerir* and *anahtarKelimeIcindedir*. They are inverse properties of each other. *anahtarKelimeIcerir* link individuals from the domain *Dokumanlar*,

IsBirimleri, *IsSurecleri*, *Roller*, *Tanimlar*, and *UygulamaAraclari* classes to individuals from the range *AnahtarKelimeler* class.

All individuals who are members of any class can refer to keywords. Thus, logical relations can be established between all classes. For example, *Merkezilesme* is a keyword. If centralized transactions (processes), the business units responsible for carrying out these centralized transactions, and declarations (documents) on how the transactions should be sent to the center give reference to the *Merkezilesme* keyword, then logical queries based on *Merkezilesme* can be made, such as,

- Which transactions can be sent to the center?
- What rules should be followed when sending transactions to the center?
- Which business units are responsible for the transactions carried out at center?

In OWL, properties may have sub-properties, so that it is possible to form hierarchies of properties. Sub properties specialize their super properties (in the same way that subclasses specialize their super classes).

As seen in Figure 4.21, ***ilgiliSurecler*** is a super property of ***aittir***, ***bahseder***, ***sorumlusu*** properties. ***ilgiliSurecler*** property is used to see processes that give reference to business units via ***aittir***, ***bahseder***, or ***sorumlusu*** properties.

ilgiliBirimler is a super property of ***bahsedilir***, ***sahiptir***, ***sorumlusudur*** properties. ***ilgiliBirimler*** property is used to see business units that take reference from processes via ***bahsedilir***, ***sahiptir*** or ***sorumlusudur*** properties, see Figure 4.21.

referanslar is a super property of ***referansAlir*** and ***referansVerir*** properties. ***referanslar*** property is used to see processes that give reference to other processes via ***referansAlir*** or ***referansVerir*** properties, see Figure 4.21.

adiDegisenler is a super property of ***adiEski*** and ***adiYeni*** properties. ***adiDegisenler*** property is used to see processes or business units that has new or old name via ***adiEski*** or ***adiYeni*** properties, see Figure 4.21.

tumOzellikler is a super property of whole properties. **tumOzellikler** property is used to see all relations of an individual, see Figure 4.21. This property can be thought of as the identity of individuals. If a process (individual) is queried via this feature, the individuals referenced by the process, or all individuals referring to the process, can be listed with a single query.

After finishing the definition of *object properties*, *data properties* are defined as seen Figure 4.25. Data properties describe relationships between an individual and data values.

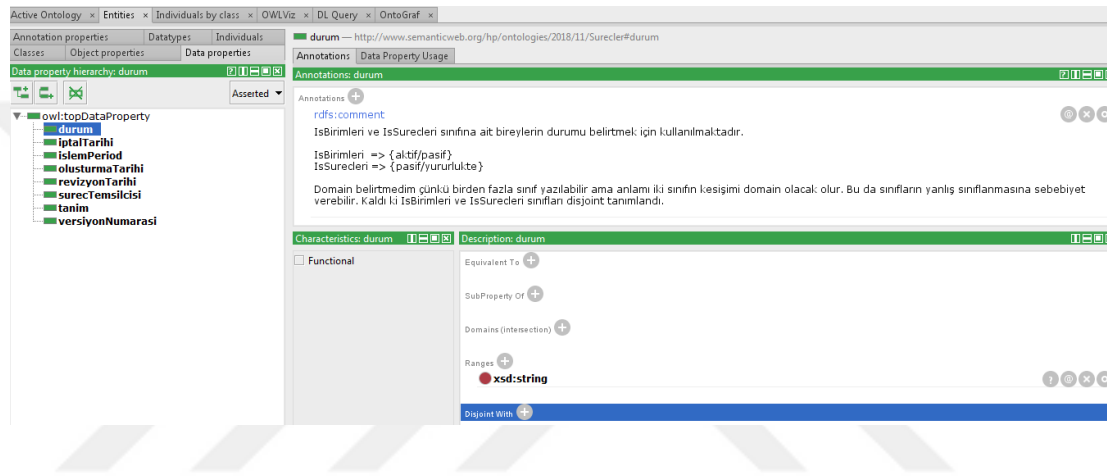


Figure 4.25 Data properties defined in the ontology

The ***durum*** property indicates the status of the business units and processes. If a business unit is turned off, the ***durum*** value is "pasif", otherwise it is "aktif". If a process is canceled, the ***durum*** value is "iptal", otherwise it is "yururlukte". ***durum*** is the most commonly used data properties in queries. Since processes can refer to each other, processes that give reference to a canceled process need to be examined, at least the relationship between them must interrupted. The same situation applies to closed business units.

The ***iptalTarihi*** property indicates the cancellation date of business process if the process is canceled.

The ***islemPeriod*** property indicates monthly operation frequency of the business process if specified in the Excel document, see Figure 4.2.

The ***olusturmaTarihi*** property indicates the first creation date of the business process.

The ***revizyonTarihi*** property indicates the latest revision date of the business process. This property can also be used for logical queries. For example, it may be useful to examine a process that has not been revised for a long time. There may be processes that are overlooked and not revised because they are not very common processes. Or there can be processes that were repealed but forgotten to cancel.

The ***surecTemsilcisi*** property indicates the person or persons responsible for the processes that the business unit has. If there are any processes that need to be updated or if information about the processes is requested, the persons who are responsible for processes are contacted.

The ***tanim*** property is used to define the meaning of abbreviations or expressions used in processes. As seen in Figure 4.26, ***tanim*** property is used to define *Tanimlar* class. *Tanimlar* class is defined using *necessary and sufficient* condition that if any (random) individual has at least one ***tanim*** property then this condition is *sufficient* to determine that the individual must be a member of the class *Tanimlar*.

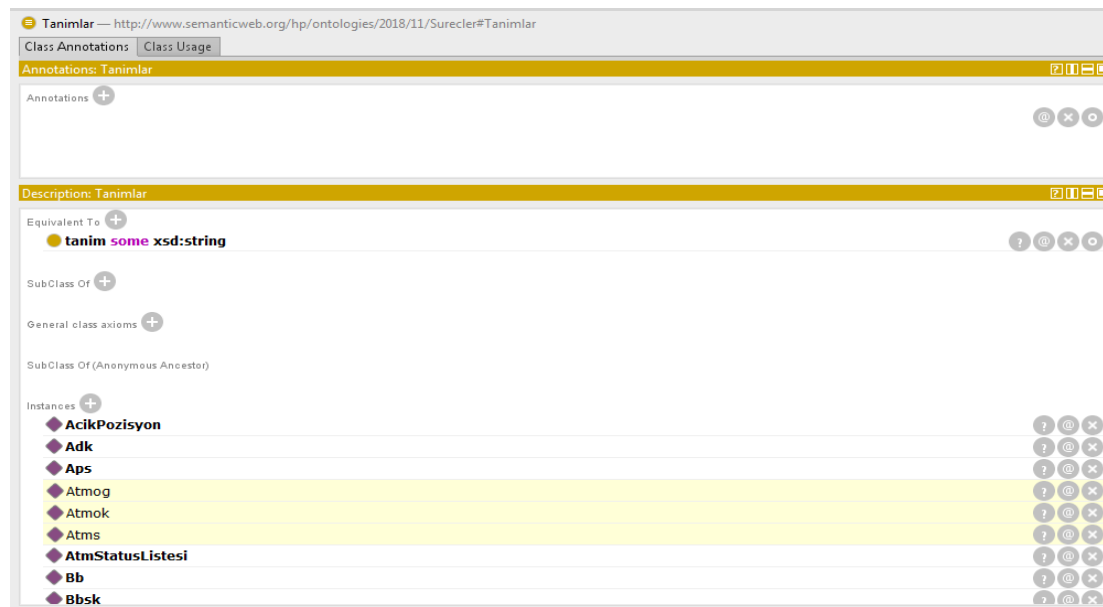


Figure 4.26 Description of *Tanimlar* class

For example, as seen in Figure 4.27, individual *Atmok* is member of *Operasyon* class that is subclass of *Roller* class, and it has definition “ATM Operasyon Koordinatörü” via *tanım* data property. It means that *Atmok* that is member of *Operasyon* class is member of *Tanimlar* class, too at the same time.

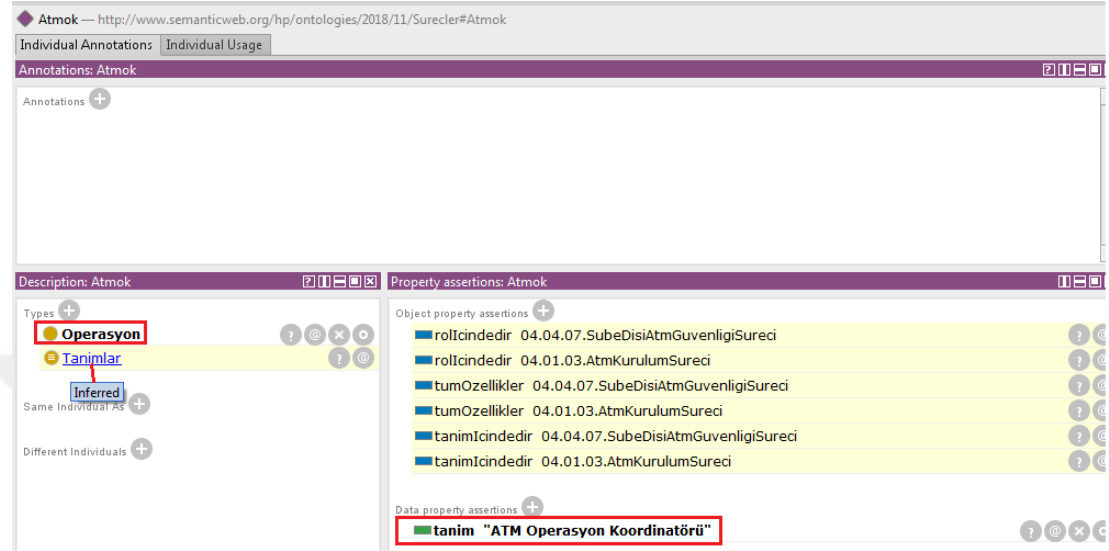


Figure 4.27 Description of individual Atmok

4.2.3. Individuals

After all classes and properties are defined, individuals of classes are created and relations between them are settled.

In the available structure, for all processes (total 621), there is no summary list or document that shows what are the business units, roles, application tools, reference documents, definitions, or reference processes that take place in processes. However, such a list was needed to evaluate the results of the queries. An Excel document was created for this purpose. In the first page of document, processes list take place. The other pages contain information about these processes such as roles, documents, business units, reference processes, definitions, keywords, application tools, that are related to processes. Screenshots of this document is given in Appendix A. Since the creation and management of such a document for all processes would be both time consuming and difficult, only 83 processes were selected and individuals that are given reference in these 83 processes, are defined in the ontology. Some examples of the defined individuals are given in Figure 4.28-4.34. The “purple diamond symbol”

in the figures represents individuals, and “arc types” represents object properties that are settled between individuals.

Figure 4.28 shows four subclasses of *01.Mevduat* class and thirteen processes (individuals) are defined under them.

