

**A GENERIC ONTOLOGY CREATION TOOL: A CASE STUDY ON
BUSINESS SECTORS**



A DOCTOR OF PHILOSOPHY (PhD) THESIS

in

Modeling and Design of Engineering Systems

(Main Fields: Computer Engineering and Information Systems Engineering)

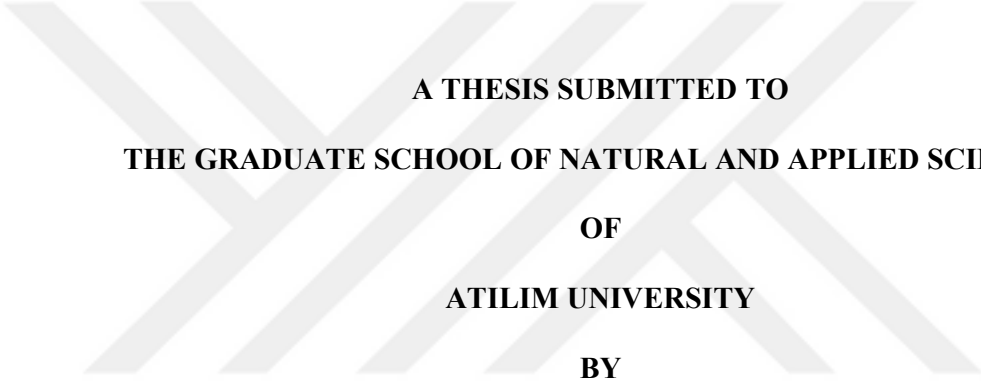
Atılım University

by

EKREM AĐLAR YILMAZ

JULY 2017

**A GENERIC ONTOLOGY CREATION TOOL: A CASE STUDY ON
BUSINESS SECTORS**



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

EKREM AĐLAR YILMAZ

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF**

DOCTOR OF PHILOSOPHY

IN

MODELLING AND DESIGN OF ENGINEERING SYSTEMS

JULY 2017

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 Doctor of Philosophy.

Prof. Dr. Abdulkadir Erden

Program Chair

This is to certify that we have read the thesis “A GENERIC ONTOLOGY CREATION TOOL: A CASE STUDY ON BUSINESS SECTORS” submitted by “Ekrem Çağlar Yılmaz” and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Assoc. Prof.Dr. Cenk Güray

Co-Supervisor

Asst. Prof. Dr. Çiğdem Turhan

Supervisor

Examining Committee Members

Asst. Prof. Dr. Çiğdem Turhan

Asst. Prof. Dr. Güzin Tirkeş

Asst. Prof. Dr. Hasan Umut Akın

Asst. Prof. Dr. Erol Özçelik

Asst. Prof. Dr. Atilla Bostan

Date: 14.07.2017



I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Ekrem Çağlar Yılmaz

Signature:

ABSTRACT

A GENERIC ONTOLOGY CREATION TOOL: A CASE STUDY ON BUSINESS SECTORS

Yılmaz, Ekrem Çağlar

PhD., Modelling and Design of Engineering Systems

Supervisor: Asst.Prof.Dr. Çiğdem Turhan

Co-Supervisor: Assoc.Prof.Dr. Cenk Güray

July 2017, 117 pages

To retrieve any information from the Web, a search has to be performed on billions of documents which are unorganized, unstructured and unreadable by machines. To overcome this problem, the data on the Web has to be formalized in a machine readable format. One of the solutions is to use the Semantic Web technology which provides structure and meaning to data on the Web. In order to provide machine readable and semantically identified information, the Semantic Web technology utilizes ontologies which include resources, properties and their relations to identify metadata about data. The current ontology editors require expertise to create, organize, edit and manage ontologies. In this study, a generic ontology creation tool is developed for users with no expertise in ontology creation. The tool which can be easily and effectively used at every level of a business gathers information about the ontology from a non-expert providing step by step guidance with user interfaces. The aim is to enable any employee of a firm to create an ontology in their domain to be able to share information in machine-readable form with the rest of his/her company or other companies. The tool is tested on users who have different working experiences in terms of years and sectors of business. The results are evaluated with statistical methods which show that on the average, the users are satisfied with the tool and are able to create ontologies in their own domains.

Keywords: Semantic Web, Ontology, Ontology Creation

ÖZ

GENEL AMAÇLI ONTOLOJİ OLUŞTURMA ARACI:

İŞ SEKTÖRLERİNE YÖNELİK UYGULAMA

Yılmaz, Ekrem Çağlar

Doktora, Mühendislik Sistemlerinin Modellenmesi ve Tasarımı

Tez Yöneticisi: Yrd.Doç. Dr. Çiğdem Turhan

Ortak Tez Yöneticisi: Doç.Dr. Cenk Güray

Temmuz, 2017, 117 sayfa

Ağ üzerinde bilgi aramak için, makinelerin milyarlarca sayıda olan düzensiz ve yapısız dökümanı araştırması gerekmektedir. Bu problemi çözmek için ağ üzerindeki veriler yeniden yapılandırılmalıdır. Bu probleme getirilecek çözümlerden bir tanesi de yapılandırılmış ve anlamlandırılmış veri yapısı sunan Anlamsal Ağ teknolojisidir. Anlamsal ağ, makine-okunabilir ve anlamsal olarak tanımlı bilgiyi sunabilmek için ontolojileri kullanmaktadır. Ontoloji oluşturmak, revize etmek ve yapılandırmak için kullanılan mevcut ontoloji biçimleyicileri uzmanlık gerektirmektedir. Bu çalışma ile, ontoloji oluşturma uzmanlığı olmayan kişiler için genel amaçlı kullanılabilen ontoloji oluşturma aracı geliştirilmiştir. Araç, adım adım ilerleyen kullanıcı arayüzü ile, kullanıcıları yönlendirerek kendi alanlarına özgü ontolojileri oluşturmalarını sağlamaktadır. Amaç, ontoloji oluşturma uzmanlığı olmayan kişilerin, kendi alanlarına uygun ontolojileri oluşturup, mevcut çalıştıkları firma içi veya iş sektörleri arasında makine-okunabilir şekilde bilgi paylaşımı yapabilmeleridir. Araç, farklı iş sektörlerinden farklı sürede çalışma deneyimine sahip kişiler tarafından test edilmiştir. İstatistiksel olarak uygun yöntemler ile değerlendirilen sonuçlar göstermektedir ki; teste katılan kişiler ortalama ve üzerinde memnuniyet bildirmişlerdir ve kendi alanlarında ontoloji oluşturabilmişlerdir.

Anahtar Kelimeler: Anlamsal Ağ, Ontoloji, Ontoloji Yaratma



To My Parents, Nejat-Nuray Yılmaz

ACKNOWLEDGMENTS

I express sincere appreciation to my supervisor Asst.Prof.Dr. ıgdem Turhan for her guidance and insight throughout the research. Thanks also go to my co-supervisor Assoc. Prof. Dr. Cenk Gray. I am also grateful to Asst.Prof.Dr. Gzin Tirkeş for her valuable assistance. To my wife, Tuęba, and my son, Kerem Nejat, I offer sincere thanks for their continuous support and patience during this period.

TABLE OF CONTENTS

ABSTRACT	iii
ÖZ	iv
DEDICATION.....	v
ACKNOWLEDGMENTS	vi
TABLE OF CONTENTS	vii
LIST OF TABLES	ix
LIST OF FIGURES	xi
CHAPTER	
1. INTRODUCTION	1
Problem Statement	2
Organization of the Thesis	2
2. LITERATURE REVIEW	4
2.1 SEMANTIC WEB	4
2.1.1 Semantic Web Applications.....	6
2.1.2 Search Engines for the Web and E-commerce applications.	8
2.2 ONTOLOGY	
2.2.1 Ontology Essentials and Development Tools for S.W.....	10

2.2.2 RDF and the Semantic Web technology.....	12
2.2.3 Primary Ontology Creation Tools.....	14
2.2.4 The S.W. technology and Ontology usage in ERP.....	18
2.2.5 Semantic Web Applications in ERP.....	20
3. DESIGN and IMPLEMENTATION	24
3.1 Design of System	24
3.2 Technical Specifications.....	25
3.3 System interfaces and capabilities.....	25
3.4 Case Study: Project Ontology Creation.....	30
4. EVALUATION	34
4.1 Purpose of Questionnaire	34
4.2 Questionnaire and Participant’s Results.....	37
4.3. Statistical Analysis Steps and Inferences of Questionnaire Results.....	40
4.4 Discussion.....	45
5. CONCLUSION.....	51
REFERENCES.....	54
APPENDICES.....	59
A. Histograms of questionnaire questions of Frequency testing.....	60
B. Programing Code of Tool	69
C. Statistical Analysis Results	86

LIST OF TABLES

TABLE

1. Participant Profiles.....	36
2. Questionnaire Form.....	38
3. Individual Comparisons.....	43
4. Working Experience Groups' Comparisons for "Understandability".....	44
5. Working Experience Groups' Comparisons for "Usability".....	45
6. Working Experience Groups' Comparisons for "Consistency".....	46
7. Working Experience Groups' Comparisons for "Ease-of-use".....	47
8. Working Experience Groups' Comparisons for "Productivity".....	48

LIST OF FIGURES

FIGURES

1. Semantic Web Framework.....	14
2. RDF Graph Example.....	16
3. RDF Graph creation example code in Jena.....	16
4. RDF Graph creation example 2 in Jena.....	17
5. RDF Graph creation example 2 code in Jena.....	18
6. The Components of the System.....	24
7. Starting Window.....	25
8. Help Menu Content.....	26
9. Add Resource window.....	27
10. Resource Addition Feedback.....	27
11. Setting a relation between resources.....	28
12. Relation Control Property.....	28
13. Generated RDF output on Textbox.....	29
14. RDF graph representations of a Project.....	31
15. Ontology Resource.....	31
16. Resource Addition Window.....	32
17. Relation setting between Resources and Control.....	32
18. RDF output of the created ontology.....	33
19. RDF of Project Ontology.....	35
20. 1-5 years and 6-10 years Test Results.....	44
21. 6-10 years and 10 years and above Test Results.....	46
22. 1-5 years and 10 years and above Test Results.....	47



CHAPTER 1

INTRODUCTION

In the Internet, there are billions of documents which are used for information retrieval by businesses and people. When a person or a firm tries to reach the required information, retrieving information from these uncountable number of documents using search engines such as Google, Yandex, etc. may cause many problems such as retrieving irrelevant documents or information due to unorganized, not related and outdated information on the Web (Booth et al., 2004). Nowadays, one of the platforms used for effective information retrieval is called the Semantic Web. In Semantic Web technology, the contents of the current web documents are organized into semantic representations and information retrieval is much more simpler than the current data-documentation approach where the results are more relevant with the requested information (Davies et al., 2006). The most relevant search results are obtained from the Semantic Web approach due the structure of the Semantic web, since the Semantic Web holds the metadata about the information stored on the web (Baader et al., 2005).

The Semantic Web technology provides semantically-understandable and semantically-relevant results when a search for an information about a subject or in a document is needed (Berners-Lee et al., 2001). To provide this technology in the Semantic Web, the information is identified with the meaning i.e. relationships between data objects, etc. to provide a machine-readable format which makes it possible to process this information with computers (Buhler & Vidal, 2005; Ding et al., 2002).

To provide machine-readable and semantically identified information, the Semantic Web utilizes ontologies (Gibbins et al., 2003; Hendler, 2001). The ontology approach provides data organization in terms of classes, hierarchy of these classes, and the properties of the classes as data members similar to the object-oriented approach

(Haugen & McCarthy W.E., 2000). Also, the ontology provides a declaration of relationships between the objects that are identified in the ontology.

All of these properties of the ontology provide a more declarative and easy information retrieval property to the Semantic Web technology (Lin et al., 2004). There are some popular ontology creation tools and editors which have been created as the Semantic Web technology developed, such as; Protégé (Protégé, 2017), NeOn toolkit (NeOn, 2017), etc. These ontology editors require expertise to create, organize, edit and manage the required and currently used ontologies.

Problem Statement:

In the real world, there are many entities and relationships that describe the essentials of a business. The entities are interconnected physically and/or semantically. Most of the semantic meanings are described by humans who are experts in a domain (Lin & Harding, 2007). Therefore, the modeling of entities and their relationships may vary from person to person causing problems such as inefficient structuring of entity and relationships (Fox et al., 1998). The integration of Ontology structure that is data with semantic meaning may offer effective support to the business world in terms of time and budget, since; the human-processed information is not economical (Grubic & Fan, 2010). For the requirement of structuring data in the form of an ontology, there exist ontology creation tools, which have become popular such as Protégé, OntoLingua etc. which require ontology experts with professional knowledge to create the ontology.

In this thesis, a generic ontology creation tool will be developed for the users who do not have any ontology expertise. The tool which can be easily and effectively used at every level of a business gathers information about the ontology from a non-expert providing step by step guidance with the user interfaces.

The research questions for the thesis are stated as below:

Can a generic ontology creation tool be used by users with no ontology creation expertise?

Can a generic ontology creation tool be used to develop ontologies in different domains?

Organization of the Thesis:

The organization of the thesis is as follows:

Chapter 2 provides background information about the Semantic Web and ontologies. Moreover, Semantic Web essentials, ontology tools, and basics of an ontology are presented.

Chapter 3 presents information about the design and implementation of the system.

Chapter 4 includes the evaluation techniques and results of the evaluation of the overall system.

Chapter 5 presents the conclusion of the thesis with some possible future work.



CHAPTER 2

LITERATURE REVIEW

2.1 SEMANTIC WEB

With the development and growth of the World Wide Web, a vast amount of data has been provided to computers and end users of the internet. When we consider the amount of data and information sources, the task of managing and understanding the data becomes very difficult (Lyman & Varian, 2003). The users continuously need to deal with data in terms of sending, searching, receiving, managing the e-documents, e-catalogs, mails in different sectors of the business. To overcome these problems for novice users who try to reach the required, relevant and pure data, some efficient and practical tools are required (Studer et al., 1998). From another perspective, the integration of raw data is not a practical process for IT managers to formalize either (Koriat & Gelbard, 2014). The integration of data, requires a generic view about the data and also understanding the relationships between data objects through the context.