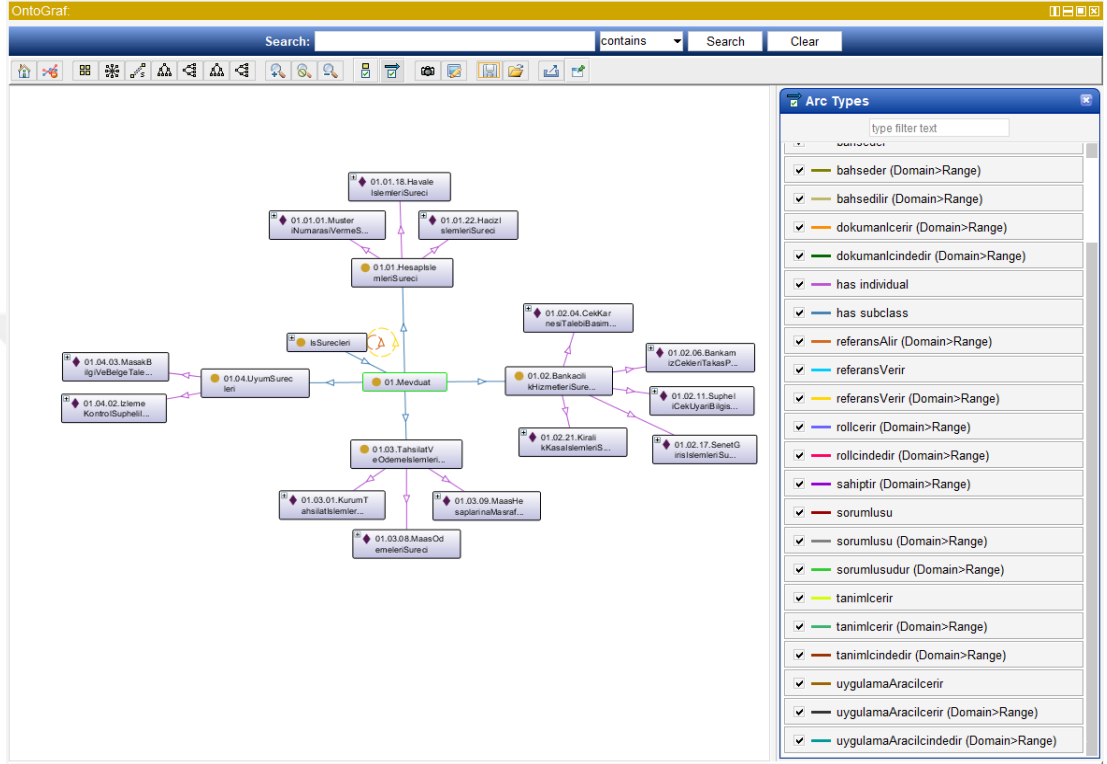


Figure 4.28 Individuals defined under 01.Mevduat main process

Figure 4.29 shows five directorates under the *BankacilikOperasyonlariBaskanligi* that is member of *IsBirimleri* class, and eighteen processes are defined under these five directorates.

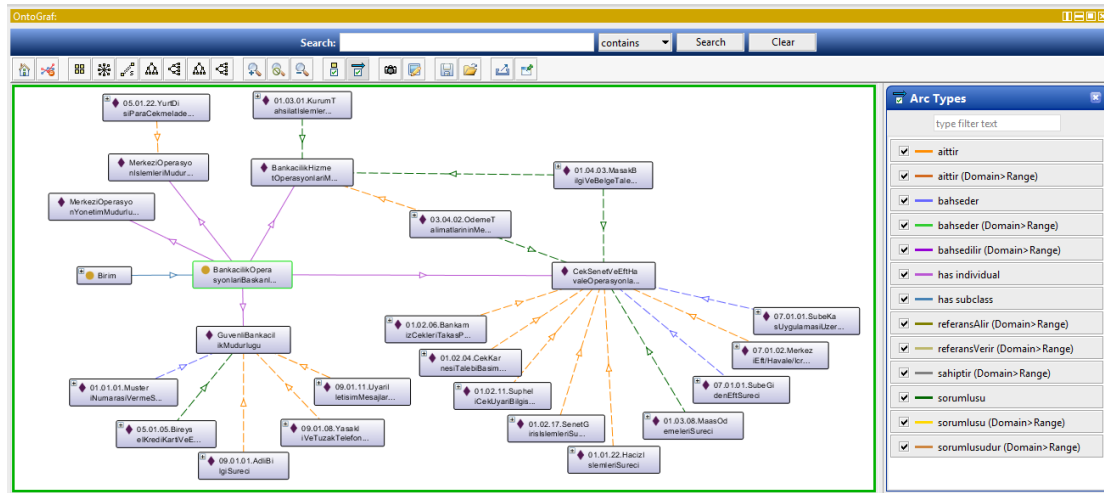


Figure 4.29 Individuals defined under IsBirimleri class

Figure 4.30 shows the twelve subclasses of the *Documents* class. There are 180 defined individuals under the *Documents* class. Since not all of these individuals can be shown in Figure 4.30, only individuals (four) belonging to *MakamOnaylari* and *Duyurular* subclasses are shown. Figure 4.30 also shows individuals related to these four individuals.

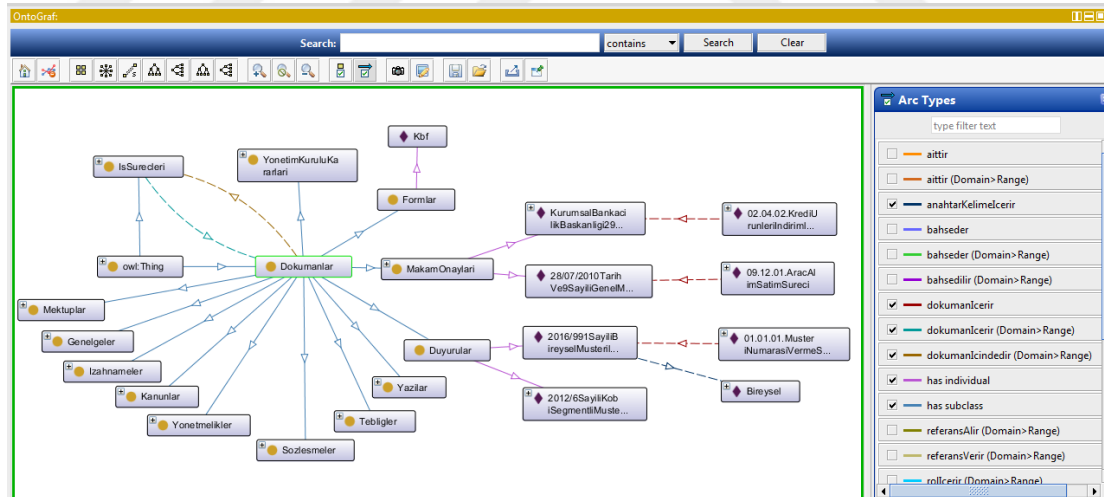


Figure 4.30 Individuals defined under Dokumanlar class

Figure 4.31 shows the two subclasses that *Satis* and *Operasyon*, of the *Roller* class and a total of 18 defined roles are under them.

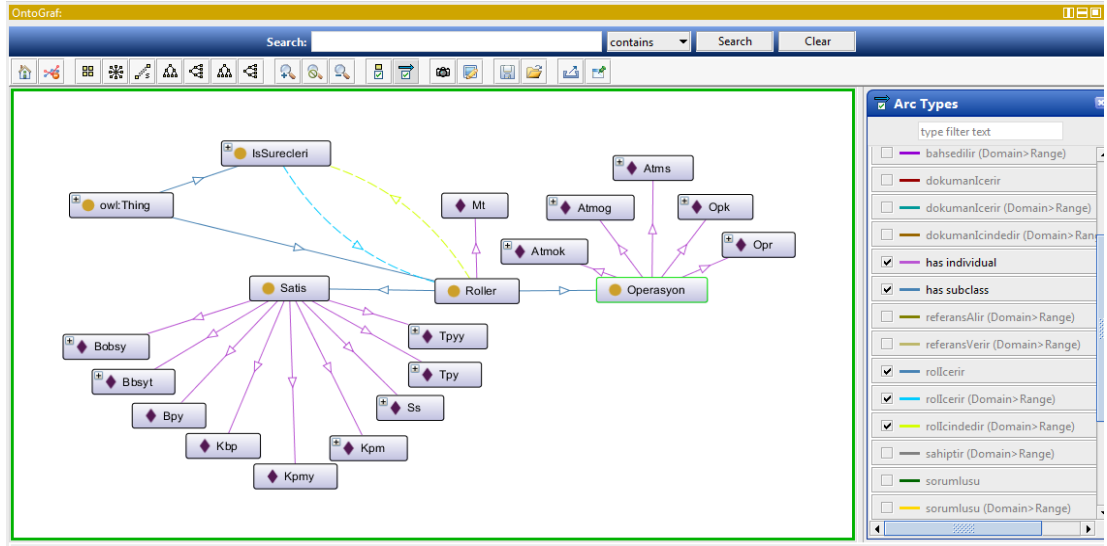


Figure 4.31 Individuals defined of Roller class

Figure 4.32 is an extension of Figure 4.31. It shows a total of eight processes that give reference to the *Opk* role defined under *Operasyon* class.

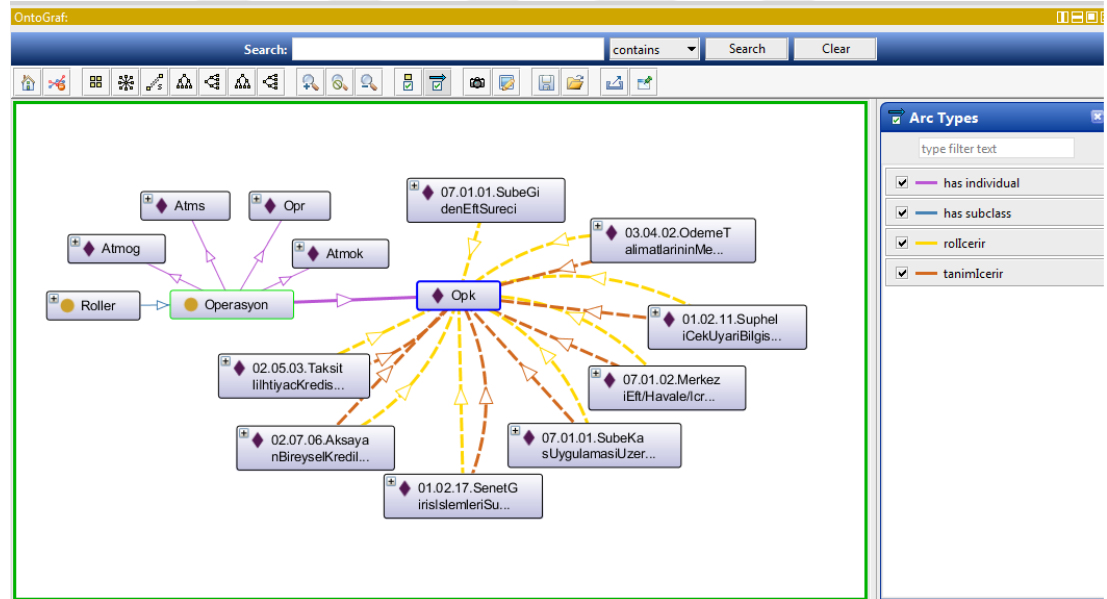


Figure 4.32 Individuals that give reference to the role *Opk*

CHAPTER 5

EVALUATION AND VERIFICATION OF THE ONTOLOGY

5.1 Ontology Verification

The purpose of ontology evaluation is to determine whether the ontology defines the real world completely, consistently and concisely. Two basic approaches to ontology evaluation are ontology verification and ontology validation.

While *Ontology verification* means to check whether the ontology is logically consistent and whether properties are linked correctly with classes based on the axioms, *ontology validation* means to check whether the ontology is coherent with the domain knowledge.

Ontology verification can automatically be done via DL reasoners that evaluate three main taxonomic errors of the ontology that are inconsistency, incompleteness and redundancy. Verification of the developed ontology is done via DL reasoner HermiT 1.3.8.

The result of the reasoner evaluations by HermiT on taxonomic error checking for the developed ontology is presented as follows:

- Consistent: The ontology is consistent without contradiction. There are no roaming errors, section errors and semantic inconsistency errors in ontology.
- Complete: All classes and properties are defined in the ontology. There is no missing relationship or data property claim or explanation.

- No Redundancy: There are no inferred information from the relations, classes and instances of the ontology. All information is defined only once. For *ontology validation*, there is no tool available that can automatically validate the domain knowledge. Ontology design and development involves a tremendous amount of human decisions, including many mental processes. Sometimes even domain experts differ in this process unless they follow the same philosophy and have equal skills. The most common way to validate the ontology is querying it via test cases.

5.2 Evaluation of the Ontology

Figure 5.1 shows a sample set of questions created to test the developed ontology. The Turkish version of the questions set is also given in Appendix B. The questions set can easily be expanded by changing class names, property types, and individual names. In addition, more detailed and complex queries can be made by adding property restrictions.

NO	QUESTION
1	What are the main processes?
2	What are the sub-processes of 01.Mevduat main process?
3	What are the processes, in canceled or in force status, of the 01.Mevduat main process?
4	What are the processes, in canceled or in force status, of the 01.01.HesapIslemleriSureci subprocess?
5	What is the status of the 01.01.01.MusteriNumarasiVermeSureci?
6	What is the creation date of the 01.01.01.MusteriNumarasiVermeSureci?
7	What is the revision date of the 01.01.01.MusteriNumarasiVermeSureci?
8	What is the cancelation date of the 01.01.01.MusteriNumarasiVermeSureci?
9	What is the version number of the 01.01.01.MusteriNumarasiVermeSureci?
10	What are the application tools used in the 01.01.01.MusteriNumarasiVermeSureci?
11	What are the documents referenced by 01.01.01.MusteriNumarasiVermeSureci?
12	What are the definitions in the 01.01.01.MusteriNumarasiVermeSureci?
13	What are the roles in the 01.01.01.MusteriNumarasiVermeSureci?
14	What are the processes that give reference to 01.01.01.MusteriNumarasiVermeSureci?
15	What are the processes that 01.01.01.MusteriNumarasiVermeSureci gives reference to?
16	What are the business units that own the 01.01.01.MusteriNumarasiVermeSureci?
17	What are the processes that belong to MusteriVeHesapOperasyonlariMudurlugu?
18	What are the processes that belong to the BankacilikOperasyonlariBaskanligi?
19	What are the processes, in force status, which involve role Opk?
20	What are the processes, in force status, which involve application tool Ets?
21	What are the processes, in force status, which involve 6415SayiliTerorizminFinansmanininOnlenmesiHakkindaKanun
22	What are the processes, in force status, which involve Masak definition?
23	What are the processes, that give reference to processes that in canceled status?
24	What are the individuals that give reference to Merkezilesme keyword?
25	What are the processes, in force status, with version number zero and revision date older than 2017?
26	What are the processes, in force status, which give reference to old business unit names?
27	What are the processes, in force status, which give reference to old processes names?

Figure 5.1 Sample queries set

As mentioned in Section 4.2.3, an Excel document is created to check the query results. This document keeps all the needed information about the processes that are defined in the ontology, see Appendix A.

Approximately, 150 queries are tried, including the questions shown in Figure 5.1, and the results are compared with the Excel document. The results of all the queries tested are completely consistent with the Excel document. 10 examples of the queries tested will be presented below.

To search the ontology via queries, Protégé DL Query Tab is used. The DL Query tab provides a powerful and easy-to-use syntax based on the Manchester OWL syntax.

Properties link individuals from the domain to individuals from the range (*filler*). A restriction describes a *class* of individuals based on the relationships that members of the class participate in.

Existential restrictions describe classes of individuals that participate in at least one relationship along a specified property to individuals that are members of a specified *class*. The keyword '**some**' is used to denote existential restrictions.

Universal restrictions describe classes of individuals that for a given property only have relationships along this property to individuals that are members of a specified *class*. The keyword '**only**' is used to denote universal restrictions.

The keyword '**value**' is used to denote restriction with a special *individual*.

To create a restriction in DL Query Tab, three things must be done:

1. Property to be restricted is written,
2. Restriction type is written. e.g. 'some' for an existential restriction,
3. Filler (range) is written for the restriction.

Ten of the questions from Figure 5.1 are selected and questioned. The query results are explained below.

Question 1. What are the processes that *give reference* to 01.01.01.MusteriNumarasiVermeSureci?

In the ontology, **referansVerir** and its inverse property **referansAlir** properties are used to establish reference relations between processes (individuals).

Question asks the class of individuals that are linked to 01.01.01.MusteriNumarasiVermeSureci process (individual) via **referansVerir** property.

To write the query, firstly property (**referansVerir**) then restriction type (**value**) and finally filler is written (**01.01.01.MusteriNumarasiVermeSureci**)

Figure 5.2 shows the query written in DL Query Tab and its result. Result of query is presented as 01.02.21.KiralikKasaIslemleriSureci.

It means that 01.02.21.KiralikKasaIslemleriSureci gives reference to 01.01.01.MusteriNumarasiVermeSureci.

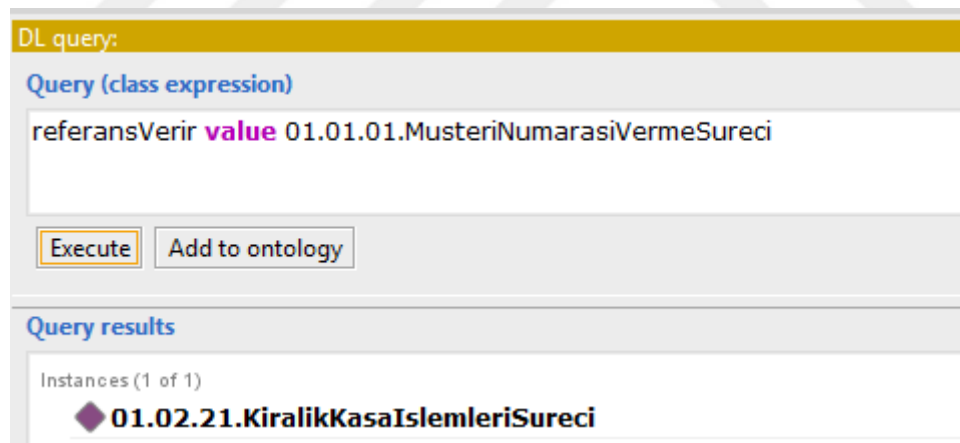


Figure 5.2 Result of the query: What are the processes that *give reference* to 01.01.01.MusteriNumarasiVermeSureci?

Question 2. What are the business units that *own* the 01.01.01.MusteriNumarasiVermeSureci?

Each process belongs to at least one process owner business unit. In other words, there is at least one business unit that owns the process. To show these relations,

aittir and its inverse property ***sahiptir*** properties are defined. ***aittir*** property link individuals from the domain *IsSurecleri* to individuals from the range *IsBirimleri*; ***sahiptir*** property link individuals from the domain *IsBirimleri* to individuals from the range *IsSurecleri*.

In the query, filler (*range*) is 01.01.01.MusteriNumarasiVermeSureci that is member of *IsSurecleri* class. The query result gives the *domain* of individuals that are member of *IsBirimleri* class. Therefore, ***sahiptir*** property is used in the query.

Figure 5.3 shows the query written in DL Query Tab and its result. Result of query is shown as MusteriVeHesapOperasyonlariMudurlugu.

It means that MusteriVeHesapOperasyonlariMudurlugu is owner of 01.01.01.MusteriNumarasiVermeSureci.

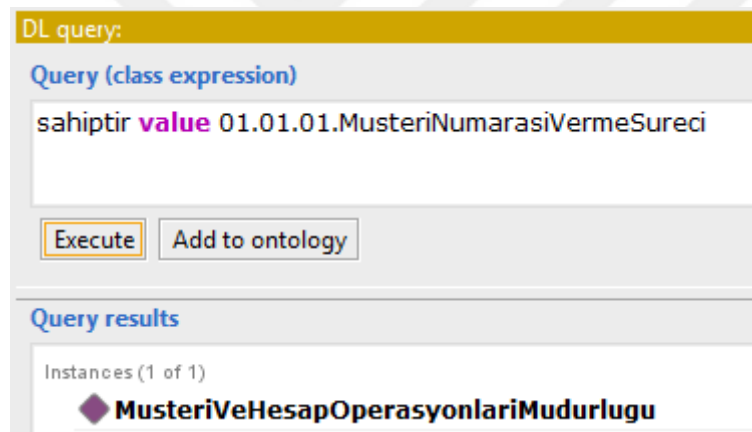


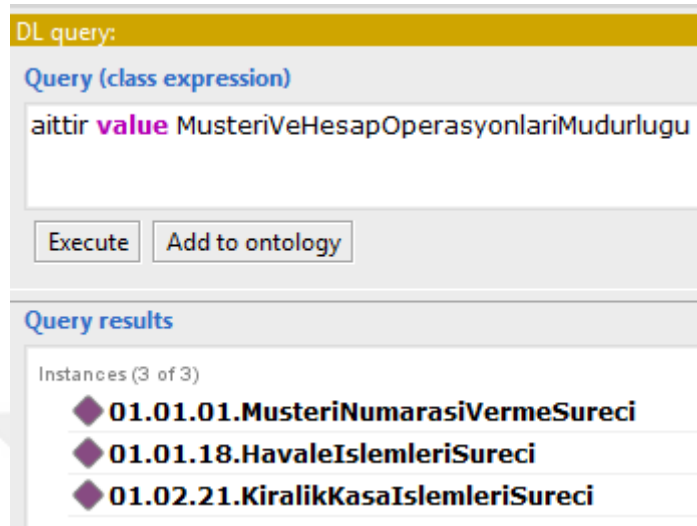
Figure 5.3 Result of the query: What are the business units that own the 01.01.01.MusteriNumarasiVermeSureci?

Question 3. What are the processes that belong to the *MusteriVeHesapOperasyonlariMudurlugu*?

Question 3 is similar to question 2, but this time the question is asked in reverse.

In the query, filler (*range*) is *MusteriVeHesapOperasyonlariMudurlugu* that is member of *IsBirimleri* class. The query result gives the *domain* of individuals that are member of *IsSurecleri* class. Therefore, ***aittir*** property is used in the query.

As seen Figure 5.4, there are the three processes that belong to *MusteriVeHesapOperasyonlariMudurlugu*, in other words, *MusteriVeHesapOperasyonlariMudurlugu* is the owner of the three processes.



The screenshot shows a web interface for a DL query. At the top, there is a yellow bar labeled "DL query:". Below it, a text input field contains the query "aittir value MusteriVeHesapOperasyonlariMudurlugu". Below the input field are two buttons: "Execute" and "Add to ontology". Below the buttons is a section titled "Query results" which shows "Instances (3 of 3)". The results are listed in a table with three rows, each preceded by a purple diamond icon:

Instances (3 of 3)
01.01.01.MusteriNumarasiVermeSureci
01.01.18.HavaleIslemleriSureci
01.02.21.KiralikKasaIslemleriSureci

Figure 5.4 Result of the query: What are the processes that *belong to* the *MusteriVeHesapOperasyonlariMudurlugu*?

Question 4. What are the processes, *in force* status, which involve role *Opk*?

The question asks the processes in force that are associated with *Opk*, which is a member of the *Roller* class.

If a process is in force status, value of ***durum*** data property is “*yururlukte*”. If a process is in cancelled status, value of ***durum*** data property is “*iptal*”.

The object property that link individuals from the domain *IsSurecleri* to individuals from the range *Roller* is ***rollcerir***.

Figure 5.5 shows the query written in DL Query Tab and its result. Be careful, since there are two constraints, so they are connected to each other via ***and*** operator. Result shows that there are seven processes that give reference to *Opk* that member of *Roller* class.

DL query:

Query (class expression)

durum **value** "yururlukte" **and** rolIcerir **value** Opk

Query results

Instances (7 of 7)

- ◆ 01.02.11.SupheliCekUyariBilgisiGirisIslemleriSureci
- ◆ 01.02.17.SenetGirisIslemleriSureci
- ◆ 02.05.03.TaksitliIhtiyacKredisiKullandirimSureci
- ◆ 02.07.06.AksayanBireyselKredilerinYenidenYapilandirilmesiGayrimenkulAlimiileTahsil/TasfiyeSureci
- ◆ 03.04.02.OdemeTalimatlarininMerkezdenYapilmasiSureci
- ◆ 07.01.01.SubeKasUygulamasıUzerindenMesajGonderimSureci
- ◆ 07.01.02.MerkeziEft/Havale/IcraTalimatlarıSureci

Figure 5.5 Result of the query: What are the processes, *in force* status, which involve role *Opk*?

Question 5. What are the processes, *in force* status, which involve application tool *Ets*?