Since searching for data through the internet is a time-consuming process, the search engines for the Web have become useful tools when reaching a specific information. The principle behind the current search engines such as Google, Yandex, etc. is gathering data from pre-indexed web pages from the flash memory into the storage for each particular search page. Some of the problems with the keyword-based search is the ambiguity, because the irrelevant information through different concepts may result in the same results. For example, when a user tries to search for the keyword 'Jaguar' from the big-cat family, the resulting search list may include the big-cat family information and also the car producer 'Jaguar'. Since, this situation produces a particular case for that scenario, the user should continue to examine the resulting search list to get the relevant information (Kuter et al., 2005). This scenario is a time-

consuming process when a user tries to reach the relevant information through the served results. If we consider that the content of the Web in terms of document volume is rising each year, reaching the relevant or requested information becomes harder through the keyword-based searching tools.

For all of these time-consuming problems, a solution arises with the Semantic Web technology. Since main problem about the irrelevant search results are originated from the lack of semantic information about the data, the Semantic Web technology can provide a solution with structured data on the web (Berners-Lee & Hendler, 2001).

In early years, the World Wide Web was solely designed to provide an information space to computer users. On Web 1.0, the pages on the Web almost had a static mechanism between the years 1989 to 2005 (Choudry, 2014). The structure of the static Web pages mainly concentrated on publishing information rather than information exchange (Berners-Lee et al., 2001).

In the following years of the Web, Dale Dougherty introduced a new approach in 2004 which is known as Web 2.0. Web 2.0 identifies Web of data not only for human information consumption, but also that the information on the Web should be interpretable and manageable by humans and computers (Berners-Lee et al., 2001; Shafiq et al., 2006; KwangWook & Ravichandran, 2014). The introduced Human-centric web idea caused more linked data in terms of “relationships” which primes the essential idea behind the Semantic Web technology (Choudry, 2014).

The Semantic Web technology has been developed and introduced by Timothy John Berners-Lee in 2001 as an extension of the current Web which encodes the information with well-defined meanings which enables the cooperation between people and computers and the World Wide Web Consortium has recommended tools for the Semantic Web since 2004 (Berners-Lee et al., 2001; Paolucci & Sycara, 2003; W3C, 2017). W3C is the short name for World Wide Web Consortium. W3C is an international community which includes member organizations, full time staff

and the public, working together. The main purpose of the Consortium is defining Web standards and helping the Web to reach its full potential with the leadership of Tim Berners-Lee and CEO Jeffrey Jaffe (W3C, 2017).

The Semantic Web technology, with the new functionality of machines that originated from the machine-readable format, has increased the processing power of machines instead of only displaying the present data (Zhang, 2007). To achieve this functionality, the computers should be able to access the linked data as an information, and rules of inferences about the information to establish a reasoning (Anjomshoaa et al., 2006).

The Semantic Web technology has two different aspects in terms of people and computer networks. People with the Semantic Web technology can store data and query that data on the Web, also, create their own vocabulary and additionally create and use rules to handle the data on the Semantic Web (Jacob, 2003). The network computers can query and interpret the Web of data, that is restructured by the Semantic Web, to make inferences, because the data is identified, structured and are ruled with the definitions by the Semantic Web technology. The structured or linked data, provides a large scale integration of Web of data in a formal implementation. The machine-readable and processable data format for the Web of data, almost is realized as the redesign of the World Wide Web in terms of the infrastructure of the original Web (Blacoe & Portabella, 2005).

The Semantic Web technology stores information i.e the data objects and knowledge that is related with this data, together. The structured data models include; classes with the subclasses and their properties which are the essential parts of the Semantic Web technology. The Semantic Web technology provides a more declarative information and knowledge mechanisms to make inferences.

2.1.1 Semantic Web Applications (SWW, 2017)

Biogen Idec: A Supply-Chain Management application

Biogen is a pharmaceutical maker especially concentrated on the multiple Sclerosis treatment and for that purpose, Biogen utilizes the Semantic Web technologies. The main support properties of the Semantic Web technology for the Biogen is the supply-chain management. The data currently used by the system, shipping time, rules, regulations and also the views for that data change constantly. The suppliers of the system can see the request on time and the collaboration takes place for the supply chain. Also some inferences can be taken from the data over a time period, and as a result, some improvements are made to the supply-chain management system. The idea behind the usage of the Semantic Web technology in supply-chain management is that the data model is transparent to all users (Grubic & Fan, 2010). This way, any employee of the company can understand and manage the system without having any expertise in IT or ontologies.

BBC: Media Management

The BBC utilizes the benefits of Semantic Web technology for declaration, arrangement and storage of media. Another usage of the Semantic Web technology is for the Web pages of BBC. Since the pattern of the Web sites change constantly, to make that manipulation in an easy and controlled way, the links for blogs, relationships between pages, and also the links between videos and the pages are stored and controlled with the Semantic Web technology.

Chevron: Oil and Gas Company

The oil companies have many of the oil rigs in offshore locations. Each of the oil rigs produces many heterogeneous data i.e. measurements, signals, graphics etc. each day with the size of terabytes of data. This data includes; production amount, drilling figures, machinery signals, seismic analysis combined with the marketing strategies, etc. Also each improvement to the oil rig in terms of a machine with new measurement information increases the amount of data. The manipulation of the

heterogeneous and collected data is time-consuming, ad-hoc and costly with the traditional industry approaches like relational database systems. The Semantic Web solution is integrated to enable easy collection and correlation of arbitrary data. The benefit of this approach makes it possible to predict mechanical anomalies, take precautionary measures and also plan the marketing strategy in a consistent way with the current situation.

Bio2RDF project

The purpose of the project is to provide a large-scale Semantic Web about interlinked biological data. The project began in 2005 and today has reached large scales of data that includes almost sixty headings about life sciences. Those main headings can be summarized as chemical data, gene interpretation, interaction of proteins and genome information.

DBPedia

DBPedia is an archive application on the Web that includes extraction of Semantic Web data from Wikipedia. DBPedia provides a considerable amount of structured data on the Web. DBPedia works as an advisor for the Semantic Web since it provides core structured data for any topic that has been already stored in Wikipedia.

2.1.2 Search Engines and e-commerce applications using the Semantic Web

The organized and related data with the meta data results in a more relevant search result than the search engines. To make this adoption, most of the e-commerce companies are using GoodRelations vocabulary for their definition of goods to generate Semantic data to be served with the current Web pages of their own. Those changes aim to achieve the essential intent of the Semantic Web technology which is the creation of more structured and readable Web content (Macris, 2011).

To make more structured web page content which is a Semantic Web version of the current web pages, the Resource Description Framework (RDF) or a structure which is similar to RDF is used for the web pages which are embedded to current web pages (W3C, 2017).

For example, Facebook developed its own Open Graph Protocol, and the Schema.org has been utilized by Microsoft, Yahoo and Google. In general, for the biggest players of the e-commerce industry, the GoodRelations framework is used which is also based on RDFa. Nowadays, with the improvements and the arising Semantic Web technologies, some popular Semantic Search Engines have been introduced such as; Kngine, Hakia, DuckDuckGo, Evri, etc. (Choudry, 2014).

Since the Semantic Web technology has been introduced, the linked data principle, i.e. structured data about data and metadata, has grown and used among many of the Web sites such as; Social data, Friend-of-a-friend, e-commerce, e-catalogs, Biological projects, Supply-chain managements, Government data, etc. Semantic Web technology utilizes ontologies as an essential component, since the ontology provides the interlinked data structure which has a more declarative format.

2.2 ONTOLOGY

Semantic Web makes the Web more readable and processable by machines and this technology enriches the Web content to reach its almost full potential (Berners-Lee et al., 2001). In the beginning of the Semantic Web, the provided languages required more effort to create the structured and declarative contents (Asuncion & Corcho, 2002). These languages which were thought as revolutionary languages to enrich the Web content are; Resource Description Framework (RDF), Simple HTML Ontology Extensions (SHOE), Ontobroker, and Defence Advanced Research Project Agency's Agent Markup Language (DAML).

The main idea behind the ontology approach is to define the object more declaratively in terms of their relations and connections (Lin et al., 1996; Fayez et

al., 2005). Also, a rule set can be defined for an ontology which permits making an inference from the information stored semantically. This property provides an identical object definition to distinguish the object from other objects which have similarities (Fadel et al., 1994). Ontologies provide a collection of information for identifying explicit links between objects (Gruber, 1995). This content-based data model identification of the ontologies, contributes to interoperability and also to communication among the different platforms on the web.

Ontology and database technologies have some similarities regarding data. The main difference between the ontology and the database is that; ontology does not store data but stores the interconnection of data as a map. On the other hand, a database stores the data in formal tables where the relations are exactly defined. Due to this main difference, when a query runs for an ontology, the inference rules are used to make reasoning due to the ontology representation, but in a database system, a query returns the exact values that are stored in an exact way of relations i.e. not a reasoning result (Elmasri & Navathe, 2003). On the other hand, the ontology technology utilizes the paradigm from object-orientation such as class and inheritance when declaring relations between data models. Different than the object-orientation, ontology does not include the multiple inheritance principle.

To create an ontology which agrees with the requirements of a domain, the ontology experts spend time to understand and extract the metadata for the domain-specific application retrieved from the domain experts (Silva & Rocha, 2003). The extraction of the metadata is not always a trivial process due to extraction from the textually-documented data or the conversation based on the natural language skills and abilities. Another important point for an ontology is the evaluation (Liu et al., 2011). Evaluation of the ontology should be parallel to the knowledge, since the knowledge is not a static construction and changes should be applied in some time periods (SWSF, 2005).

2.2.1 Ontology Essentials and Development Tools for the Semantic Web

An ontology definition has four parts; set of relations, instances, concepts and axioms. The Ontology creation has an Extensible Mark-up Language based structure. The tools and languages to create Ontologies are described in the following sections.

XML

XML is a short name for Extensible Mark-up Language (W3C, 2017). The main purpose of XML is to transport data to different platforms on the Web since XML utilizes plain text format to store data. XML provides a base for structured data but does not include any Semantic information. The basic structure of XML makes it readable for both humans and machines (Park et al., 2014). The structure of XML also provides independence of data among different software and platforms. Another property of XML is that a user can declare and use his/her own tags according to the design requirements of the platform on the Web. The language can also be used to define application-specific mark-up languages.

Another important component of XML is XMLSchema. With Schema, a user can define the required restrictions about structures of an XML document. Schema can also be applied for the syntax validation of XML documents according to declarations. Some of the languages defined with XML are; RSS language used for news feeds, WSDL used to declare available web services, RDF used to declare resources for an ontology, and OWL which refers to Ontology Web Language, etc.

XML documents are created with a labeled and ordered structure which can include all kinds of data structures according to its syntax of declaration with XML. XML text format is derived from Standard Generalized Markup Language (ISO 8879) and starting from Feb. 10, 1998 has become a W3C Recommendation.

Xpath

XPath is a W3C recommendation and a major element of XQuery and XSLT. XPath is an XML language to declare a path in an XML document which provides a syntax to find information in an XML document. Since there exists the declaration about paths, Xpath uses the expressions to navigate in XML documents.

XQuery

XQuery is built on Xpath expressions as a query language for XML. XQuery is a W3C Recommendation (W3C, 2017). Similar to SQL for Relational database systems, XQuery is used to query the XML documents and also is supported by the major database engines; Oracle, IBM, Microsoft, etc.

RDF and RDFSchema

RDF is a data model for objects and refers to Resource Description Framework which is written in the XML language. RDF utilizes the language RDF/XML. RDF provides a structure only for machine-readable format of information, so that the described information can be readable and understandable by applications on computers. RDF is a W3C Recommendation starting from Feb. 10, 2004 and is a part of W3C's Semantic Web Activity (W3C, 2017).

Some of the examples for the usage of RDF;

- Defining properties for e-catalogs that may include stock count etc.,
- Declaring information like author, content, last modified, etc., about web pages ,
- Declaring content for search engines,
- Declaring digital libraries, etc.

RDF includes three parts namely Resource, Property, and Property value:

- A Resource could be anything that can have a Uniform Resource Identifier (URI), for example "http://www.vacation.com/hotels"
- A Property is a Resource that has a name, such as "room" or "kitchen"
- A value is the value of a Property, such as "Suit Room" or can be another resource such as "http://www.hotelloccationa.com"

The following RDF document describes the resource "http://www.vacation.com/hotels":

```
<?xml version="1.0"?>
<RDF>
<Description about="http://www.vacation.com/hotels">
```

```
<room>Suit Room</room>
  <homepage>http://www.vacation.com</homepage>
</Description>
</RDF>
```

RDFSchema is a vocabulary of RDF documents to describe properties and classes of RDF resources. RDFSchema is a W3C Recommendation starting from Feb. 10, 2004 (W3C, 2017). To represent machine-interpretable knowledge, some Semantic Web applications use RDF.

2.2.2 RDF and the Semantic Web technology

RDF uses Uniform Resource Identifier (URI) when identifying resources and resources are described with properties and their property values. RDF has been accepted as a part of W3C's Semantic Web Activity to achieve the following;

- to represent exact meaning,
- to be understandable and processable by machines,
- to be incorporated by computers.

OWL

OWL refers to Web Ontology Language which is based on XML (OWL, 2004). OWL is designed for machine-readable information structure for applications that need to process information in a document. Ontology includes representation of terms and their interrelationships, and OWL is more expressive when compared with XML, RDF, and RDF-S in terms of representing the meaning of terms in vocabularies and the relationships between those terms (W3C, 2017).

OWL has three versions designed to be used by different communities of developers and users:

- OWL Lite
- OWL DL (includes OWL Lite)
- OWL Full (includes OWL DL)

“OWL is a component of the Semantic Web activity”, and OWL became a W3C (World Wide Web Consortium) Recommendation in February 2004 (W3C, 2017).

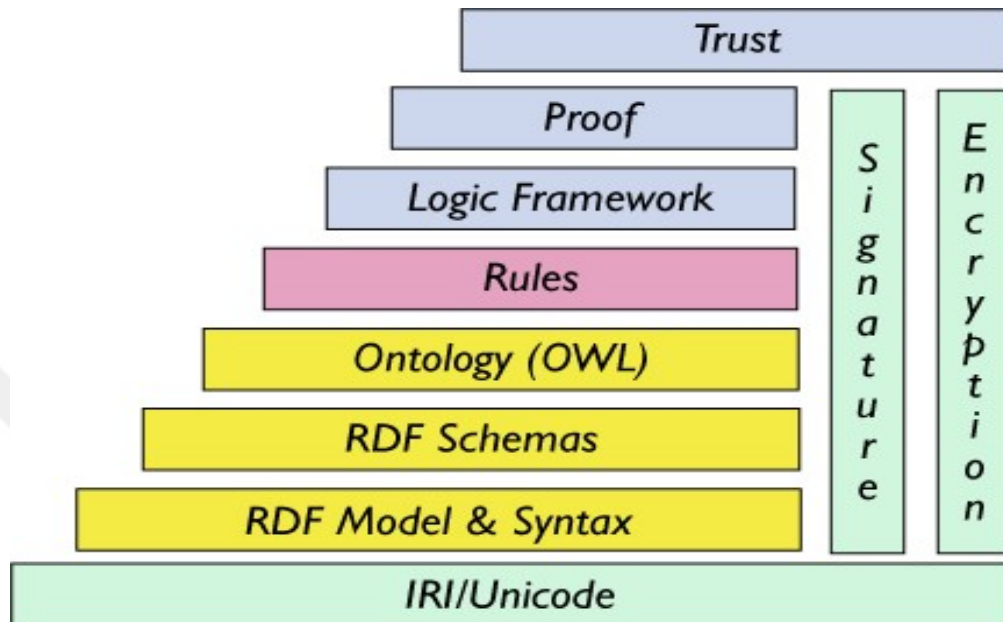


Figure 1. Semantic Web Framework (W3C, 2006)

2.2.3 Primary Ontology Creation Tools

Protégé: is an Ontology editor developed by Stanford University to build knowledge-based solutions in 2000 (Protégé, 2017). The knowledge-based solutions include e-commerce applications, biomedical knowledge bases and modelling of organizations. Protégé is an open-source and free framework and Ontology editor. The tool, supports different plug-ins to support varied scale of ontology-based applications. The tool supports Ontology Web Language and RDF specifications standards of World Wide Web Consortium (Protégé, 2017; W3C, 2017).

NeOn toolkit: is an open source ontology editor. NeOn toolkit provides support for the ontology engineering and extensive set of plug-ins to be used with Eclipse Java programming editor. NeOn toolkit was introduced in 2006 as a part of the NeOn project which aims to support large-scale Ontology applications (NeOn, 2017).

Altova SemanticWorks: is not an open-source software tool which is introduced in 2012 by Altova company (Altova, 2017). Altova SemanticWorks provides a

platform-independent, high-performance and standard-based software creation tool including visual RDF generation.

DOMÉ: is an XML editor, which is short for DERI Ontology Management Environment which is introduced by Ontology Management Working Group in 2006 (DOMÉ, 2017). The tool is developed to be used in knowledge extraction of Web pages to generate RDF versions and can be used as a plug-in to Eclipse Java editor.

Knoodl: is a platform that supports Java service-based and SPARQL-based interfaces for communities to develop OWL based ontologies and also RDF documents. Knoodl was introduced by Revelytix Incorporation in 2006 (Knoodl, 2017).

Apache Jena

Jena was developed in 2000 by the developers in Hewlett Packard laboratories as an open-source project (Jena, 2017). Jena is used for a wide range of Semantic Web projects. In 2010, Jena developer team applied for adoption of Jena to the Apache Software Foundation. Apache Jena is a Java framework which is used to create Linked Data and Semantic Web applications [jena.apache.org]. Apache Jena utilizes different Application Programming Interfaces (API) on the Java platform to process RDF data.

In Jena, Resource Description Framework is used to describe resources. A resource is a Uniform Resource Identifier (URI). Resources have properties and each property has a value called a literal in Apache Jena. Graph representation of Jena for the RDF includes;

- a resource shown as an ellipse,
- a property shown as a labeled arc between the resource and the literal, label identifies the name of a property. In Jena, there exists predefined namespace i.e. vcard,
- a literal is shown as a rectangle.

To give an example for the RDF graph; Let

- resource (URI): <http://.../EkremYilmaz>,

- property : vcard:FN (stands for First Name), vcard is a namespace prefix, FN is called a local name,
- literal: Ekrem Caglar Yilmaz (as a value of property).

The given information about the RDF graph can be shown as;

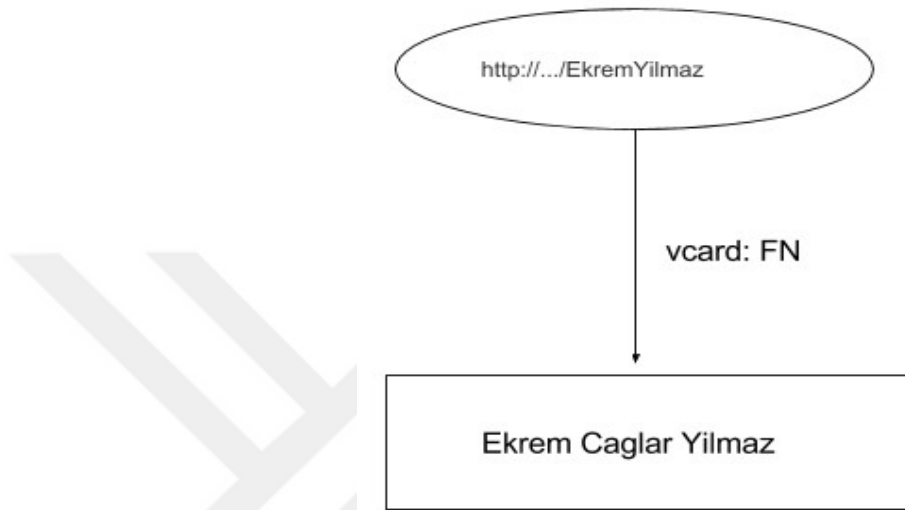


Figure 2. RDF Graph Example

While declaring the interfaces in Jena, resources are used as Resource, properties are used as Property and literals are used as Literal. Graphs in Jena are called Models which have Resource, Property and Literal. The required Jena code to create our example is shown in Figure 3.

```
// pre-definitions
static String person_URI = "http://somewhere/EkremYilmaz";
static String full_Name = "Ekrem Caglar Yilmaz";

// create an empty Model to represent RDF Graph
Model model = ModelFactory.createDefaultModel();

// The Resource creation
Resource ekremYilmaz = model.createResource(person_URI);

// adding the property to resource
johnSmith.addProperty(VCARD.FN, full_Name);
```

Figure 3. RDF Graph creation example in Jena

In our example, the property has a literal which holds a value. In many cases, the property can point to other resources instead of literals. In the following example, a more complex RDF Graph representation is given in Figure 4.

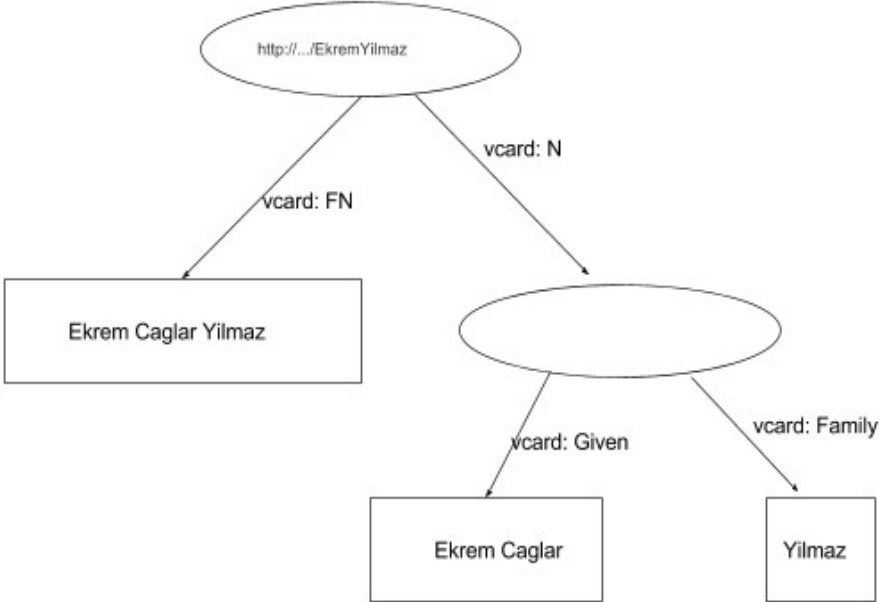


Figure 4. RDF Graph creation example 2 in Jena

In this example, a new property is added as vcard:N, to represent name in the structure of Ekrem Yilmaz's name. As shown in the Model, the vcard:N property points to another resource as a value, and new resource has no URI. This time, the resource is called the blankNode.

In Figure 5, the required segment code is given to create RDF graph of the second example in Jena.

```
// pre-definitions
static String person_URI = "http://somewhere/EkremYilmaz";
static String full_Name = "Ekrem Caglar Yilmaz";
String givenName = "Ekrem Caglar";
String familyName = "Yilmaz";
String fullName = givenName + " " + familyName;
// create an empty Model to represent RDF Graph
Model model = ModelFactory.createDefaultModel();
// The Resource creation
Resource ekremYilmaz = model.createResource(person_URI);
// adding the property to resource
ekremYilmaz.addProperty(VCARD.FN, full_Name);
// creating another resource
// and adding other properties in a cascading style
Resource ekremYilmaz
= model.createResource(personURI)
.addProperty(VCARD.FN, fullName)
.addProperty(VCARD.N,
model.createResource()
.addProperty(VCARD.Given, givenName)
.addProperty(VCARD.Family, familyName));
```

Figure 5. RDF Graph creation example 2 in Jena

Another component in the RDF model is called statement. Each statement declares a relation of each property. A statement is defined by three parts;

- subject part is the resource from which the arc originated,
- predicate part is the property which labels the arc,
- object part is the destination of an arc, i.e either the resource or literal.

The set of statements are used to represent an RDF Model in terms of relations.

2.2.4 The Semantic Web technology and Ontology usage in ERP

ERP is a business management system that includes marketing, finance, ordering of material, control of inventory, manufacturing, planning of production, testing, delivery, sales, and after sales, customer support (Bingi et al., 1999). The companies which use the ERP solution invest a lot in software customization, personnel training, integration of modules, test of modules, conversion and analysis of data. Other problems for an ERP are the information heterogeneity for contents and the relations between various sources such as customer requirements, suppliers, etc.(Bingi et al., 1999). The process of complex business processes which include many functionalities may involve human interaction to assemble pieces for functionality. The human participation in such activities may cause a gap in time and knowledge. The Semantic Web can be used to encode the information in a machine readable format as a knowledge base which provides automation . The Semantic Web may overcome the problems that are originated from human participation in functions of ERP systems (Anjomshoaa et al., 2006).

The goal of the Semantic Web is mainly concentrated on formalizing information into knowledge resources (Berners-Lee et al., 2001). In the Semantic Web, the information is declared with the meaning, therefore, this approach provides a machine-readable format of information for the computers to process the information (Cabral et al., 2006). This property of the Semantic Web technology can also be used in organizations that have the context of ERP, and this property of Semantic Web can also provide readiness to integrate and share business for particular web services of the world of business (Fayez et al., 2005).

The Semantic Web technology can be also used to overcome the problems due to the non-standard schemas of information contents available from various sources such as organization, cost, resource etc. According to the nature of business, the ERP software performs the task of business in a modular way (Fensel & Bussler, 2002). The main purpose of an ERP is to work for the achievement of common goals for a

company. Therefore, different goals of ERP require specific vocabulary and concept hierarchy for a particular achievement. The Semantic Web technology is suitable for ERP and can be utilized in the context of an ERP to provide efficiency for the automation of processes (Fayez et al., 2005). The timing activities and information processing in ERP systems are very important for efficiency, and the integration of Semantic Web into ERP systems provides opportunities for solution to timing and orchestration of business processes.

The present Semantic Web applications that have proven to assist in ERP mainly concentrate on human resource, manufacturing, planning and organization.

2.2.5 Semantic Web Applications in ERP

There are several Semantic Web applications in business and most of the applications are located in Business to Business environments that integrate Semantic Web services as a solution to overcome complex business structures (Tamma et al., 2005).

Digital Enterprise Research Institute

DERI implements intelligent Web Services with Semantic Web technologies. DERI mainly concentrates on comparison of services and negotiation of automated services. DERI provides Web services when an enterprise service requires a business interaction with another enterprise's services in an automated process according to the relied selection policies. Required services can be invoked automatically and in the following steps, a payment process can be initiated (DERI, 2006).

Resource-Event-Agent

REA is a business model that utilizes ontologies for accounting. In later applications, REA model is used in supply chain management. The key change for REA model is the application to Manufacturing Resource Planning (MRP) systems on Dependent Demand relationships (Haugen & McCarthy, 2000). The REA model utilizes the Semantic Web to make it possible to negotiate with different companies and industries. These links are generated semantically between person-to-person, agent-to-agent and activity-to-activity, therefore in RAE model; each individual can directly be linked to every other individual.

Supply Chain Operations Reference Model

The SCOR model is developed as a process reference model by Supply-Chain council to work as a diagnostic tool for supply-chain management in cross-industry. SCOR model enables users to communicate and interact with supply-chain management practices among all interrelated parties. The other beneficial properties for supply-chains in the SCORE model are: measurement of performance and benchmarking among industry standards (SCOR, 2006).