The question asks the processes in force that are associated with *Ets*, which is a member of the *UygulamaAraclari* class.

The object property that link individuals from the domain *IsSurecleri* to individuals from the range *UygulamaAraclari* is ***uygulamaAraciIcerir***.

Figure 5.6 shows the query written in DL Query Tab and its result. Result shows that there are twenty seven processes that give reference to *Ets* that member of *UygulamaAraclari* class.

DL query:

Query (class expression)

durum **value** "yururlukte" **and** uygulamaAraciIcerir **value** Ets

Execute Add to ontology

Query results

Instances (27 of 27)

- ◆ 01.01.22.HacizIslemleriSureci
- ◆ 01.02.04.CekKarnesiTalebiBasimiVeMusteriyeTeslimiSureci
- ◆ 01.02.06.BankamizCekleriTakasProvizyonSureci
- ◆ 01.02.11.SupheliCekUyariBilgisiGirisIslemleriSureci
- ◆ 01.02.21.KiralikKasaIslemleriSureci
- ◆ 01.03.01.KurumTahsilatIslemleriSureci
- ◆ 01.03.08.MaasOdemeleriSureci
- ◆ 01.04.02.IzlemeKontrolSupheliIslemBildirimiVeEgitimSureci
- ◆ 01.04.03.MasakBilgiVeBelgeTalepleriIleMalVarligiDondurmaKararlarininYerineGetirilmesiSureci
- ◆ 02.02.02.TicariKredilerTahsisSureci
- ◆ 02.03.01.TeminatlandirmaSureci
- ◆ 02.03.03.IpotekTesisSureci
- ◆ 02.04.02.KrediUrunleriIndirimliFiyatlamaSureci
- ◆ 02.07.06.AksayanBireyselKredilerinYenidenYapilandirilmesiGayrimenkulAlimiIleTahsil/TasfiyeSureci
- ◆ 02.08.02.TakipRisklerininYapilandirilmesiVeTasfiyeEdilmesiSureci
- ◆ 03.01.10.MenkulIslemlerininMuhasebelestirilmesiSureci
- ◆ 03.02.01.VergiBeyanVeOdemeSureci
- ◆ 03.03.01.PersonelOdemeleriSureci
- ◆ 03.04.02.OdemeTaliimatlarininMerkezdenYapilmasiSureci

Figure 5.6 Result of the query: What are the processes, *in force* status, which involve application tool *Ets*?

Question 6. What are the processes, *in force* status, which involve *6415SayiliTerorizminFinansmanininOnlenmesiHakkindaKanun* document?

The question asks the processes in force that are associated with *6415SayiliTerorizminFinansmanininOnlenmesiHakkindaKanun*, which is a member of the *Dokumanlar* class.

The object property that links individuals from the domain *IsSurecleri* to individuals from the range *Dokumanlar* is ***documanIcerir***.

Figure 5.7 shows the query written in DL Query Tab and its result. Result shows that there are two processes that give reference to *6415SayiliTerorizminFinansmanininOnlenmesiHakkindaKanun* which are members of *Dokumanlar* class.

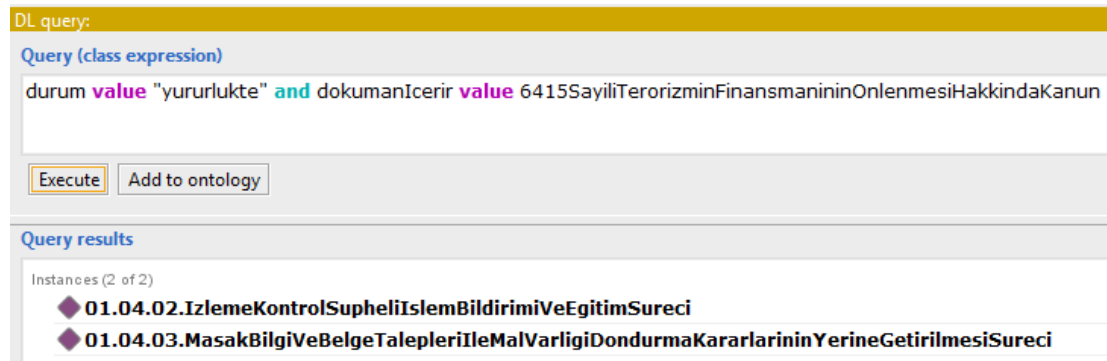


Figure 5.7 Result of the query: What are the processes, *in force* status, which involve 6415SayiliTerorizminFinansmanininOnlenmesiHakkindaKanun law?

Question 7. What are the processes, *in force* status, which involve *Masak definition*?

The question asks the processes in force that are associated with *Masak*, which is a member of the *Tanimlar* class.

The object property that links individuals from the domain *IsSurecleri* to individuals from the range *Tanimlar* is ***tanimIcerir***.

Figure 5.8 shows the query written in DL Query Tab and its result. Result shows that there are seven processes that give reference to *Masak* that member of *Tanimlar* class.

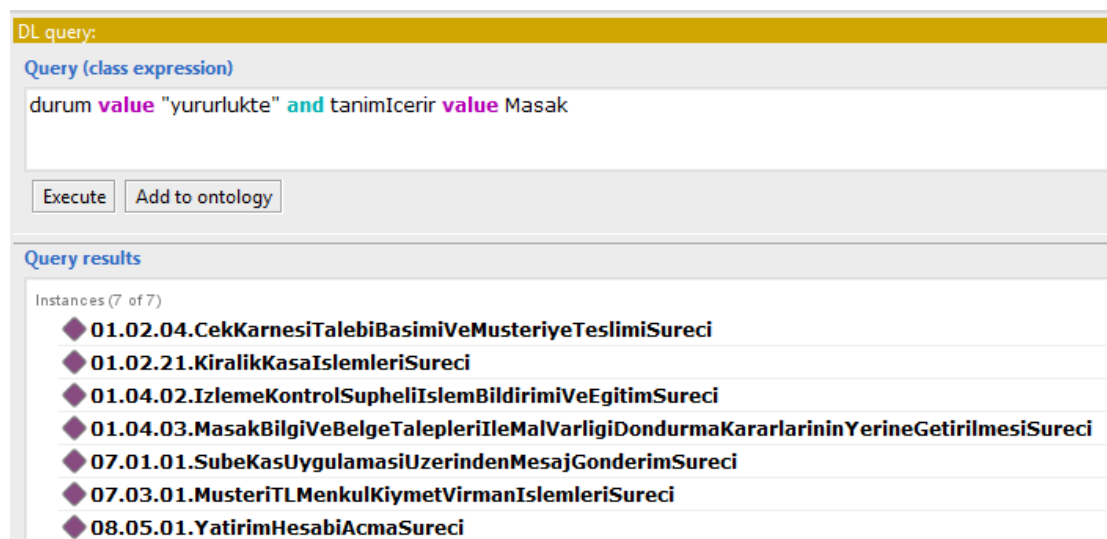


Figure 5.8 Result of the query: What are the processes, *in force* status, which involve *Masak definition*?

Question 8. What are the processes that *give reference* to processes that *in canceled* status?

In the ontology, **referansVerir** and its inverse property **referansAlir** properties are used to establish reference relations between processes (individuals). Domain and range of both properties is *IsSurecleri* class.

Figure 5.9 shows the query written in DL Query Tab and its result. Be careful, restriction type is **some** (existential restriction) not **value**. Because, this time, range is not an *individual*, it is a *class* of individuals. Restriction between () parentheses represents the *class of individuals* that have **durum** value “iptal”

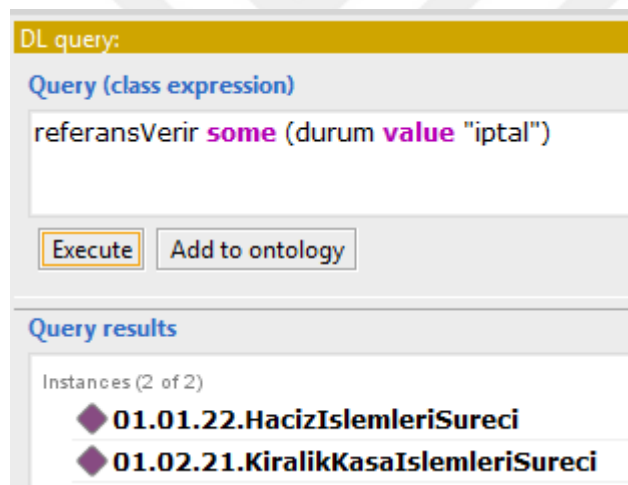


Figure 5.9 Result of the query: What are the processes that *give reference* to processes that *in canceled* status?

Question 9. What are the individuals that give reference to *Merkezilesme* keyword?

In the ontology, two object properties are defined to give reference to *AnahtarKelimeler* class from other classes.

anahtarKelimeIcerir property links individuals from the domain *IsBirimleri*, *IsSurecleri*, *Roller*, *Tanimlar*, *UygulamaAraclari* classes to individuals from the range *AnahtarKelimeler*; **anahtarKelimeIcindedir** property links individuals from

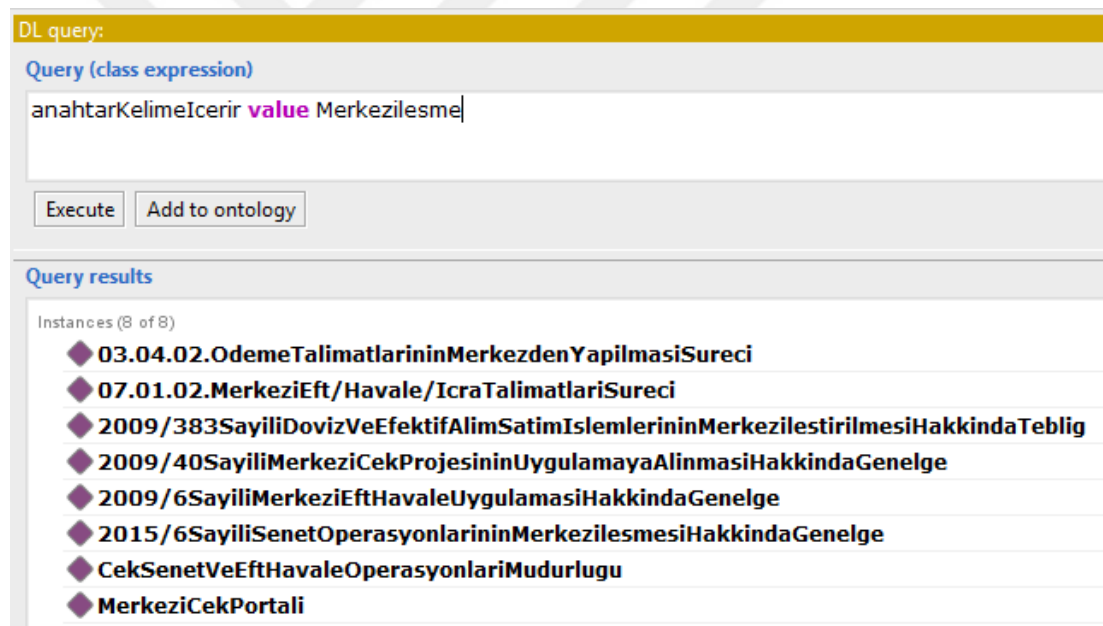
the domain *AnahtarKelimeler* to individuals from the range *IsBirimleri*, *IsSurecleri*, *Roller*, *Tanimlar*, *UygulamaAraclari*.

The question asks the individuals that are members of *any classes* and are associated with *Merkezilesme*, which is a member of the *AnahtarKelimeler* class.

Figure 5.10 shows that there are eight individuals that are members of *different classes*, and are associated with *Merkezilesme*.

If the *business units* that are associated with *Merkezilesme* needs to be found, *IsBirimleri* after **and** operator needs to be written.

The new query will be “*anahtarKelimeIcerir value Merkezilesme and IsBirimleri*”. Then result is only *CekSenetVeEftHavaleOperasyonlariMudurlugu*.