SCOR+ utilizes more specific ontologies for the process of business description that has been developed by SCOR. The main concentration of SCOR+ is to overcome the limitations of the basic SCOR model via using ontology description power in order to declare any level of detail for a Supply Chain. These limitations can be grouped as the supply-chain level, the enterprise level, the element level and the interaction level (SCOR, 2006).

Enterprise ontology

The Enterprise Ontology (EO) has been developed at the University of Edinburg as a part of the Enterprise Project . The main purposes for EO are: enhancing

communication between humans, supporting interoperability and providing a basis for end-user applications. Enterprise Ontology has some sections such as Activity, Plan, Organization, Strategy, Resource, Marketing, etc. . Some of the concepts in EO are generic for many organizations like Person, Activity but the remaining concepts like Ownership are at an abstract level. The EO provides a basis for the area of enterprise ontology and supply-chain ontology models .

TOVE ontologies

The Toronto Virtual Enterprise Ontology Project has been developed to answer the queries about the tasks in industrial environments . The TOVE utilizes the Enterprise Model. The tasks that are specified in TOVE include a great detail of supply-chain management which implements mainly Material Resource Planning (MRP) that includes logistics and distribution for the concurrent engineering with coordination of engineering design . For the generated ontologies, a single company perspective has been taken. The generated ontologies are: cost, resource, organization, product, quality management .

Model

The ontology model introduced by Soares et al. (2000) utilizes EO as a base model. The main concentration about this model is to develop human communication and specify system requirements for production planning in semiconductor sector . The proposed ontology is used as a part of the trans-European project which includes many of the industrial companies and with multiple academic intuitions' for that project. The concepts are gathered in the form of natural language and then transferred into object models under the main sections: networked organizations, orders management and plans.

IDEON ontology

The IDEON ontology has been proposed by A.M. Madni et al . The purpose of the ontology is to capture concepts and relationships which describe an enterprise. For that purpose, four views were introduced and each of these views are represented as an object-oriented model using UML notation . The views include different levels of information for the interaction of an enterprise in a particular environment, goal, strategy, process, person and objective. According to inferences from the views, replanning for the design of a foundation, invention, management and collaboration of an enterprise may be advised .

Manufacturing system engineering ontology

The manufacturing system engineering (MSE) ontology has been proposed by H.K. Lin et al. The purpose of the ontology provides an intelligent support for an MSE Moderator for the environment of prolonged or virtual enterprises in terms of enhancing the degree of awareness, cooperation and coordination of concurrent engineering team members . Ontology of MSE utilizes classes in a hierarchy with super class and subclass approach. These classes are Enterprise, Process, Flow, Project and Strategy. Each class has own role such as managing processes and aggregation of Enterprises .

Supply-Chain Model

The ontology has been developed for supply-chain ontology and proposed by Y.Ye et al. The main purpose for this ontology is to integrate heterogeneous information among the systems in a semantic way. For that purpose, the participating enterprises agreed on a vocabulary and this approach overcomes the ambiguities for the ontologies . The partnerships of enterprises are created dynamically and this participation may be short or long for the supply-chain ontology. The ontology

utilizes different level of classes like Resource, Performance, Metric of Performance, Supply Chain Structure, etc . The Supply chain ontology has been created using the Protege tool and stored in the format of Ontology Web Language (OWL). For further use, this ontology model was used to create virtual supply chain management to overcome possible problems in a scenario to integrate information to supply chain .

As seen in the Semantic Web examples given above, usage of Semantic Web is an important tool to assist in the human processing of ERP. The general approach to create a Semantic Web for a company includes the following steps: allocation of an expert to create the required ontology, explanation of business and requirements of business, and creation of the ontology by the expert. These steps may include some limitations or problems such as, structuring of the ontology may be limited to ontology expert's level of clearance for the business, the orchestration of processes may require to involve some confidential information or clues that should stay as a secret for the business, etc. .

CHAPTER 3

DESIGN AND IMPLEMENTATION

3.1 Design of System

The tool is designed for users who have no expertise in information technologies, especially about ontology creation tools. For this purpose, a stepwise and menu-driven user interface has been created. In each step of the ontology creation, user is asked to enter the components of the ontology in the following order;

1. Resource,
2. Property and Value (if exists),
3. Relation.

In Figure 6, the components of design and their orchestration are illustrated.

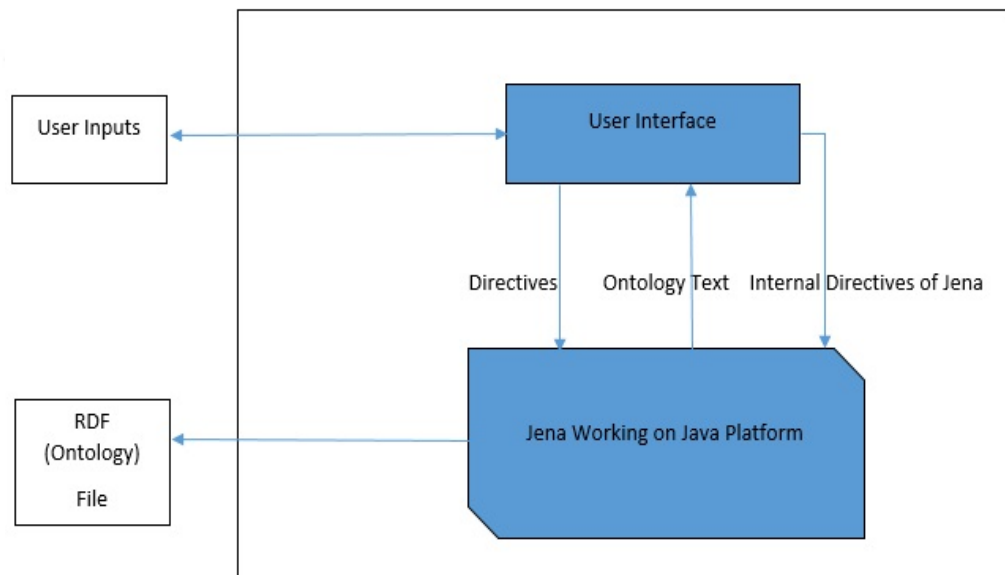


Figure 6. The Components of the System

As shown in Figure 6, interaction between the system and user can be summarized as; the user inputs the main Resource which is located as URI of the ontology. Then the system asks the user to input resources and/or properties. After each resource or

property is created by Jena, a proper message is shown on the user interface. Then the system asks the user to set the relations between resources as a next step. The last step includes the creation of the ontology to display on the screen or output to an RDF file.

3.2 Technical specifications

The developed system is a generic tool to create ontologies by novice users who have no information technology expertise. For this purpose, the system is developed using the Java programming language as a desktop application instead of creating a plug-in for the currently being used ontology tools such as Protege. Java is an object-oriented programming language and almost independent from platforms, i.e device to be used. There exist Application Programming Interfaces (API) for domain specific applications to create through the Java language such as Apache Jena (Jena, 2017) which is a free and open source API of Java. Jena is used to create Semantic Web applications and ontologies as a programming tool. The system utilizes Jena API to create ontologies, and Java Swing classes to create the user interfaces of the system.

3.3 System interfaces and capabilities

On the starting window, the user of the system is asked to enter the main resource into the textbox labeled as “Create Ontology About”. After requested information is entered into the box, user need to be click to “Next>>” button.

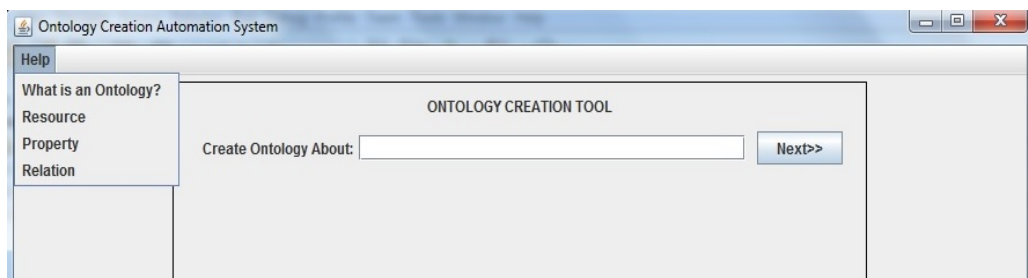


Figure 7. Starting Window

A help menu is located at the top-left location of all windows through the system.

“Help” menu includes the items;

- “What is an Ontology”,
- “Resource”,
- “Property”,
- “Relation”.

When a user tries to get information about “Resource” item under “Help” menu, another window opens to give the requested information as shown in Figure 8.

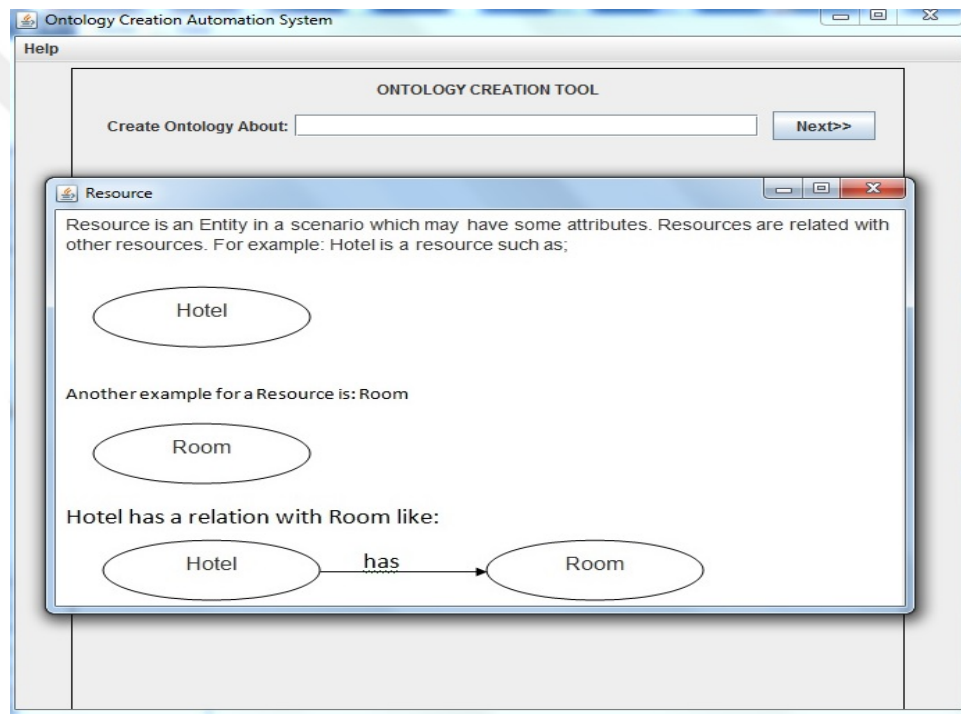


Figure 8. Help Menu Content

After “Create Ontology About” page’s textbox is filled, the next page requests to enter the “Resource Name” to add Resource into the Ontology. After “Resource Name” is entered into the textbox, user can add by entering Properties with corresponding values if exists, by clicking “Add Property” button. Then, the user needs to click “Add Resource to Ontology” button to locate the Resource with its own properties as shown in Figure 9.

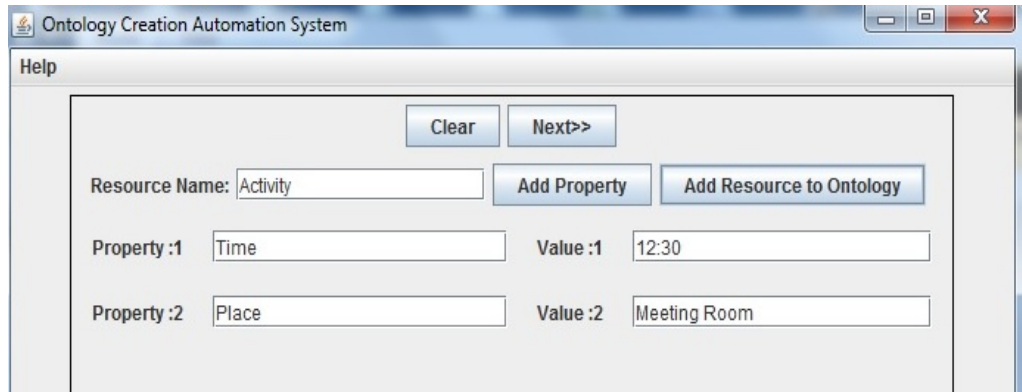


Figure 9. Add Resource window

After “Add Resource to Ontology” button is clicked, a message “Resource 1 added” appears since only one Resource exists. User can continue to add new resources with new properties pertaining to that resource using the same procedure. Figure 10 shows the resource added message.

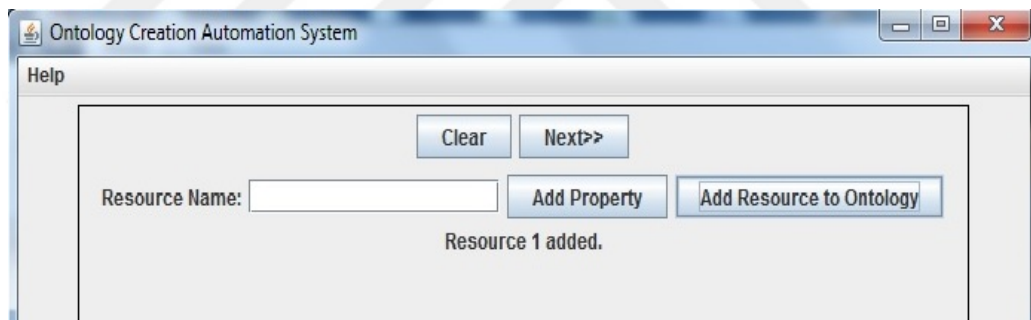


Figure 10. Resource Addition Feedback

As the next window of the system, relation declaration window is shown to the user. On that window, user-defined resources are listed in drop-down menus on both sides to be selected, and in the centered drop-down menu, the predefined relation type items are listed. Using these menus and predefined relation types, user can set a relation between properties such as: isA, hasA, actsAs, etc. as shown in Figure 11.

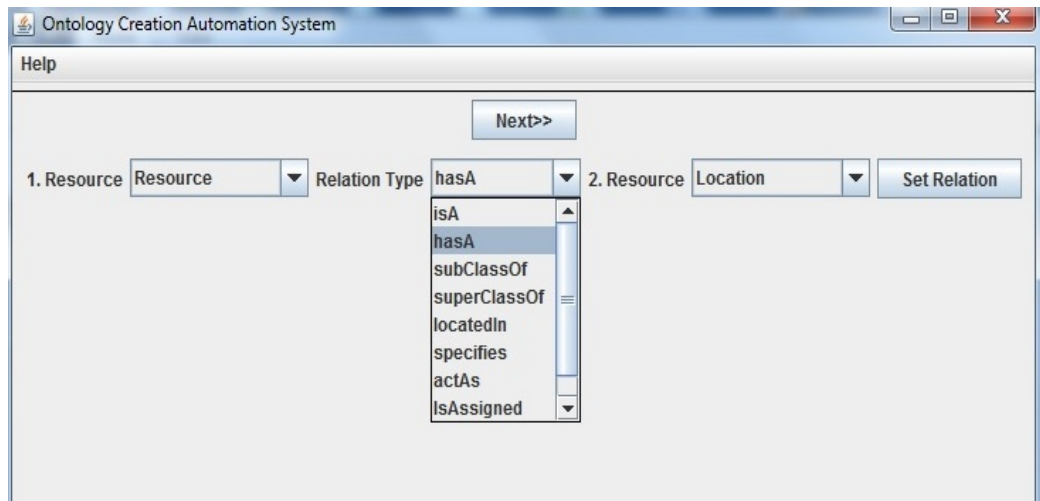


Figure 11. Setting a relation between resources

On the same window, user can delete the relations which are declared by clicking the erase button located in each relation previously as shown in Figure 12.

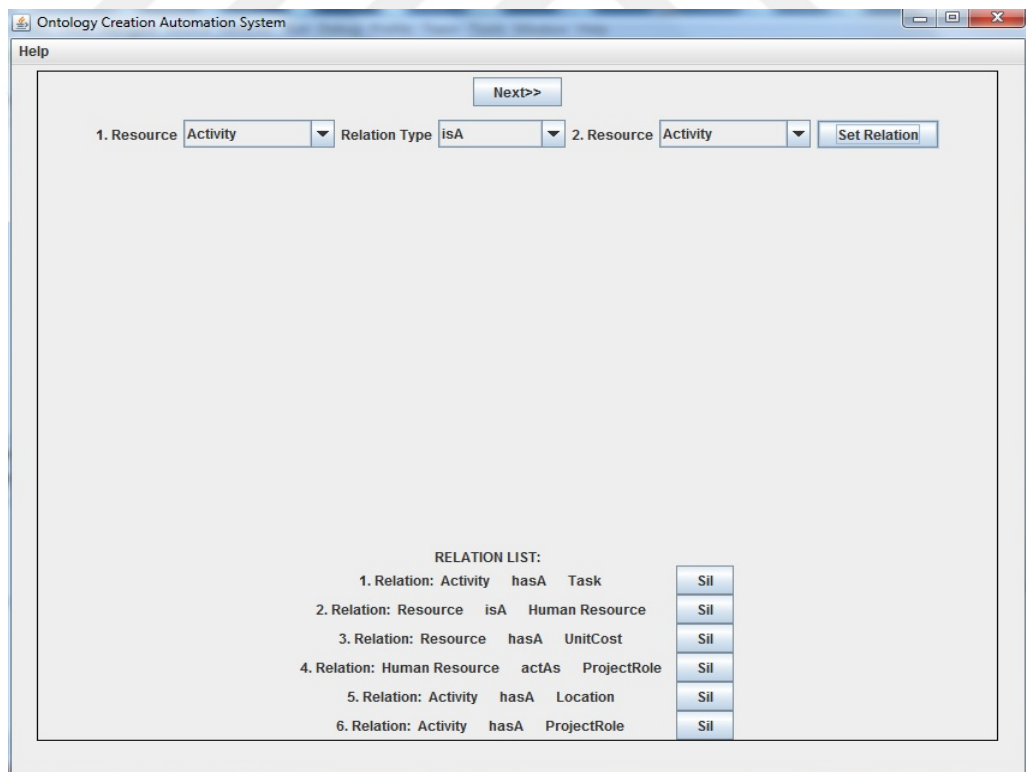


Figure12. Relation Control Property

On the last page of the system, there are three buttons located before the empty text page.

The buttons are;

- “Generate RDF” button: shows the generated RDF of the newly created ontology into the textpage as shown in Figure#,
- “New RDF” button: erases the generated RDF and starts the whole process,
- “Write RDF to a File”: creates a file which includes the user defined RDF which is the generated ontology.

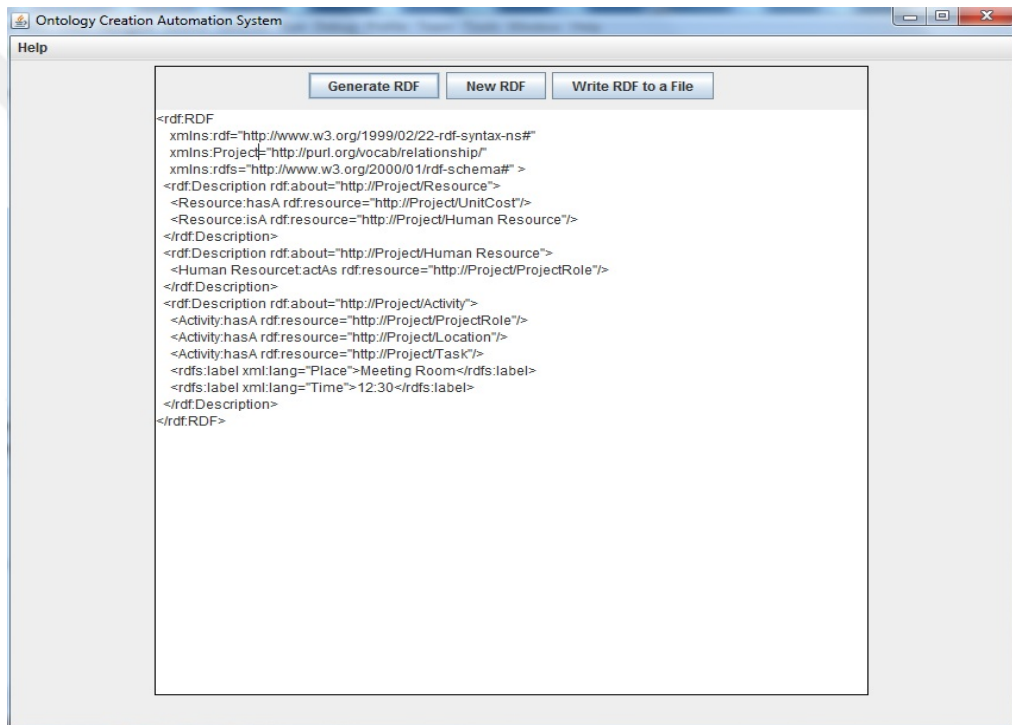


Figure 13. Generated RDF output on Textbox

3.4 Case Study: Project Ontology Creation

In our case study, an RDF graph of a Project is selected to create an ontology. In the selected RDF graph, all items are Resources with no properties. The “Resource” information of the ontology is declared as “Project”. Project ontology has 14 “Resources” as;

- Activity, Task,
- TimeFrame,
- Production Plan,
- Location,
- Task Type,
- Resource,
- Human Resource,
- Non-Human Resource,
- Unit Cost,
- Bill of Material,
- Project Type,
- Project Role.

There are relation identifiers between resources such as;

- hasA,
- specifies,
- performedAt,
- isA,
- actsAs,
- locatedIn as shown in Figure 14.

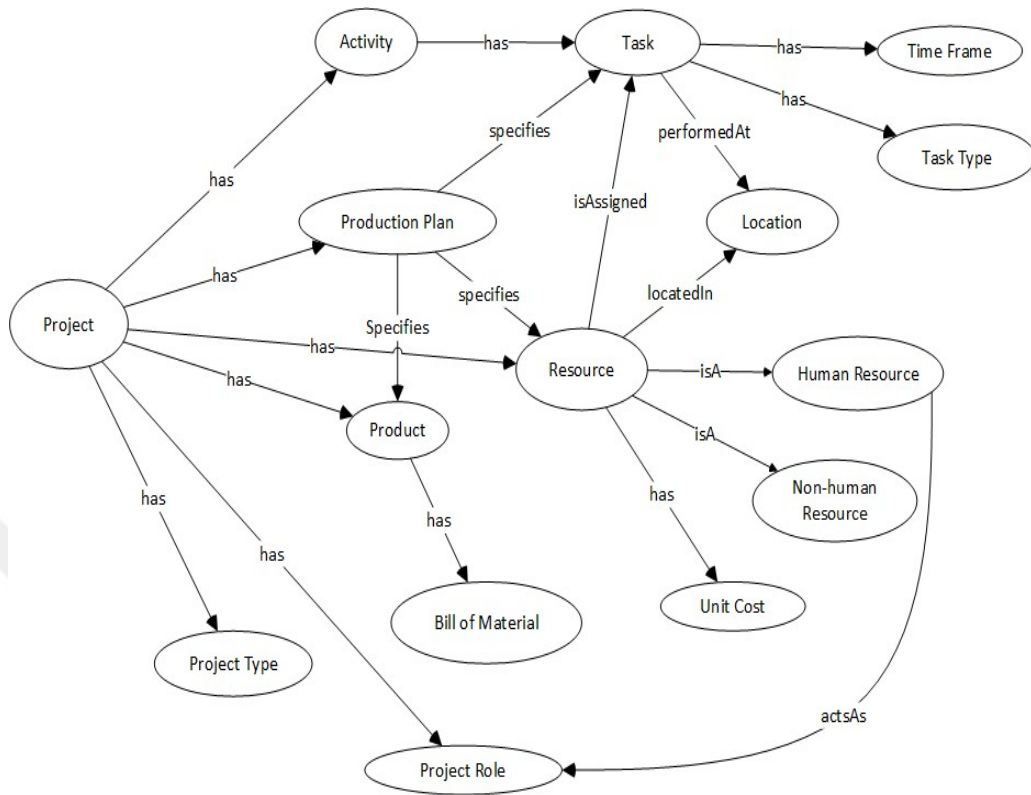


Figure 14. RDF graph representations of a Project

First step of creation of ontology is entering “Resource” part of information which is “Project” as shown in Figure15.

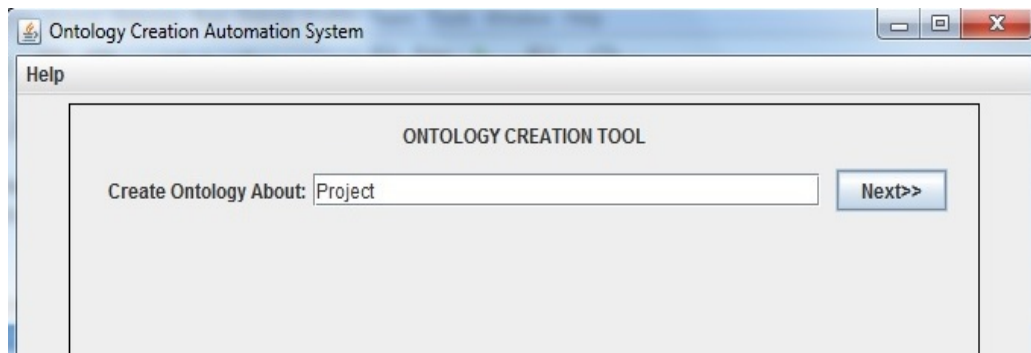


Figure 15. Ontology Resource

The second step is entering subresources named Activity, Task, TimeFrame, Production Plan, Location, Task Type which are declared under Project resource as shown in Figure 16. After each addition of a Resource, user is informed with “Resource number added”.

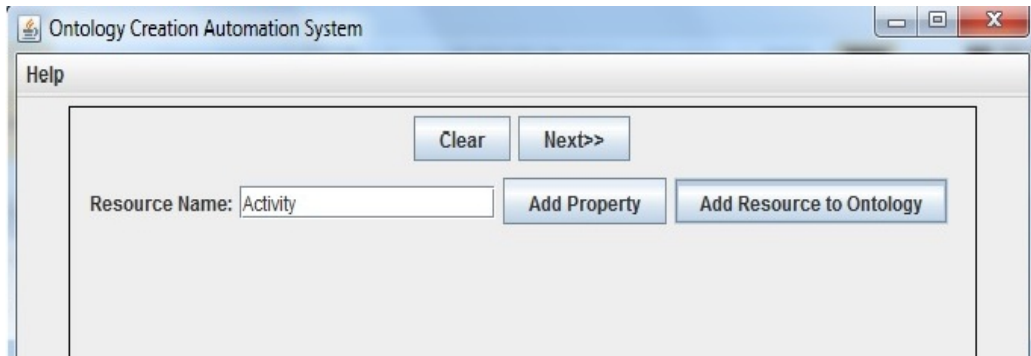


Figure 16. Resource Addition Window

Third step is one of the most important components of an ontology which is declaring relations between Resources. As shown in Figure 17, user-defined resources are listed in a dropdown menu on both sides, and in the middle of the window, predefined relation types are listed again. Each of added relation are shown in the list located in the bottom part of the window with the “Erase” button to control the relation list by the user as shown in Figure 17.