The screenshot shows a web interface for a DL query. At the top, there is a yellow bar labeled "DL query:". Below it, a section titled "Query (class expression)" contains a text input field with the query "anahtarKelimeIcerir value Merkezilesme". Below the input field are two buttons: "Execute" and "Add to ontology". Below these buttons is a section titled "Query results". Under "Query results", it says "Instances (8 of 8)". There are eight instances listed, each preceded by a purple diamond icon:

- 03.04.02.OdemeTalimatlarininMerkezdenYapilmasiSureci
- 07.01.02.MerkeziEft/Havale/IcraTalimatlariSureci
- 2009/383SayiliDovizVeEfektifAlimSatimIslemlerininMerkezilestirilmesiHakkindaTebliğ
- 2009/40SayiliMerkeziCekProjesininUygulamayaAlinmasiHakkindaGenelge
- 2009/6SayiliMerkeziEftHavaleUygulamasıHakkındaGenelge
- 2015/6SayiliSenetOperasyonlarınınMerkezilesmesiHakkındaGenelge
- CekSenetVeEftHavaleOperasyonlariMudurlugu
- MerkeziCekPortali

Figure 5.10 Result of the query: What are the individuals that give reference to *Merkezilesme* keyword?

Question 10. What are the processes, *in force* status, with *version number* zero and *revision date* older than 2017?

Figure 5.11 show that six individuals meet the given conditions.

DL query:

Query (class expression)

durum **value** "yururlukte" **and** versiyonNumarasi **value** 0 **and** revizyonTarihi **some** xsd:dateTime[< 2017-01-01T00:00:00]

Query results

Instances (6 of 6)

- ◆ 05.02.01.UyeIsyeriKampanyaYonetimiSureci
- ◆ 08.06.03.KondorProgramiAdministrationSureci
- ◆ 09.02.04.TakipVeHacizDisiIslemlerleIlgiliHukukiMutalaaSureci
- ◆ 09.04.05.EgitimIhtiyacAnaliziSureci
- ◆ 09.05.01.GayrimenkulAlimVeTakipSureci
- ◆ 09.09.02.PerformansPrimiHesaplanmasiSureci

Figure 5.11 Result of the query: What are the processes, *in force* status, with version number zero and *revision date* older than 2017?

CHAPTER 6

CONCLUSION AND DISCUSSION

Business Process Management (BPM) is an endless effort that requires continuous management of processes after completing process modeling. Although there are BPM tools, they usually put a strong emphasis on the graphical representation of processes where these graphical representations do not supply machine readable semantics.

Usage of ontologies in business process design and management is not a new idea. With the help of ontologies, business users can have a unified view of the business process (including the models and instances of the models) both in notation and in a machine-readable format. This way, querying these processes becomes possible. In addition, ontologies can be easily designed, developed, and extended to new needs. This contributes to the facilitation of process management process itself.

Until today, various ontology based BPM research was done that resulted in methods, techniques and tools to support the design, implementation, management and analysis of operational business processes. These studies are divided into two groups. The first group includes studies that add semantic annotations to indicate the meaning of process elements. Fragments of languages like BPMN and EPCs have been formalized by various authors. Yet, these formalizations cover only selected parts of the language and are at meta-model level. In the second group, there are studies that add semantic annotations to indicate the dynamic behavior of a business process. This group can be evaluated into several subgroups. Some have focused on ontologies to be automatically generated from process models, some on improving

the selected processes models through performance analysis and others on comparing the similarity of the two process models.

The problem is that all business process model based analysis and studies seriously trust on the existence of high-quality models. However, there is a serious lack of harmony between hand-made models and reality. One other problem is that, business processes are not part of the company knowledge but they are available only in the business process tool (if there are any) and/or in the documents. Therefore, they are neither executable nor machine-readable.

The originality of the thesis is that it is not a meta-model. It is a domain specific and reflects the real world example for the selected bank. In doing so, it focuses on the management of the entire process space, rather than focusing on improving one or a few selected process models. Machine readable representations of processes have been defined and thus the harmony between hand-made models and reality has been achieved through logical queries.

In the Turkish Banking System, according to the regulations made by the Banking Regulation and Supervision Agency (BRSA), all Banks are obliged to create work flow charts showing the controls and work steps of the Bank's business processes. Yet, the banks' obligations are not limited to that. Banks are also obliged to keep their processes up-to-date because the Bank's business processes are audited once a year by the Independent Audit Company. As a result of the said audit, penalty is applied if a negative finding is reported. Such a situation not only damages the bank materially, but also damages the image of the bank.

By its very nature, the banking system is one of the sectors where change and development are most frequent. In a sector where change is that fast, keeping all business processes up-to-date is not an easy task.

In the selected bank example, the main problem is that the components and systems that make up the processes are not integrated. Therefore, to determine the changes

that caused the process update or what processes would be affected from these changes is difficult.

In the constructed ontology, the machine-readable representations of processes of the bank are included. This allows the process domain to query with logical expressions. With the help of querying whether there is any process that will be affected by the frequent changes in the processes can be easily determined. While reflecting the detected changes to processes, there may be opportunity to improve processes, or according to query results, new requirements may be detected. This may result in the domino effect.

Reliability and validity analysis of the developed ontology have been performed. Verification of the developed ontology is done via DL reasoner HermiT. According to the verification results of HermiT the ontology is consistent, complete and there is no redundancy.

The ontology has also been tested with the help of competency questions. One hundred and fifty different test cases have been tried and the results compared with the real world findings. According to the results, it was observed that ontology works correctly and completely.

Future Work

The future direction of our research study is the inclusion of all processes inherent in the bank system into the ontology. Once the processes are included, it is planned to develop an interface in which users can create automatic queries using natural language. And finally, after "the operational indicators project" that is currently being developed is finished, it is planned to include key performance indicators (KPI) to be identified, to the ontology. Thus, the scope of the domain will be expanded.

REFERENCES

- [1] OMG, “Business Process Model and Notation 2.0 (BPMN),” 2011.
- [2] H. J. Genrich and K. Lautenbach, “System modelling with high-level Petri nets,” *Theor. Comput. Sci.*, vol. 13, no. 1, pp. 109–135, 1981.
- [3] G. Keller, A.-W. Scheer, and M. Nüttgens, *Semantische Prozeßmodellierung auf der Grundlage" Ereignisgesteuerter Prozeßketten (EPK)"*. Inst. für Wirtschaftsinformatik, 1992.
- [4] O. M. G. A. Specification, “Unified Modeling Language Specification (version 2.1),” *Object Manag. Gr.*, vol. 70, 2007.
- [5] W. M. P. Van Der Aalst, “Business process management: a comprehensive survey,” *ISRN Softw. Eng.*, vol. 2013, 2013.
- [6] Carlos Pedrinaci and J. Domingue, “Towards an ontology for process monitoring and mining,” 2007.
- [7] A. L. Guido, A. Pandurino, and R. Paiano, “An Ontological Meta-model for Business Process Model and Notation (BPMN),” *Int. J. Bus. Res. Manag.*, vol. 7, no. 3, pp. 40–52, 2016.
- [8] M. Hepp, F. Leymann, J. Domingue, A. Wahler, and D. Fensel, “Semantic business process management: A vision towards using semantic web services for business process management,” in *IEEE International Conference on e-Business Engineering (ICEBE'05)*, 2005, pp. 535–540.

- [9] M. R. Genesereth and N. J. Nilsson, “Logical foundations of artificial,” *Intell. Morgan Kaufmann*, vol. 2, 1987.
- [10] T. R. Gruber, “A translation approach to portable ontology specifications,” *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, Jun. 1993.
- [11] W. N. Borst, “Construction of Engineering Ontologies for Knowledge Sharing and Reuse,” 1997.
- [12] R. Studer, V. R. Benjamins B’c, and D. Fensel, “Knowledge Engineering: Principles and methods,” *Data Knowl. Eng.*, vol. 25, pp. 161–197, 1998.
- [13] A. Haller, “Ontologies and semantic Business Process Management,” p. 55, 2007.
- [14] N. Guarino and P. Giaretta, “Ontologies and Knowledge Bases. Towards a Terminological Clarification,” in *Towards very large knowledge bases*, 1995, pp. 25–32.
- [15] B. Swartout, R. Patil, K. Knight, and T. Russ, “Toward Distributed Use of Large-Scale Ontologies,” *AAAI Symp. Ontol. Eng.*, 1997.
- [16] G. Schreiber, B. Wielinga, and W. Jansweijer, “The KACTUS view on the ‘O’ word,” in *IJCAI workshop on basic ontological issues in knowledge sharing*, 1995, pp. 159–168.
- [17] M. S. Fox, “The TOVE Project Towards a Common-Sense Model of the Enterprise,” in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 1992, pp. 25–34.
- [18] M. Uschold, M. King, S. Moralee, and Y. Zorgios, “The Enterprise

- Ontology,” *Knowl. Eng. Rev.*, vol. 13, no. 1, pp. 31–89, 1998.
- [19] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, “METHONTOLOGY: From Ontological Art Towards Ontological Engineering,” 1997.
- [20] R. Fikes, A. Farquhar, and J. Rice, “The Ontolingua Server: a Tool for Collaborative Construction,” *Comput. Sci. Dep. Stanford Univ.*, 1996.
- [21] A. Gangemi, G. Steve, and F. Giacomelli, “ONIONS: An ontological methodology for taxonomic knowledge integration,” in *ECAI-96 Workshop on Ontological Engineering, Budapest*, 1996, vol. 95.
- [22] R. Iqbal, M. A. A. Murad, A. Mustapha, and N. M. Sharef, “An analysis of ontology engineering methodologies: A literature review,” *Res. J. Appl. Sci. Eng. Technol.*, vol. 6, no. 16, pp. 2993–3000, 2013.
- [23] A. Gomez-Perez, “Ontological engineering: A state of the art,” vol. 2, 1999.
- [24] N. F. Noy and D. L. McGuinness, *Ontology Development 101 : A Guide to Creating Your First Ontology*, vol. 32. 2001, pp. 1–25.
- [25] M. Uschold, “Building Ontologies: Towards a Unified Methodology,” in *Proceedings of 16th Annual Conference of the British Computer Society Specialists Group on Expert Systems*, 1996.
- [26] M. Uschold and M. Gruninger, “Ontologies: principles, methods and applications,” *Knowl. Eng. Rev.*, vol. 11, no. 2, pp. 93–136, 1996.
- [27] O. Lassila and D. McGuinness, “The Role of Frame-Based Representation on the Semantic Web Introduction: Frame-based Representation Systems,” *Linköping Electron. Artic. Comput. Inf. Sci.*, vol. 6, no. 5, 2001.

- [28] M. Von Rosing, W. P. R. Laurier, and S. M. Polovina, "The value of ontology," *Comple. Bus. Process Handb.*, pp. 91–99, 2015.
- [29] N. Guarino, "Semantic matching: Formal ontological distinctions for information organization, extraction, and integration," in *International Summer School on Information Extraction*, 1997, pp. 139–170.
- [30] O. Corcho, M. Fernández-López, and A. Gómez-Pérez, "Methodologies, tools and languages for building ontologies. Where is their meeting point?," *Data Knowl. Eng.*, vol. 46, no. 1, pp. 41–64, 2003.
- [31] M. El Ghosh, H. Naja, H. Abdulrab, and M. Khalil, "Towards a middle-out approach for building legal domain reference ontology," *Int. J. Knowl. Eng.*, vol. 2, no. 3, pp. 109–114, 2016.
- [32] O. Corcho and A. Gómez-Pérez, "A roadmap to ontology specification languages," in *International Conference on Knowledge Engineering and Knowledge Management*, 2000, pp. 80–96.
- [33] T. Slimani, "Ontology development: A comparing study on tools, languages and formalisms," *Indian J. Sci. Technol.*, vol. 8, no. 24, pp. 1–12, 2015.
- [34] T. Bray, J. Paoli, C. M. Sperberg-McQueen, Y. Mailer, and F. Yergeau, "Extensible Markup Language (XML) 1.0 5th Edition, W3C recommendation, November 2008." ed.
- [35] F. Gandon and G. Schreiber, "RDF 1.1 XML Syntax: W3C Recommendation 25 February 2014," *World Wide Web Consortium. Available [https://www. w3.org/TR/rdfsyntax-grammar](https://www.w3.org/TR/rdfsyntax-grammar)*, 2014.
- [36] I. S. Ramoutsaki, "Ontology Based Semantic Annotation of Business

Processes with Semi-Automated Suggestions and Processing of Semantic Queries in Business Process Models,” 2016.