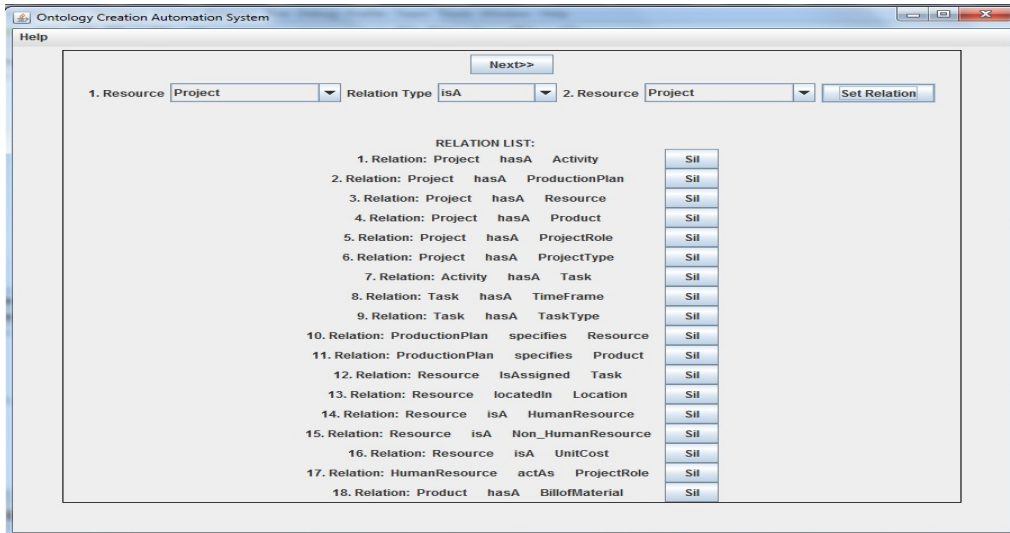


Figure 17. Relation setting between Resources and Control

The last step is the generation of the RDF output of the created ontology. On this window, to generate the RDF output, the user clicks on the “Generate RDF” button to get the resulting ontology as shown in Figure#. User can also choose “Write RDF to a File” to get the generated ontology output in an RDF file.

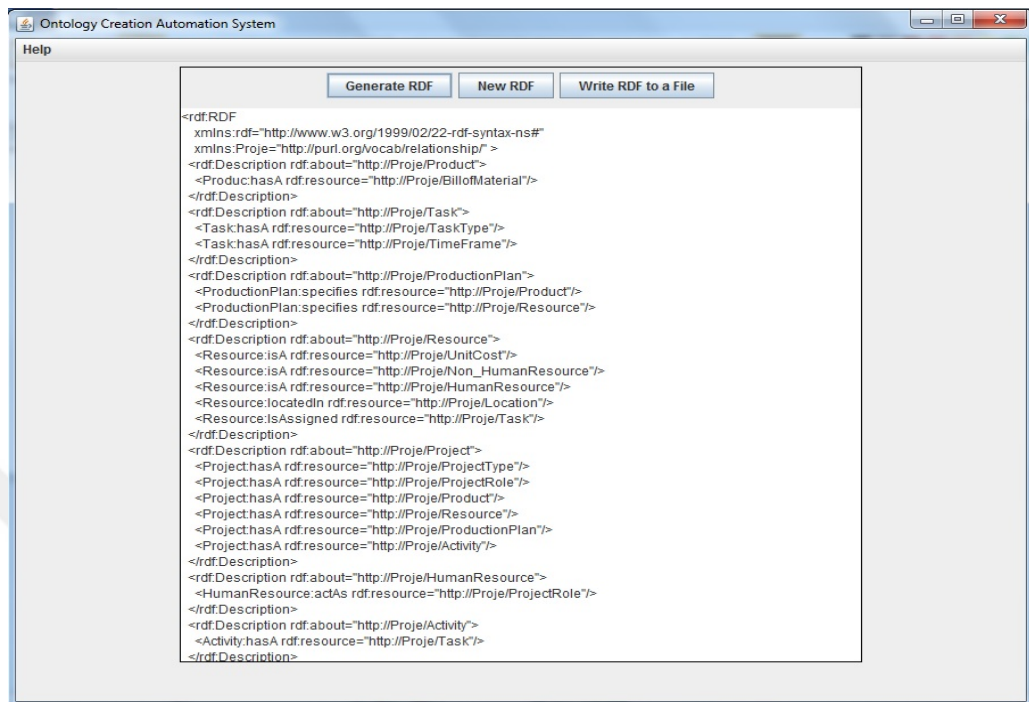


Figure 18. RDF output of the created ontology

CHAPTER 4

EVALUATION

For evaluation purposes, 25 participants from different divisions of 8 different companies were asked to use the ontology creation tool for different purposes. All the users were observed in each step of tool usage and ontology creation, and were asked to fill a questionnaire. The ontologies created by the participants were then evaluated and in addition, notes were taken from observations to make refinements for the future works.

4.1 Purpose of Questionnaire

A generic ontology creation tool has been developed for the users located in different sectors of business utilizing the Semantic Web technology. Users from different sectors such as supply management, purchasing, human resources, etc. were asked to use this tool.

The evaluation process is done in three steps by the users:

1. Develop the ontology given as an RDF graph in Figure 19 using the tool,
2. Create your own domain-specific ontology that is related with your sector.
3. Fill a questionnaire to evaluate the tool.

Finally, Statistical evaluation techniques are applied to show the effectiveness of the tool.

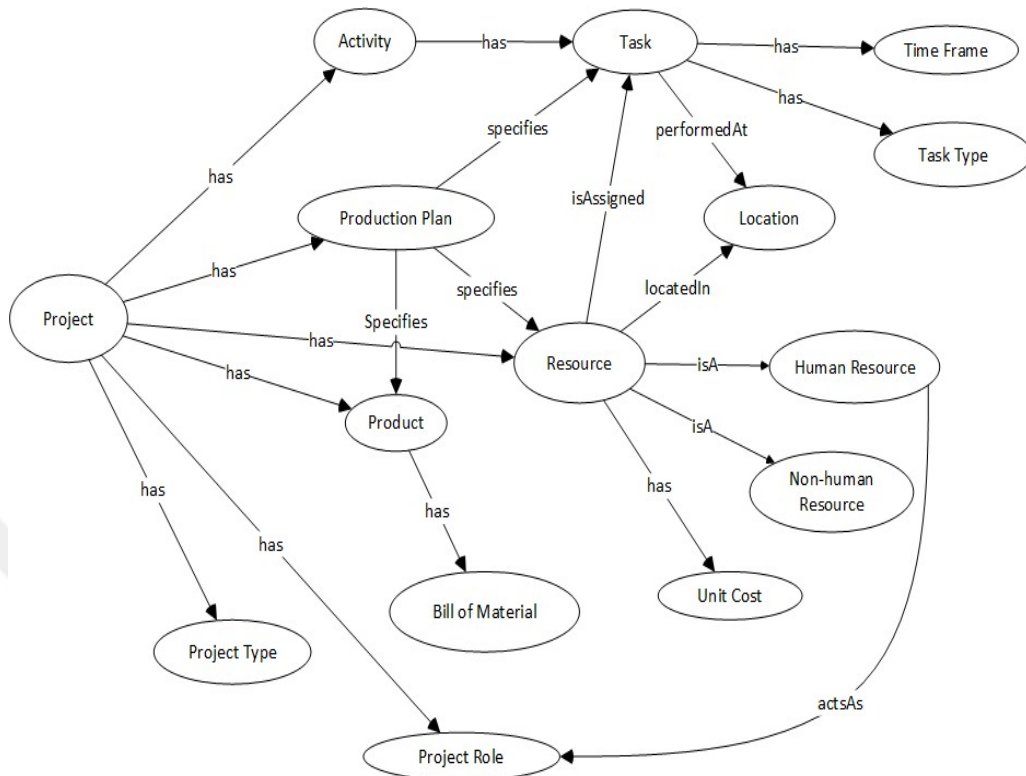


Figure 19. RDF of Project Ontology

In the following section, user profiles in terms of positions, working experience and age information are presented.

Profiles of Participants

Participants are selected among different sectors and categorized as; age, working experience, and working positions shown in Table1: Participant Profiles.

Table1. Participant Profiles

Position in the Company	Age of Participant	Working Experience		
		1-5 years	6-10 years	10 years and above
Purchasing	27	1		
Simulation Developer	32		1	
Project Manager	35		1	
Purchasing(Medical)	47			1
Project Manager	52			1
Production Material Manager	36	1		
Consumables(Medical)	41	1		
Pricing and Analysis	39		1	
Internal Production	35			1
External Production			1	
Purchasing Expert(Medical)	22	1		
Logistic	33	1		
Purchasing(Logistic)	25	1		
Finance and Planning(Construction)	28	1		
Business Development(Construction)	34			1
Human Resoruces(Construction)	39		1	
Supply Management(Stock)	41			1
Pricing and Analysis(Construction)	43		1	
Human Resoruces	29	1		
Production Material Manager			1	
Domestic Area Seller	38		1	
Foreign Area Seller	53			1
Importing Transactions(Medical)	48		1	
Hotel Management	42		1	
Human Resoruces Management(Hotel)	28	1		

As represented in Table 1, 25 people participated in the evaluation. Distribution of working experience is as follows; 9 people with 1-5 years working experience, 10

people working for 6-10 years and 6 people working for 10 years and above. Participants are selected who have different working positions such as; Importing and Transactions, Supply Management, Purchasing, etc.

Participants were feeling comfortable since user requirements do not need any expertise in terms of interface and usability steps of the tool. Each participant was asked to write any observations and feedbacks regarding to the tool. The main observations and feedbacks received from participants can be summarized as;

- more contrast colors can be applied,
- more user-friendly interface can be created in terms of button positions,
- process transaction is one way from beginning to end, there may be “return back” button,
- “Write RDF to a File” button may ask users to select a storing location,
- There should be more relation types between resources and properties.

Feedbacks and observations were recorded to make refinement as a futurework steps for the tool.

4.2. Questionnaire and Participant’s Results

In the questionnaire, the following questions were asked to evaluate the developed system. Since the developed system is a generic tool to create domain-specific ontologies, there is no comparison perspective with other ontology creation tools. 1-5 were given as grade where; 1 stands for “Very dissatisfied”, 2 stands for “Dissatisfied”, 3 stands for “Ok”, 4 stands for “Satisfied” and 5 stands for “Very Satisfied”. In Table 2, the questionnaire is given.

Table 2. Questionnaire Form.

ONTOLOGY CREATION TOOL EVALUATION QUESTIONNAIRE

Age of participant:

Working position of participant:

Working Experience a) 1-5 years b) 6 – 10 years c) 10 years and above

Evaluation grading: 1-Very dissatisfied, 2-Dissatisfied, 3- Ok, 4-Satisfied, 5-Very Satisfied	1	2	3	4	5
1.Instant System State					
1.a) Giving feedback for each step?					
2.System expression’s understandability					
2.a) Is the Help Menu visible?					
2.b) Is the Help Menu descriptive?					
2.c) Does the Help Menu assist in your problem solving?					
3.Usability of Sytem Functions					
3.a) Can you cancel of the processed state easily?					
3.b) Can you pass to the next step and its easiness					
4.Understandability of Sytem Functions					
4.a) Is the current step descriptive?					
5.System Consistency					
5.a) Are your steps consistent with each other in the system?					
5.b) Are the covered steps satisfying for you?					
6.Relation between the system application and reality					
6.a) Does the system guide you in a proper way in the work flow?					
6.b) Are the choices enough to achieve product outputs?					
7.Relations between Sytem and Standards					
7.a) Is the system interface clear to you?					
7.b) Evaluate the system for ease-of-use					
8.Flexibility and productivity state					
8.a) Did you start the System easily?					
8.b) Is the Systemeasy to use?					
8.c) Did you create your “Domain-specific” Ontology easily?					
8.d) Does the System complete your task in a sufficient amount of time?					

The main headings of questionnaire include the followings:

- Instant System State,
- System Expression’s Understandability,
- Usability of System Functions,
- Understandability of Sytsem Functions,
- System Consistency,
- Relation between the system application and reality,
- Relations between Sytem and Standards,
- Flexibility and productivity.

4.3 Statistical Analysis Steps and Inferences of Questionnaire Results

The hypothesis tested in the evaluation phase are as follows;

H1: The experience level of the user has significant influence on the perceived understandability of the ontology creation tool.

H2: The experience level of the user has significant influence on the perceived usability of the ontology creation tool.

H3: The experience level of the user has significant influence on the perceived consistency of the ontology creation tool.

H4: The experience level of the user has significant influence on the perceived ease-of-use of the ontology creation tool.

H5: The experience level of the user has significant influence on the perceived productivity of the ontology creation tool.

The statistical analysis is applied for the headings and related questions from questionnaire as follows;

- Understandability: 2b, 2c, 4a, 7a,
- Usability: 2a, 3a, 5b,
- Consistency: 1a, 5a, 6a,
- Ease-of-use: 3b, 7b, 8a, 8b,
- Productivity: 6a, 8c, 8d.

As we have an ordinal type of data (5-point likert scale), and the number of data is considerably small; the data group cannot be considered or even approximated as normal. Therefore, non-parametric methods which do not have a 'normality assumption' should be used to test the effect of the 'domain expertise' on the utilization of the tool for the creation of a domain ontology. In order to test this

effect, it is required to compare whether the data distributions of the results of the experienced groups and the unexperienced groups acquired from the results of survey are significantly different from each other or not. Mann-Whitney U test seems to be a fine test to examine this situation. The Mann-Whitney U test determines whether there is a statistically significant difference between two unrelated, independent groups on a dependent variable (Minitab tutorial, 2017).

For example, you could use a Mann-Whitney U test to determine whether there is a difference in test anxiety between undergraduate and postgraduate students (i.e., the dependent variable would be "test anxiety", and the independent variable would be "educational level", which has two groups: "undergraduate students" and "postgraduate students"). Alternately, you could use a Mann-Whitney U test to determine whether there is a difference in typing speed based on room lighting (i.e., the dependent variable would be "typing speed" and the independent variable would be "room lighting", which has two groups: "red lighting" and "blue lighting").