- [37] M. Krötzsch, F. Simancik, and I. Horrocks, “A Description Logic Primer,” 2012, no. June, pp. 1–17.
- [38] W. O. W. Group, “Formalizing knowledge by ontologies: Owl and kif,” *Relatório apresentado L’Istituto di Inform. e Telemat. (IIT). Cons. Naz. delle Ric. (CNR). Ital.*, 2008.
- [39] W3C OWL Working Group, “OWL 2 Web Ontology Language Document Overview (Second Edition),” *W3C Recommendation*, 2012. [Online]. Available: https://www.w3.org/2007/OWL/wiki/Document_Overview. [Accessed: 25-Apr-2019].
- [40] M. Boris, C. G. Bernardo, I. Horrocks, W. Zhe, F. Achille, and L. Carsten, “OWL 2 Web Ontology Language Profiles (Second Edition),” *W3C Recomm.*, no. December, pp. 1–43, 2012.
- [41] P. Hitzler and S. Rudolph, “OWL 2 Web Ontology Language Primer (Second Edition),” *W3C Recommendation*, 2012. [Online]. Available: <http://www.w3.org/TR/owl-primer>. [Accessed: 25-Apr-2019].
- [42] M. Sir, Z. Bradac, and P. Fiedler, “Ontology versus Database,” *IFAC-PapersOnLine*, vol. 48, no. 4, pp. 220–225, 2015.
- [43] E. Alatrash, “Using web tools for constructing an ontology of different natural languages.” 2015.
- [44] M. Horridge, S. Jupp, G. Moulton, A. Rector, R. Stevens, and C. Wroe, “A practical guide to building owl ontologies using protégé 4 and co-ode tools

- edition 1. 3,” *Univ. Manchester*, vol. 178, 2011.
- [45] K. Dentler, R. Cornet, A. Ten Teije, and N. De Keizer, “Comparison of reasoners for large ontologies in the OWL 2 EL profile,” *Semant. Web*, vol. 2, no. 2, pp. 71–87, 2011.
 - [46] S. Abburu, “A survey on ontology reasoners and comparison,” *Int. J. Comput. Appl.*, vol. 57, no. 17, 2012.
 - [47] K. Belhajjame and M. Brambilla, “Ontology-based description and discovery of business processes,” in *Enterprise, Business-Process and Information Systems Modeling*, 2009, pp. 85–98.
 - [48] M. Dumas, W. van der Aalst, and A. Ter Hofstede, *Process aware information systems*, vol. 1. Wiley Online Library, 2005.
 - [49] C. Pedrinaci, J. Domingue, and A. K. Alves De Medeiros, “A core ontology for business process analysis,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008.
 - [50] V. der Aalst, W. M. P., and H. M. Arthur, “Business process management: A survey,” *Lect. Notes Comput. Sci.*, vol. 2678, pp. 1–12, 2003.
 - [51] B. List and B. Korherr, “An evaluation of conceptual business process modelling languages,” in *Proceedings of the 2006 ACM symposium on Applied computing*, 2006, pp. 1532–1539.
 - [52] M. Rospocher, C. Ghidini, and L. Serafini, “An ontology for the Business Process Modelling Notation,” in *FOIS*, 2014, pp. 133–146.

- [53] I. Markovic and A. C. Pereira, “Towards a formal framework for reuse in business process modeling,” in *International Conference on Business Process Management*, 2007, pp. 484–495.
- [54] F. Smith and M. Proietti, “Rule based Behavioral Reasoning on Semantic Business Processes,” *Proc. 5th Int. Conf. Agents Artif. Intell. SciTePress Digit. Libr.*, 2013.
- [55] M. Ehrig, A. Koschmider, and A. Oberweis, “Measuring similarity between semantic business process models,” in *Proceedings of the fourth Asia-Pacific conference on Conceptual modelling-Volume 67*, 2007, pp. 71–80.
- [56] M. Born, F. Dörr, and I. Weber, “User-friendly semantic annotation in business process modeling,” in *International Conference on Web Information Systems Engineering*, 2007, pp. 260–271.
- [57] G. G. Petrova, A. F. Tuzovsky, and N. V. Aksenova, “Application of the Financial Industry Business Ontology (FIBO) for development of a financial organization ontology,” in *Journal of Physics: Conference Series*, 2017, vol. 803, no. 1, p. 12116.

APPENDIX A

RAW DATA TABLES

A.1 "Processes" page of the created summary document

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	
1	No	İndex	Adı	Durum	Yerleşim	Oluşturma Tarihi	Revizyon Tarihi	İşletim Tarihi	Birim Pozisyon	Müdürlük Kodu	Baskanlık Kodu	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama		
2	01.01.01	Müşteri Numarası Verme Süreci	İşletim	6	27.01.2011	28.09.2017	28.09.2017	28.09.2017	Sahip / Sorumlu /	M-005 / BHM-000 /	B-002 / / B-001, B-	S-003	RD-033, RD-021, RD-004	K-032, K-039, K-082, K-084, K-	U-012, U-017, U-018, U-049												
3	01.01.02	Hesap İşlemleri Süreci	Yürürlükte	1	06.11.2012	10.11.2014			Sahip / Sorumlu /	M-009 / BHM-000 /	B-001 / /	S-038, S-039, S-048	RD-181, RD-189	K-051, K-052, K-054, K-140	U-012, U-049												
4	01.01.03	Hesap İşlemleri Süreci	Yürürlükte	2	15.10.2009	14.09.2016			Sahip / Sorumlu /	M-002 / M-017 / BHM-000 /	B-004 / /	S-002, S-003	RD-128, RD-181	K-055, K-051, K-131	U-017, U-018, U-039, U-049												
5	01.01.04	Gek Karnesi Talebi, Basım ve Müşteri Süreci	Yürürlükte	1	09.11.2012	01.10.2015			Sahip / Sorumlu /	M-002 / BHM-000 /	B-001 / /	S-079	RD-017, RD-018, RD-020	K-013, K-022, K-046, K-052, K-	U-018, U-021, U-031, U-039, U-044, U-049												
6	01.02.08	Bankamız Çekleri Takas Projesi	Yürürlükte	1	09.11.2012	28.07.2017			Sahip / Sorumlu /	M-002 / BHM-000 /	B-001 / /	S-006		K-018, K-110	U-008, U-017, U-018, U-029, U-036, U-049												
7	01.02.11	Göç Yönetim İşleri Bilgi Giriş İşlemleri	Yürürlükte	1	09.11.2012	30.07.2015			Sahip / Sorumlu /	M-002 / BHM-000 /	B-001 / /	S-009		K-019, K-102	U-017, U-018, U-049												
8	01.02.17	Satış ve Alış İşlemleri Süreci	Yürürlükte	1	09.11.2012	16.03.2015			Sahip / Sorumlu /	M-002 / BHM-000 /	B-001 / /	S-002		K-019, K-102	U-017, U-018, U-049												
9	01.02.21	Kirak Kasa İşlemleri Süreci	Yürürlükte	2	09.11.2012	30.11.2015			Sahip / Sorumlu /	M-005 / BHM-000 /	B-002 / /	S-001		K-019, K-082, K-136	U-018, U-049												
10	01.02.22	Kurum Kira ve İşletim Süreci	Yürürlükte	2	09.11.2012	18.06.2016			Sahip / Sorumlu /	M-005 / BHM-000 /	B-002 / /	S-001		K-019, K-082, K-136	U-018, U-049												
11	01.02.28	Maas Odemeleri Süreci	Yürürlükte	3	09.11.2012	29.09.2016			Sahip / Sorumlu /	M-005 / BHM-000 /	B-003 / B-001	S-011		K-019, K-082, K-136	U-018, U-049												
12	01.01.09	Maas Hesaplaması Masraf Muafiyet Süreci	Yürürlükte	1	09.11.2012	03.02.2014			Sahip / Sorumlu /	M-010 / M-048	B-003 / B-015	S-011			U-001, U-021, U-044, U-050												
13	01.04.02	İzleme, Kontrol, Güçlendirme İşlemleri	Yürürlükte	6	27.01.2011	09.09.2016			Sahip / Sorumlu /	M-022 / BHM-000 /	B-009 / /	S-009		K-013, RD-128, RD-189	K-052, K-039, K-040, K-042, K-	U-012, U-017, U-018, U-029, U-036, U-049											
14	01.04.03	MAAS Bilgi ve Belgeler Talebi ve Alımı	Yürürlükte	4	09.11.2012	08.07.2016			Sahip / Sorumlu /	M-022 / BHM-000 /	B-001 / /	S-009		K-013, RD-128, RD-189	K-052, K-039, K-040, K-042, K-	U-012, U-017, U-018, U-029, U-036, U-049											
15	01.04.01	Değerlendirme ve Deneyselendirme Süreci	Yürürlükte	3	22.01.2009	18.07.2017			Sahip / Sorumlu /	M-017 / BHM-000 /	B-001 / /	S-015		K-013, RD-086, RD-112	K-097	U-017, U-049											
16	01.02.02	Ticarî Krediler Tahsis Süreci	Yürürlükte	4	22.01.2009	18.07.2017			Sahip / Sorumlu /	M-013, M-012, M-018	B-012, B-012, B-015	S-015		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
17	01.02.03	Kurumsal Kredî Tahsis Süreci	İşletim	2	22.01.2009	10.12.2014		30.12.2014	Sahip / Sorumlu /	M-013	B-013	S-017															
18	01.02.01	Terminalendirme Süreci	Yürürlükte	5	22.01.2009	18.11.2015			Sahip / Sorumlu /	M-013, M-012, M-018	B-012, B-012, B-015	S-017		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
19	01.02.03	İpotek Tesisi Süreci	Yürürlükte	3	22.01.2009	24.01.2017			Sahip / Sorumlu /	M-048 / BHM-000 /	B-015 / /	S-017		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
20	01.02.04	Kredi Üzerinden İndirimli Fiyatlandırma Süreci	Yürürlükte	3	22.01.2009	24.01.2017			Sahip / Sorumlu /	M-048 / BHM-000 /	B-015 / /	S-017		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
21	01.02.05	Bireysel Kredi Kullandırma Süreci	Yürürlükte	2	22.01.2009	27.12.2013			Sahip / Sorumlu /	M-013 / BHM-000 /	B-012 / /	S-021		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
22	01.02.03	Takvîm İşlemleri Kullandırma Süreci	Yürürlükte	1	20.12.2012	27.12.2013			Sahip / Sorumlu /	M-013 / BHM-000 /	B-012 / /	S-021		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
23	01.02.01	Ticarî Kredi Kullandırma Süreci	Yürürlükte	3	22.01.2009	10.12.2014			Sahip / Sorumlu /	M-013, M-012, M-018	B-012, B-012, B-015	S-017		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
24	01.02.01	Limit İşlemleri Tahsis Süreci	İşletim	4	22.01.2009	14.08.2016		30.09.2016	Sahip / Sorumlu /	M-013	B-013	S-017															
25	01.02.01	Terminatlandırma Süreci	Yürürlükte	6	22.01.2009	02.04.2018			Sahip / Sorumlu /	M-013, M-012, M-018	B-012, B-012, B-015	S-017		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
26	01.02.03	Aktarılan Bireysel Kredilerin Yedileme Süreci	Yürürlükte	8	22.01.2009	02.04.2018			Sahip / Sorumlu /	M-013, M-012, M-018	B-012, B-012, B-015	S-017		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
27	01.02.01	İdari Takip ve Kanunî Takip Değiş Süreci	Yürürlükte	3	22.01.2009	02.04.2018			Sahip / Sorumlu /	M-013, M-012, M-018	B-012, B-012, B-015	S-017		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
28	01.02.01	Takip Süreçlerinin Yürütülmesi Süreci	Yürürlükte	4	22.01.2009	02.04.2018			Sahip / Sorumlu /	M-013, M-012, M-018	B-012, B-012, B-015	S-017		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
29	01.02.08	YTS Operasyonları Süreci	Yürürlükte	1	20.12.2012	12.12.2014			Sahip / Sorumlu /	M-013	B-013	S-017															
30	01.02.10	Mevkuf İşlemlerinin Muhasebeleştirilmesi Süreci	Yürürlükte	1	16.11.2012	07.09.2015			Sahip / Sorumlu /	M-061 / BHM-000 /	B-019 / /	S-006		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
31	01.02.14	Elektronik Posta İşlemleri Süreci	Yürürlükte	2	16.11.2012	07.09.2015			Sahip / Sorumlu /	M-061 / BHM-000 /	B-019 / /	S-006		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
32	01.02.01	Vergi Beyan ve Ödeme Süreci	Yürürlükte	2	16.11.2012	08.09.2014			Sahip / Sorumlu /	M-041, M-045, M-046	B-019, B-019, B-019	S-004, S-030		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
33	01.02.01	Bağlı Şube Vergileri Yürütme Süreci	İşletim	2	16.11.2012	01.10.2015			Sahip / Sorumlu /	M-060 / M-063	B-019 / B-019	S-004, S-030		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
34	01.02.01	Personel Ödemeleri Süreci	Yürürlükte	2	16.11.2012	25.02.2016			Sahip / Sorumlu /	M-061, M-067	B-019, B-019 / /	S-003, S-030, S-034, S-079		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
35	01.02.02	Ödeme Tutarlarının Hükümetten İstenebilmesi Süreci	Yürürlükte	2	16.11.2012	23.01.2017			Sahip / Sorumlu /	M-061, M-067	B-019, B-019 / /	S-003, S-030, S-034, S-079		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
36	01.02.06	Paşpa Ödenen Kiraların Gözetilmesi Süreci	İşletim	2	16.11.2012	28.07.2017			Sahip / Sorumlu /	M-060	B-019	S-003, S-030, S-034, S-079		K-013, RD-041, RD-065	K-074, K-075, K-130	U-018, U-049											
			Süreci	Tanıtım	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	Referans Sıralama	