The Mann-Whitney U test has four "assumptions":

- **Assumption #1:** Your **dependent variable** should be measured at a **continuous** (i.e., they are **interval** or **ratio** variables) or **ordinal** level. Examples of **continuous variables** include height (measured in feet and inches), temperature (measured in °C), salary (measured in US dollars), revision time (measured in hours), intelligence (measured using IQ score), firm size (measured in terms of the number of employees), age (measured in years), reaction time (measured in milliseconds), grip strength (measured in kg), power output (measured in watts), test performance (measured from 0 to 100), sales (measured in number of transactions per month), academic achievement (measured in terms of GMAT score), and so forth. Examples of **ordinal variables** include Likert items (e.g., a 7-point scale from "strongly agree" through to "strongly disagree"), amongst other ways of ranking categories (e.g., a 5-point scale explaining how much a customer liked a product, ranging from "Not very much" to "Yes, a lot").

- **Assumption #2:** Your **independent variable** should consist of **two categorical, independent (unrelated) groups**. Examples of such independent variables include gender (two groups: male or female), treatment type (two groups: medication or no medication), experience group (two groups: experienced or unexperienced), educational level (two groups: undergraduate or postgraduate), health insurance (two groups: yes or no), intensity of religious practice (two groups: practicing or non-practicing), personality type (two groups: introversion or extroversion), and so forth.
- **Assumption #3:** You should have **independence of observations**, which means that there is no relationship between the observations in each group or between the groups themselves. Therefore, there must be different participants in each group with no participant being in more than one group. For example, if you wanted to determine whether there was a statistically significant difference in anxiety level between undergraduates and postgraduates (i.e., your two independent groups), a participant cannot be both an undergraduate and a postgraduate. The participant can only be an undergraduate or a postgraduate. The participant cannot be in both groups.
- **Assumption #4:** A Mann-Whitney U test can be used when your two variables are **not normally distributed**. However, for a Mann-Whitney U test to be able to provide a valid test of the difference between two medians, **both distributions must be the same shape** (e.g., the distribution of the scores for "males" and the distributions of the scores for "females" must have the same shape). (Minitab tutorial, 2017)

As can be observed from the above knowledge Mann-Whitney U test can be used to test the effect of 'working expertise' on the ontology creation process. In order to test this issue, random pairs of participant data are chosen from groups of different expertise levels and Mann Whitney U test is applied for the comparison of these groups.

Table 3. Individual Comparisons

Comparison pair	Significance	M-W U	Z Value	Median	Result of the TEST
A-B	0,000	44	-3,56	Median A=3 Median B=1,5	A and B are significantly different, the satisfaction level of A is greater then the satisfaction level of B.
A-C	0,05	90	-1,961	Median A=4 Median C=2	A and C are almost significantly different, the satisfaction level of A is greater then the satisfaction level of C.
B-C	0,017	77,50	-2,391	Median B=3 Median C=2,5	B and C are significantly different, the satisfaction level of B is greater then the satisfaction level of C.

A: 1-5 years of working experience, B: 6-10 years of working experience, C: 10 years and above working experience.

As the level of significance after the test is below 0,05 for these trials, it can be certified that the experience level has an effect on the ontology creation process; in other words the workers with less experience are more satisfied than the more experienced ones in the process. Assuming that the ontology model works successfully-which is the case, the result is an indication of the assisting capability of the model for the inexperienced workers.

Afterwards, the same test is carried out for testing the success of the application of the model in different ‘applicational categories’ such as ease of use, usability, productivity, etc. For this case, the mean point assessed by the users of different expertise levels are utilised and compared among groups. To perform the test, 6 people are selected randomly from each group, as ‘6’ was the number of users in the smallest group.

Table 4. Working Experience Groups’ Comparisons for “Understandability”

Comparison pair	Category	Sig.	M-W U	Z Value	Median	Result of the TEST
A-B	Understandability	0,01	2	-2,589	Median A=3,125 Median B=1,5	A and B are significantly different, the satisfaction level of A is greater then the satisfaction level of B.
A-C	Understandability	0,049	6	-1,967	Median A=3,3750 Median C=2,0	A and C are significantly different, the satisfaction level of A is greater then the satisfaction level of C.
B-C	Understandability	0,140	9	-1,475	Median B=3,3250 Median C=2,5	B and C are not significantly different.

A: 1-5 years of working experience, B: 6-10 years of working experience, C: 10 years and above working experience.

Understandability is defined as “Do uniquely recognisable functions contain adequate comments so that their purpose is clear?” by WIKIBOOKS (2017). The findings provided in Table 4 indicate that the less experienced group (1-5 years)

found the tool to be more understandable than both of the remaining more experienced groups (6 years or more). No significant differences were obtained between the groups with 6-10 years and more than 10 years experience.

Table 5. Working Experience Groups' Comparisons for "Usability"

Comparison pair	Category	Sig.	M-W U	Z-Value	Median	Result of the TEST
A-B	Usability	0,06	1	-2,756	Median A=3,33 Median B=1,5	A and B are significantly different, the satisfaction level of A is greater then the satisfaction level of B.
A-C	Usability	0,107	8	-1,613	Median A=3,67 Median C=2,0	A and C are not significantly different.
B-C	Usability	0,169	9,5	-1,376	Median B=3,5 Median C=2,5	B and C are not significantly different.

A: 1-5 years of working experience, B: 6-10 years of working experience, C: 10 years and above working experience.

Usability is defined as "Is a GUI used? Is there adequate on-line help?" (WIKIBOOKS, 2017). The results provided in Table 5 show that the less experienced group (1-5 years) found the tool to be more usable than group with 6-10 years experience. Surprisingly, no significant difference was found for any other groups.

Table 6. Working Experience Groups' Comparisons for "Consistency"

Comparison pair	Category	Sign.	M-W U	Z Value	Median	Result of the TEST
A-B	Consistency	0,038	5,5	-2,079	Median A=3,67 Median B=1,5	A and B are significantly different, the satisfaction level of A is greater than the satisfaction level of B.
A-C	Consistency	0,012	2,5	-2,517	Median A=3,5 Median C=2,0	A and C are significantly different, the satisfaction level of A is greater than the satisfaction level of C.
B-C	Consistency	0,007	1,5	-2,704	Median B=3,67 Median C=2,5	B and C are significantly different, the satisfaction level of B is greater than the satisfaction level of C.

A: 1-5 years of working experience, B: 6-10 years of working experience, C: 10 years and above working experience.

Consistency is defined as "Is one variable name used to represent different logical or physical entities in the program?" (WIKIBOOKS, 2017). As seen by the results in Table 6, the less experienced group (1-5 years) found the tool to be more consistent than both of the remaining more experienced groups (6 years or more). In addition, the participants with medium level of experience found the tool to be more consistent than the experienced group.

Table 7. Working Experience Groups' Comparisons for "Ease-of-use"

Comparison pair	Category	Sign.	M-W U	Z Value	Median	Result of the TEST
A-B	Ease-of-use	0,004	0	-2,918	Median A=3,5 Median B=1,5	A and B are significantly different, the satisfaction level of A is greater then the satisfaction level of B.
A-C	Ease-of-use	0,027	4,5	-2,217	Median A=3,75 Median C=2,0	A and C are significantly different, the satisfaction level of A is greater then the satisfaction level of C.
B-C	Ease-of-use	0,195	10	-1,297	Median B=3,5 Median C=2,5	B and C are not significantly different.

A: 1-5 years of working experience, B: 6-10 years of working experience, C: 10 years and above working experience.

Ease-of-use is defined as "Users will be able to successfully complete an activity without needing any external instruction or help screens." (WIKIBOOKS, 2017). The findings provided in Table 7 indicate that the less experienced group (1-5 years) found the tool to be easier to use than both of the remaining more experienced groups (6 years or more). No significant differences were obtained between the groups with 6-10 years and more than 10 years experience.

Table 8. Working Experience Groups' Comparisons for "Productivity"

Comparison pair	Category	Sign.	M-W U	Z Value	Median	Result of the TEST
A-B	Productivity	0,055	6,5	-1,920	Median A=3,415 Median B=1,0	A and B are almost significantly different, the satisfaction level of A is greater than the satisfaction level of B.
A-C	Productivity	0,276	11,5	-1,090	Median A=3,67 Median C=2,0	A and C are not significantly different.
B-C	Productivity	0,154	9,5	-1,425	Median B=4,0 Median C=2,5	B and C are not significantly different.

Productivity is defined as "A measure of the efficiency of a person, machine, factory, system, etc., in converting inputs into useful outputs." (WIKIBOOKS, 2017). The results provided in Table 8 show that the less experienced group (1-5 years) found the tool to be more productive than group with 6-10 years experience. Surprisingly, no significant difference was found for any other groups.

4.4 Discussion

The generic ontology creation tool was evaluated by 25 people with different levels of expertise which were categorized into three groups with 1-5, 6-10, and more than 10 years of expertise. The participants of the evaluation were selected using convenient sampling. The results of the statistical analysis for hypothesis H₁ and H₄ show that the group with the least experience found the tool to be more understandable and easy to use than the rest of the more experienced groups. The participants in the least experienced group are younger than people in the more experienced groups. Therefore, they feel more comfortable with using computer technologies in all aspects of their lives. However, the more experienced groups are more used to traditional and conventional ways of sharing information than the younger subset. Consequently, in terms of understandability and ease-of-use, the less experienced group was the most satisfied.

The results of statistical analysis for hypothesis H₂ and H₅ show that the group with less experience found the tool more usable and productive than the rest of the more experienced groups. In this study, the main objective is to share information in an efficient and effective way. Since the participants in the least experienced group are younger than the rest of the participants, they share information in a different manner than the more experienced group such as using social media applications Facebook, Twitter, Instagram, etc. and mobile applications such as WhatsApp. As such, the younger participants prefer to use a software application instead of the more conventional face-to-face or e-mail approaches for this purpose. Consequently, in terms of usability and productivity, the less experienced group was the most satisfied.

The results of statistical analysis for hypothesis H₃ show that all of the groups found the tool to be consistent. The less experienced group has more satisfaction level than the more experienced groups and 6-10 years of experience group is more satisfied than the most experienced group. In addition, the analysis of the generated ontologies by the participants during the evaluation process resulted in an average

of 86% similarity with the requested ontology output which also supports the consistency results of the survey.

Overall, all groups were satisfied with the system on average. The satisfaction level increased as the experience level decreased for the group averages. To summarize, all hypothesis were accepted and the main objective of the study was fulfilled according to the results of the evaluation.

The research questions stated in the introduction have been answered as follows. For the first research question, in the evaluation phase, 25 participants with no ontology creation expertise were able to create ontologies from scratch in their own domains. In addition, for the second research question, ontologies were created in different domains such as human-resources, marketing, supply management, etc. using this tool.

CHAPTER 5

CONCLUSION

The aim of this thesis is to develop a generic ontology creation tool for domain experts who have no expertise on ontology creation. This tool enables any employee of a firm to create an ontology about his/her domain and share the information in the ontology in machine-readable format with the rest of the company or other companies. With this generic ontology creation tool, many participants in different divisions of a company such as; Supply Management, Finance and Planning, Human Resources, Logistics, etc. can easily share information via the created ontologies. The results of the evaluation show that the participants who had no ontology creation expertise found the tool satisfactory in terms of understandability, usability, consistency, ease-of-use and productivity.

The main purpose of Semantic Web technology is formalizing the web contents as syntactically and semantically correct data to provide a machine-understandable and readable format.

The current web technologies handle many steps in reaching the requested data without the need for any human intervention. Machine-readable formatted web contents enable to make inferences about the requested information by utilizing ontologies. An ontology keeps information and relations about data which is called as the metadata for web pages.

When compared to the initial state of the Web, Web 1.0, the current state of the Web provides more structured and machine-processable data on the Web which are semantically utilized by both humans and computers to reach more relative or exact data to be searched. As such, the search results are more accurate and reliable than the outputs produced by keyword-based search engines.

The process of complex business processes which include many functionalities may involve human interaction to assemble pieces for functionality with ontologies. The human participation with no expertise about ontologies in such activities may cause time delays and loss of information. Our aim is to decrease the time delays and improve the accuracy of data in a business and also to provide a way to keep the information in the company safe by not having to interact with a third party such as an ontology expert.

The developed tool is evaluated with 25 people selected from different business sectors having different work experience levels; 1-5 years, 6-10 years and 10 years and above. Each of the participants are observed while creating an ontology from a given ontology graph, and then creating an ontology about their domain from scratch. Then, the participants are asked to fill a questionnaire about their experiences in using the tool.

The results obtained from questionnaires are evaluated and consolidated by different statistical methods. The results show that for the satisfaction level, all participants rated the tool above the average. According to the comparison of groups in working experiences; the most satisfied group has 1-5 years of work experience, then 6-10 years of experience and lastly 10 years or above experience. These results mostly match with our objectives about this thesis since we focused on the creation of ontology by users who have no expertise about ontology creation.

Observations that are gathered about the tool will be handled as future work which are; a more user-friendly interface design, an extra interface for entering new relation types for resources on different business sectors such as Purchasing, Human-Resources, etc, and more technical capabilities such as selection of language, panel design, themes of color, font size, etc.

Improvements on the Web with linked data and the Semantic Web technology create more structured, meaningful and machine-processable data. Such new developments on the Web will enhance the accuracy, reliability and effectiveness of the new systems in the future.