A.2 "Business Units" page of the created summary document

Başka	Baskanlık Adı	Müdürlük Kodu	Müdürlük Adı	Süreç Yetkilisi	Durum	Eski	Yeni
B-001	Bankacılık Operasyonları Baskanlığı	M-001	Bankacılık Hizmet Operasyonları Müdürlüğü	BARİŞ ÇELİK	aktif		
B-001	Bankacılık Operasyonları Baskanlığı	M-002	Cek Senet ve Eft Havaile Operasyonları Müdürlüğü	ALİ AYDINLIOĞLU	aktif		
B-001	Bankacılık Operasyonları Baskanlığı	M-003	Güvenli Bankacılık Müdürlüğü	UGUR TAŞDEMİR	aktif		
B-001	Bankacılık Operasyonları Baskanlığı	M-103	Merkezi Operasyon İşlemleri Müdürlüğü		pasif		
B-001	Bankacılık Operasyonları Baskanlığı	M-004	Merkezi Operasyon Yönetim Müdürlüğü	NAGİHAN DEMİRBAŞ	aktif		
B-016	Bireysel Bankacılık Pazarlama Baskanlığı	M-050	Bireysel Bankacılık Pazarlama Müdürlüğü	SÜLEYMAN ÖZDERE	aktif		
B-016	Bireysel Bankacılık Pazarlama Baskanlığı	M-051	Konut Finansman Müdürlüğü	CELAL ÖZTÜRK	aktif		
B-003	Bireysel Bankacılık Pazarlama Hizmetleri Baskanlığı	M-009	Bireysel Bankacılık Satış Koordinasyon Müdürlüğü		aktif		
B-003	Bireysel Bankacılık Pazarlama Hizmetleri Baskanlığı	M-008	Bireysel Müşteri Yönetim Müdürlüğü	GÖNÜL ÇETİN	aktif		
B-003	Bireysel Bankacılık Pazarlama Hizmetleri Baskanlığı	M-010	Maas Odemeleri Müdürlüğü	TUĞBA BİLGİN USLU	aktif		
B-012	Bireysel Bankacılık Kredileri Baskanlığı	M-030	Bireysel Krediler Müdürlüğü	CENGİZHAN ÇEBECİ	aktif		
B-012	Bireysel Bankacılık Kredileri Baskanlığı	M-031	Kobî Krediler Müdürlüğü	ÇİĞDEM ARDA MERAL	aktif		
B-012	Bireysel Bankacılık Kredileri Baskanlığı	M-032	Tarım Krediler Müdürlüğü	YAVUZ ERALP KARADENİZ	aktif		
B-023	Bireysel Bankacılık Kredileri Baskanlığı	M-033	Toplu Konut ve Bireysel Krediler Müdürlüğü		pasif		
B-023	Dagitim Kanalları Baskanlığı	M-073	Atm Müdürlüğü	NİLAY GÜRSER	aktif		
B-023	Dagitim Kanalları Baskanlığı	M-074	Çagiri Merkezi Müdürlüğü	MUSTAFA BEKTAŞ	aktif	M-102	
B-023	Dagitim Kanalları Baskanlığı	M-075	İnternet Bankacılığı Müdürlüğü	MURAT COŞKUN	aktif	M-102	
B-023	Dagitim Kanalları Baskanlığı	M-076	Sube Kuruluş ve Gelir İsteme Müdürlüğü	HANDAN GÜMÜŞ	aktif		
B-023	Dagitim Kanalları Baskanlığı	M-102	Telefon ve İnternet Bankacılığı Müdürlüğü		pasif		M-074M-075
B-011	Degerlendirme ve Derecelendirme Baskanlığı	M-029	Bolge Finansal Degerlendirme Koordinasyon Müdürlüğü		aktif		
B-011	Degerlendirme ve Derecelendirme Baskanlığı	M-027	Kurumsal Ticari ve Kobî Derecelendirme Sistemleri Müdürlüğü		aktif		
B-011	Degerlendirme ve Derecelendirme Baskanlığı	M-028	Perakende ve Bireysel Derce Sistemleri Müdürlüğü		aktif		
B-021	Emekli ve Saglik Yardim Sandigi Baskanligi	M-060	Oze Sosyal Guvenlik Vakfi Müdürlüğü	SELDİ KOÇAK	aktif		
B-019	Genel Muhasebe ve Mali İşler Baskanlığı	M-069	Mali Kontrol Müdürlüğü	COŞKUN MEMİÇ	aktif		
B-019	Genel Muhasebe ve Mali İşler Baskanlığı	M-061	Mali Operasyon Müdürlüğü	SÜLEYMAN ÖZKAN	aktif		
B-019	Genel Muhasebe ve Mali İşler Baskanlığı	M-062	Mali Raporlama Müdürlüğü	İŞİL YILDIRIM	aktif		
B-019	Genel Muhasebe ve Mali İşler Baskanlığı	M-063	Vergi ve Mali Mevzuat Yönetim Müdürlüğü	NEVAL COŞKUN	aktif		
B-019	Genel Muhasebe ve Mali İşler Baskanlığı	M-064	Yasal Raporlama ve Mali İşler Müdürlüğü	ASLI UZUNKÖPRÜ	aktif		
B-026	Hazine Baskanlığı	M-084	Aktif Pasif Yönetim Müdürlüğü	YELDA AYKUT	aktif		
B-026	Hazine Baskanlığı	M-085	Doviz ve Turev İşlemleri Müdürlüğü	MFHMET ALI AYKILIT	aktif		

A.3 "Definitions" page of the created summary document

Kısaltma No	Kısaltma	Açıklaması	TİP
K-001	AcikPozisyon	Taraflar arasında vadesi gelmemiş bir işlemin var olduğunu	TANIM
K-002	Adk	Alternatif Dağıtım Kanalları	TANIM
K-003	Aps	Adres Paylaşım Sistemi	TANIM
K-006	Atmog	ATM Operasyon Görevlisi	ROL
K-007	Atmok	ATM Operasyon Koordinatörü	ROL
K-004	Atms	ATM Sorumlusu	ROL
K-005	AtmStatusListesi	Alarm ve/veya kamera takılı olan ATM' lerin kaydının tutulduğu listedir.	TANIM
K-008	Bb	Bireysel Bankacılık	TANIM
K-009	Bbsk	Bireysel Bankacılık Satış Koordinasyon	TANIM
K-010	Bbsyt	Bireysel Bankacılık Satış Yetkilisi. Sahada emlakçı kanallı konut kredisi pazarlayan saha ekibinin lideri	ROL
	Bddk	Bankacılık düzenleme ve denetme kurumu	TANIM
K-011	Bds	Başvuru Değerlendirme Sistemi	UYGULAMA ARACI
K-012	Bist	Borsa İstanbul AŞ	TANIM
K-013	BlokeCek	Çek bedelinin bloke paralar hesabına alınarak ve keşidecinin çekin ön yüzünü imzalaması suretiyle düzenlenen çek.	TANIM
K-014	Bloomberg	Anlık piyasa bilgilerinin (fiyat/oran) çekildiği, fiyatlama, raporlama ve değerleme amaçlı da kullanılan lisanslı data programı	UYGULAMA ARACI
K-015	Bobsy	Bölge Bireysel Satış Yöneticisi. Her bölgede en az 1 kişi olarak belirlenmiş, satış ve pazarlama stratejilerini kendilerine ait Şubelerdeki Bireysel Satış Grubu Personeline aktaran, saha çalışmalarında şubelere destek veren Bölge Müdürlüğü personeli	ROL
K-016	Bpy	Bireysel Portföy Yöneticisi	ROL
K-017	Brief	Kampanya veya etkinlik detaylarının ve mesajların doğru bir ifade ile işi gerçekleştirecek olan tarafa aktarılmasını sağlayan bilgilendirme metni	TANIM
K-018	Btom	Bankalararası Takas Odaları Merkezi	UYGULAMA ARACI
K-019	Buhaps	Bireysel Ürün ve Hizmet Paket Sözleşmesi	DOKÜMAN
K-022	Cek	Türk Ticaret Kanunu hükümlerine göre düzenlenmiş bir ödeme aracıdır.	TANIM
K-020	CitFirmasi	Para taşıma, ATM FLN müdahale, ATM SLM eskortluk hizmetinin alındığı firmadır.	TANIM
K-023	CizgiAltıReklam	Mailing, afiş, broşür, ürün sunum materyalleri vb. ürünler	TANIM
K-024	CizgiUstuReklam	Gazete, dergi, TV, radyo, açık hava vb. mecralarda kullanılan ürünler	TANIM
K-025	Cm	Çağrı Merkezi	TANIM
	Crm	Müşteri İlişkileri Yönetimi	TANIM

A.4 "Reference processes" page of the created summary document

No	Referans Veren	Referans Veren Süreci Adı	Durum	No	Referans Alan	Referans Alan Süreci Adı	Kapsamda mı?	Durum
S-001	01.01.01	Müşteri Numarası Verme Süreci	iptal	01.01.12	Toplu Mevduat Hesabı Açma Süreci	Hayır		
S-001	01.01.01	Müşteri Numarası Verme Süreci	iptal	01.01.13	Toplu Kredi Kartı Verme, Kredi Bankomat, Yatırım Hesabı, Havuz Hesabı, Döviz Hesabı Açma Süreci	Hayır		
S-001	01.01.01	Müşteri Numarası Verme Süreci	iptal	09.01.08	Yazaklı ve Tuzak Telefon Liste Yönetimi Süreci	Evet	Yürürlükte	
S-001	01.01.01	Müşteri Numarası Verme Süreci	iptal	01.03.31	Mobil Saha Satış Uygulaması Süreci	Hayır		
S-002	01.01.18	Havale İşlemleri Süreci	Yürürlükte	01.01.18	Zaman Ajansına Üçüyaş Değerlerin Devri Süreci	Hayır		
S-002	01.01.18	Havale İşlemleri Süreci	Yürürlükte	01.02.02	Bankacılık Uygulama ve İzlenimi İyileştirme Süreci	Hayır		
S-002	01.01.18	Havale İşlemleri Süreci	Yürürlükte	07.01.02	Merkezi EFT/Havale/İca Taltmatan Süreci	Evet	Yürürlükte	
S-002	01.01.18	Havale İşlemleri Süreci	Yürürlükte	04.02.01	İnternet Bankacılığı Tanımlama, Güncelleme ve İptal Süreci	Evet	Yürürlükte	
S-002	01.01.18	Havale İşlemleri Süreci	Yürürlükte	04.03.01	Çağrı Karşılama Süreci	Evet	Yürürlükte	
S-003	01.01.22	Haciz İşlemleri Süreci	Yürürlükte	01.01.18	Havale İşlemleri Süreci	Evet	Yürürlükte	
S-003	01.01.22	Haciz İşlemleri Süreci	Yürürlükte	07.01.01	Gübe Göden Eft Süreci	Evet	İptal	
S-004	01.02.04	Çek Karnesi Talebi, Basımı ve Müşteriye Teslimi Süreci	Yürürlükte	09.12.06	Fatura Ödeme Süreci	Evet	Yürürlükte	
S-005	01.02.06	Bankamız Çekleri Takas Provizyon Süreci	Yürürlükte	01.02.11	Şüpheli Çek Uyan Bilgisi Giriş İşlemleri Süreci	Evet	Yürürlükte	
S-005	01.02.06	Bankamız Çekleri Takas Provizyon Süreci	Yürürlükte	01.02.10	Ödeme Yazaklı Çeklerin Bankamız Sistemine Tanımlanması Süreci	Hayır		
S-006	01.02.11	Şüpheli Çek Uyan Bilgisi Giriş İşlemleri Süreci	Yürürlükte	01.02.09	TL-VP Bankamız Çekleri Ödeme İşlemleri Süreci	Hayır		
S-006	01.02.11	Şüpheli Çek Uyan Bilgisi Giriş İşlemleri Süreci	Yürürlükte	01.02.06	Bankamız Çekleri Takas Provizyon Süreci	Evet	Yürürlükte	
S-008	01.02.21	Kırsak Kasa İşlemleri Süreci	Yürürlükte	01.01.01	Müşteri Numarası Verme Süreci	Evet	İptal	
S-008	01.02.21	Kırsak Kasa İşlemleri Süreci	Yürürlükte	01.01.11	Mevduat Hesabı Açma Süreci	Hayır		
S-008	01.02.21	Kırsak Kasa İşlemleri Süreci	Yürürlükte	01.01.16	Zaman Ajansına Üçüyaş Değerlerin Devri Süreci	Hayır		
S-009	01.03.01	Kurum Tahvilat İşlemleri Süreci	Yürürlükte	04.02.01	İnternet Bankacılığı Tanımlama, Güncelleme ve İptal Süreci	Evet	Yürürlükte	
S-009	01.03.01	Kurum Tahvilat İşlemleri Süreci	Yürürlükte	04.03.01	Çağrı Karşılama Süreci	Evet	Yürürlükte	
S-010	01.03.08	Maas Ödemeleri Süreci	Yürürlükte	01.01.11	Mevduat Hesabı Açma Süreci	Hayır		
S-010	01.03.08	Maas Ödemeleri Süreci	Yürürlükte	01.01.13	Toplu Mevduat Hesabı Açma Süreci	Hayır		
S-010	01.03.08	Maas Ödemeleri Süreci	Yürürlükte	01.03.07	DTO Firma Tanımlama/Bilgi Güncelleme Süreci	Hayır		
S-010	01.03.08	Maas Ödemeleri Süreci	Yürürlükte	01.03.09	Maas Hesaplarına Masraf Muafiyeti Tanımlanması ve İzlenmesi Süreci	Evet	Yürürlükte	
S-010	01.03.08	Maas Ödemeleri Süreci	Yürürlükte	05.04.08	Promosyon İşlemleri Süreci	Hayır		
S-010	01.03.08	Maas Ödemeleri Süreci	Yürürlükte	05.01.10	Debit Kart Talep, Basım, Teslim ve İmha Süreci	Hayır		
S-010	01.03.08	Maas Ödemeleri Süreci	Yürürlükte	09.02.03	Takip Dış Davaların İzlenmesi Süreci	Hayır		
S-011	01.03.09	Maas Hesaplarına Masraf Muafiyeti Tanımlanması ve İzlenmesi Süreci	Yürürlükte	01.03.09	Maas Hesaplarına Masraf Muafiyeti Tanımlanması ve İzlenmesi Süreci	Evet	Yürürlükte	
S-011	01.03.09	Maas Hesaplarına Masraf Muafiyeti Tanımlanması ve İzlenmesi Süreci	Yürürlükte	05.01.12	Kredi Kartı Bilgi Güncelleme ve Parametre Değişiklik Süreci	Hayır		
S-014	01.01.01	Değerlendirme ve Değerlendirme Süreci	Yürürlükte	01.02.02	Ticarit Krediler Tahsis Süreci	Evet	Yürürlükte	
S-015	01.02.02	Ticarit Krediler Tahsis Süreci	Yürürlükte	02.01.01	Değerlendirme ve Değerlendirme Süreci	Evet	Yürürlükte	
S-015	01.02.02	Ticarit Krediler Tahsis Süreci	Yürürlükte	02.02.01	Teminatlandırma Süreci	Evet	Yürürlükte	
S-017	01.03.01	Teminatlandırma Süreci	Yürürlükte	01.02.05	Tahvil ve Teminata Alınan Çeklerin Bankamız Sistemine Giriş İşlemleri Süreci	Hayır		
S-017	01.03.01	Teminatlandırma Süreci	Yürürlükte	01.02.17	Senet Giriş İşlemleri Süreci	Evet	Yürürlükte	

A.5 "Reference documents" page of the created summary document

Referans Doküman No	Türü	RefensdokumanAdi
RD-001	Yönetim Kurulu Kararı	06/04/2011TarihVe81745SayılıYönetimKuruluKararı
RD-002	Makam Onayı	07/03/2014Tarih565SayılıMakamOnayı
RD-003	Yönetim Kurulu Kararı	12/01/2012TarihVe82839SayılıYönetimKuruluKararı
RD-004	Tebliğ	2003/950SayılıTicariİşletmeRehniHakkındaTebliğ
RD-005	Tebliğ	2005/229SayılıDovizAlışSatisİşlemleriDengeKurlarıHakkındaTebliğ
RD-010	Tebliğ	2007/1503SayılıKonutFinansmanıVelpotekKarsılığıTikResmiSenetleriHakkındaTebliğ
RD-011	Tebliğ	2007/330SayılıİpotekSenetleriHakkındaTebliğ
RD-013	Tebliğ	2008/1251SayılıSmsVeWebdenKrediKartıBasvurusuHakkındaTebliğ
RD-014	Tebliğ	2008113SayılıSkorkartVeGenelKrediUygulamalarıHakkındaTebliğ
RD-015	Makam Onayı	2009/2111SayılıMakamOnayı
RD-016	Tebliğ	2009/383SayılıDovizVeEftEtkiliAlımSatımİşlemlerininMerkezileştirilmesiHakkındaTebliğ
RD-017	Genelge	2009/40SayılıMerkeziÇekProjesininUygulamayaAlınmasıHakkındaGenelge
RD-018	Genelge	2009/52SayılıÇekLimitiTespitYetkileriHakkındaGenelge
RD-019	Genelge	2009/6SayılıMerkeziEftHavaleUygulamasıHakkındaGenelge
RD-020	Tebliğ	2009/96SayılıHologramBaskılıYeniÇekÖrnekleriHakkındaTebliğ
RD-021	Tebliğ	2010/1069SayılıAracRehinSözleşmesiHakkındaTebliğ
RD-022	Tebliğ	2010/1278SayılıİadeEdilecekHavalelerHakkındaTebliğ
RD-023	Genelge	2010/132SayılıElektronikOrtamdaAracRehniTesisihakkındaGenelge

A.6 "Application Tools" page of the created summary document

Uygulama/İletişim Araç No	Uygulama/İletişimArac	İç/Dış
UI-001	BankSoft	İç
UI-002	BankVb	İç
UI-003	Bds	İç
UI-004	BilgiSistemi	İç
UI-005	BistOtasTerminalleri	Dış
UI-006	Bloomberg	Dış
UI-007	BpyPortali	İç
UI-008	Btom	Dış
UI-010	Dto	İç
UI-011	DurumPortali	İç
UI-012	Dys	Dış
UI-013	EBeyannameUygulaması	Dış
UI-014	EFaturaPortali	Dış
UI-015	EihaleSistemi	Dış
UI-016	EmlakTakipPortali	Dış
UI-017	EPosta	Dış
UI-018	Ets	Dış
UI-019	Faks	Dış
UI-020	Fidelity	Dış
UI-021	Ftp	Dış
UI-022	Grc	Dış
UI-023	GüvenliBankacılıkIntranetUygulaması	İç
UI-024	Gvp	Dış
UI-025	Host	İç
UI-026	Ivr	Dış
UI-027	Sabas	Dış
UI-028	Kondor	Dış
UI-029	MasakOnline	Dış
UI-031	MerkeziÇekPortali	İç
UI-032	Mkk	Dış
UI-033	Mks	Dış

APPENDIX B

TURKISH VERSION OF QUESTION SET

D.1 Turkish Competency Questions

NO	SORULAR
1	Ana süreçler nelerdir?
2	01.Mevduat ana sürecinin alt süreçleri nelerdir?
3	01.Mevduat ana sürecinin altındaki, iptal veya yürürlükte durumunda olan süreçleri?
4	01.01.Hesapİslemleri alt sürecinin, iptal veya yürürlükte durumunda olan süreçleri?
5	01.01.01.MusteriNumarasıVermeSüreci'nin durumu nedir?
6	01.01.01.MusteriNumarasıVermeSüreci'nin oluşturulma tarihi nedir?
7	01.01.01.MusteriNumarasıVermeSüreci'nin revizyon tarihi nedir?
8	01.01.01.MusteriNumarasıVermeSüreci'nin iptal tarihi nedir?
9	01.01.01.MusteriNumarasıVermeSüreci'nin versiyon numarası nedir?
10	01.01.01.MusteriNumarasıVermeSüreci'nde kullanılan uygulama araçları nelerdir?
11	01.01.01.MusteriNumarasıVermeSüreci'nin referans verdiği dokümanlar nelerdir?
12	01.01.01.MusteriNumarasıVermeSüreci'ndeki tanımlar nelerdir?
13	01.01.01.MusteriNumarasıVermeSüreci'nde yer alan roller nelerdir?
14	01.01.01.MusteriNumarasıVermeSüreci'ne referans veren süreçler nelerdir?
15	01.01.01.MusteriNumarasıVermeSüreci'nin referans verdiği süreçler nelerdir?
16	01.01.01.MusteriNumarasıVermeSüreci'nin süreç sahibi iş birimleri nelerdir?
17	MusteriVeHesapOperasyonlarıMudurluğu'nun sahibi olduğu süreçler nelerdir?
18	BankacılıkOperasyonlarıBaskanlığı'nın sahibi olduğu süreçler nelerdir?
19	Opk rolü içeren, durumu yürürlükte olan süreçler nelerdir?
20	Ets uygulama aracını içeren, durumu yürürlükte olan süreçler nelerdir?
21	6415 Sayılı Terörizmin Finansmanının Önlenmesi Hakkında Kanun'u içeren, durumu yürürlükte olan süreçler?
22	Masak tanımını içeren, durumu yürürlükte olan süreçler?
23	İptal olmuş süreçlere referans veren süreçler nelerdir?
24	Merkezileşme anahtar kelimesine referans veren bireyler nelerdir?
25	Yürürlükte olan, Versiyon numarası sıfır ve revizyon tarihi 2017 den eski olan süreçler nelerdir?
26	Eski birim adlarına referans veren süreçler nelerdir?
27	Eski süreç adlarına referans veren süreçler nelerdir?