REFERENCES

- Altova,(2017). <https://www.altova.com/products.html>. (Last visited at: Jun. 14, 2017)
- Anjomshoaa A., Karim S., Shayeganfar F., Tjoa A M., (2006). Exploitation of Semantic Web Technology in ERP Systems, *IFIP TC 8 International Conference on Research and Practical Issues of Enterprise Information Systems (CONFENIS 2006)*, Vienna, Austria, pp 417-427 (April 24–26, 2006).
- Apache Jena, (2017). <https://jena.apache.org/> (Last visit at: Jun. 28, 2017).
- Asuncion, G P. Corcho, O. 2002. Ontology Languages for the Semantic Web. *IEEE Intelligent Systems*. 17 (1). pp . 54 – 60.
- Baader, F, Horrocks, I & Sattler, U. (2005). Description Logics as Ontology Languages for the Semantic Web. *Lecture Notes in Computer Science*. pp. 228 – 248.
- Berners-Lee, T., Hendler J. and Lassila O., (2001). The Semantic Web. *Scientific American*, May 2001: 28-37.
- Bingi, P., M. Sharma, and J. Godla, (1999). Critical issues affecting and ERP implementation. *Information Systems Management*, 16 3: 7-14.
- Blacoe, I., Portabella, D. (2005). Guidelines for the integration of agentbased services and web-based services. Deliverable D2.4.4 (WP2.4), Knowledge Web Project.
- Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., et al. (2004). Web services architecture. W3C working group note. <http://www.w3.org/TR/ws-arch>. (Last visit at: Jun. 28, 2017).
- Buhler, P. A., & Vidal, J. M. (2005). Towards adaptive workflow enactment using multiagent systems. *Information Technology and Management Journal*, 6(1), 61–87.
- Cabral, L., Domingue, J., Galizia, S., Gugliotta, A., Tanasescu, V., Pedrinaci, C., et al. (2006). IRS-III: A Broker for semantic web services based applications. In *Proceedings of the 5th international semantic web conference (ISWC 2006)*.
- Choudry, N., (2014). World Wide Web and Its Journey from Web 1.0 to Web 4.0. *International Journal of Computer Science and Information Technologies*, Vol.5(6), 2014, 8096-8100
- Davies, J., Studer, R., Warren, P. (2006). *Semantic Web Technologies - trends and research in ontology based systems*. England: John Wiley & Sons Ltd.

Ding Y, Fensel, D. Klein, M & Omelyenko, B. 2002. 'The Semantic Web: Yet Another Hip?'. *Data and Knowledge Engineering*, pp.205 – 227.

Digital Enterprise Research Institute (DERI) (2006), <http://www.deri.ie/> (Last visit at: Jun. 28, 2017).

DOMÉ ,(2017). <http://dome.sourceforge.net/>, (Last visit at: Jun. 28, 2017).

Elmasri, R & Navathe, S B. (2003). *Fundamentals of Database Systems*. California: Pearson/Addison Wesley.

Fadel F.G., Fox M.S., Grüninger M. (1994). A generic enterprise resource ontology, in: Proceedings of the Third IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, West Virginia (WET ICE'94), April, (1994), pp.117–128.

Fayez M., Rabelo L. and Mollaghasemi M. (2005). Ontologies for supply chain simulation modeling, *Proceedings of the 2005 Winter Simulation Conference*, p.2364-2370.

Fensel, D., & Bussler, C. (2002). The web service modeling framework WSMF. *Electronic Commerce Research and Applications*, 1(2), 113–137.

Fox M.S., Barbuceanu M., Grüninger M., Lin J.(1998). An organization ontology for enterprise modelling, in: Prietula M., Carley K., Gasser L. (Eds.), *Simulating Organizations: Computational Models of Institutions and Groups*, AAAI/MIT Press, Menlo Park, 1998, pp. 131–152.

Gibbins, N., Harris, S., Shadbolt, N. (2003). Agent-based semantic web services. Proceedings of the 12th international world wide web conference (WWW2003) (pp. 710–717).

Grenon, P. (Ed.). (2003). Reference ontologies and application ontologies. Institute for Formal Ontology and Medical Information Science (IFOMIS) Reports. ISSN 1611-4019.<http://www.ifomis.uni-saarland.de>. (Last visit at: Jun. 28, 2017).

Gruber T. (1995). Toward Principles for the Design of Ontologies Used for Knowledge Sharing, *International Journal of Human-Computer Studies* 43(5-6), pp.907.

Grubic T., Fan I. (2010). Supply chain ontology: Review, analysis and synthesis, Manufacturing Department, School of Applied Sciences, Cranfield University.

Haugen R. and McCarthy W.E. (2000). REA a semantic model for Internet supply chain collaboration, *ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*, 15-19th Oct. 2000 Minneapolis, Minnesota USA.

Hendler, J. (2001). Agents and the semantic web. *IEEE Intelligent Systems*, 16(2), 30–37.

Jacob, E K. 2003. Ontologies and the Semantic Web. *Bulletin of the American Society for Information Science and Technology*. 29 (4). pp. 19 – 22.

Kuter, U., Sirin, E., Parsia, B., Nau, D. S., & Hendler, J. (2005). Information gathering during planning for web service composition. *Journal of Web Semantics*, 3(2–3), 183–205.

Knoodle, (2017). <http://www.knoodl.com/ui/home.html> (Last visit at: Jun. 28, 2017).

Koriat, N., & Gelbard, R. (2014). Knowledge sharing motivation among IT personnel: Integrated model and implications of employment contracts. *International Journal of Information Management*, 34(5), 577e591.

KwangWook, G., & Ravichandran, T. (Jan. 2011). Accessing external Knowledge: Intention of knowledge exchange in virtual community of practice. In , 2011. Paper presented at the system sciences (HICSS), 2011 44th Hawaii international conference on (pp. 4e7).

Lin H.K., Harding J.A., Shahbaz M. (2004). Manufacturing system engineering ontology for semantic interoperability across extended project teams, *International Journal of Production Research* 42 (24) (2004) 5099–5118.

Lin H.K., Harding J.A. (2007). A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration, *Computers in Industry*, 58(5) (2007) 428–437.

Lin J., Fox M.S., Bilgic T.(1996). A requirement ontology for engineering design, in: *Proceedings of the 3rd International Conference on Concurrent Engineering*, 1996, pp. 343–351.

Liu, D., Ji, Y., & Mookerjee, V. (2011). Knowledge sharing and investment decisions in information security. *Decision Support Systems*, 52(1), 95e107. <http://dx.doi.org/10.1016/j.dss.2011.05.007>. (Last visit at: Jun. 28, 2017)

Lyman P, Hal R. Varian. (2003). *How much information?*. Technical Report, UC Berkeley. <http://www.sims.berkeley.edu/research/projects/how-much-info-2003/> (Last visit at: Jun. 28, 2017).

Macris, Aristomenis M., (2011). Enhancing Enterprise Resource Planning users' understanding through ontology-based training, Volume 27, Issue 4, July 2011, Pages 1450-1459.

Minitab tutorial.,(2017). https://www.minitab.com/uploadedFiles/Documents/getting-started/Minitab17_GettingStarted-en.pdf (Last visited at: Jun. 14, 2017).

NeOn toolkit, (2017). http://neon-toolkit.org/wiki/Main_Page.html (Last visited at: Jun. 14, 2017).

OWL (2004). OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004. <http://www.w3.org/TR/owl-features/> (Last visited at: Jun. 14, 2017).

Paolucci, M., & Sycara, K. (2003). Autonomous semantic web services. *IEEE Internet Computing*, 7(5), 34–41.

Park, J. H., Gu, B., Leung, A. C. M., & Konana, P. (2014). An investigation of information sharing and seeking behaviors in online investment communities. *Computers in Human Behavior*, 31(0), 1e12. <http://dx.doi.org/10.1016/j.chb.2013.10.002>.

Protege, (2017). <http://protege.stanford.edu/> (Last visited at: Jun. 14, 2017).

Semantic Web on the Web (SWW), (2017). <http://www.cambridgesemantics.com/semantic-university/semantic-web-web/> (Last visited at: Jun. 14, 2017).

Shafiq, O., Suguri, H., Ali, A., & Fensel, D. (2006). A first step towards enabling Interoperability between software agents and semantic web services: Multi agent systems adapting web services standards. *IBIS – Interoperability in Business Information Systems*, 2(2), 97–117.

Silva, N & Rocha, J. 2003. Semantic Web Complex Ontology Mapping. *IEEE/WIC International Conference on Web Intelligence*. pp. 82 – 88.

Studer, R., Benjamins, R., & Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data and Knowledge Engineering*, 25(1–2), 161–197.

SWSF (2005). Semantic Web Services Framework (SWSF) Overview. W3C Member Submission 9 September 2005. <http://www.w3.org/SWSF/>. (Last visit at: Jun. 28, 2017).

Tamma, V., Phelps, S., Dickinson, I., & Wooldridge, M. (2005). Ontologies for supporting negotiation in e-commerce. *Engineering Applications of Artificial Intelligence*, 18(2005), 223–236.

The Supply-Chain Operations Reference-model (SCOR) (2006). Available from WorldWideWeb:<http://www.supplychain.org/page.wv?%20section=SCOR+Model&name=SCOR+Model> (Last visited at: Jun. 28, 2017).

Walpole, R. E., Myers, R. H., Myers, S. L., Ye, K. (2013). *Essentials of Probability & Statistics for Engineering & Scientists*. Pearson Education, International Edition.

WIKIBOOKS,(2017). https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Quality(Last visited at: Jun. 14, 2017).

W3C, (2017). World Wide Web Consortium. <https://www.w3.org/> (Last visited at: Jun. 14, 2017).

Zhang, J. 2007. Ontology and The Semantic Web. *Proceedings of the North American Symposium on Knowledge Organization*. pp. 9 – 20.

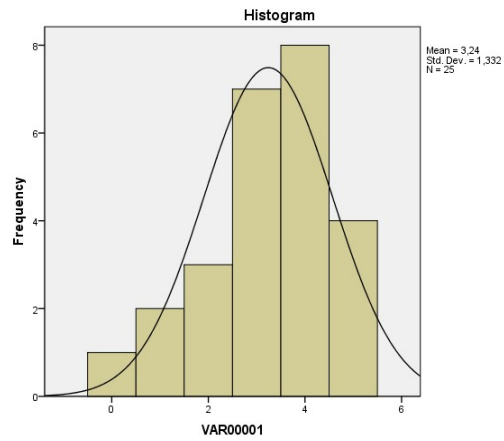


APPENDICES

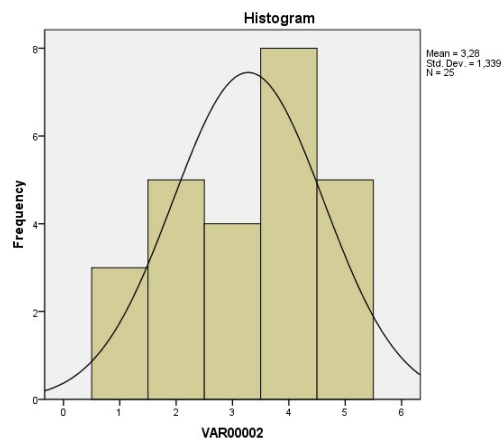


APPENDIX A. Histograms of questionnaire questions of Frequency testing

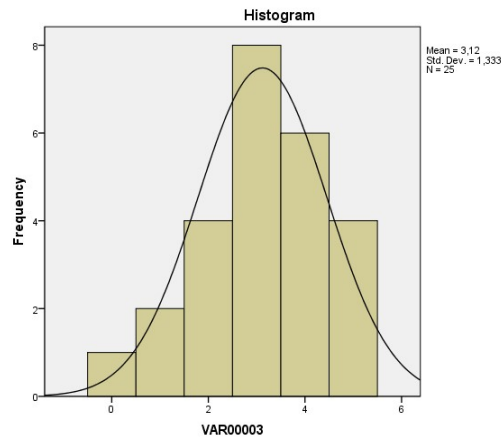
Question 1.a



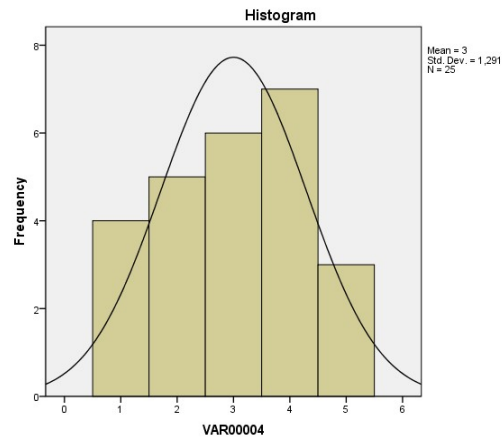
Question 2.a



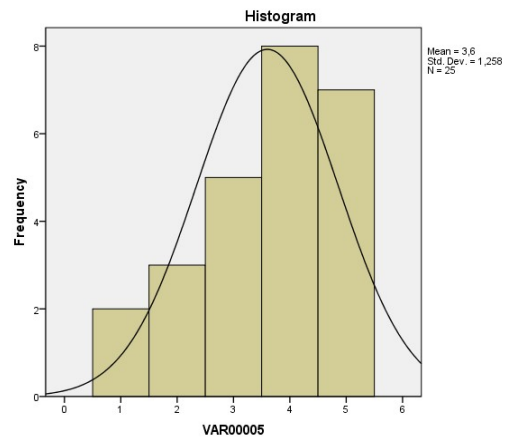
Question 2.b



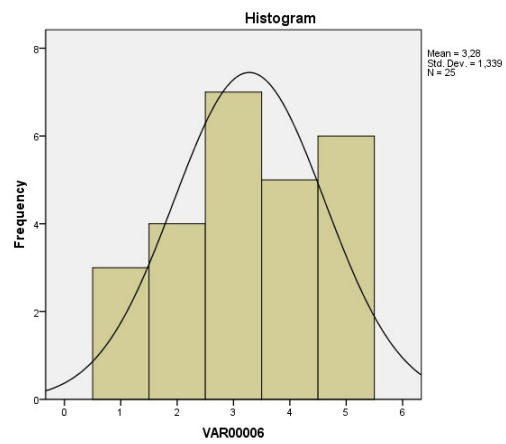
Question 2.c



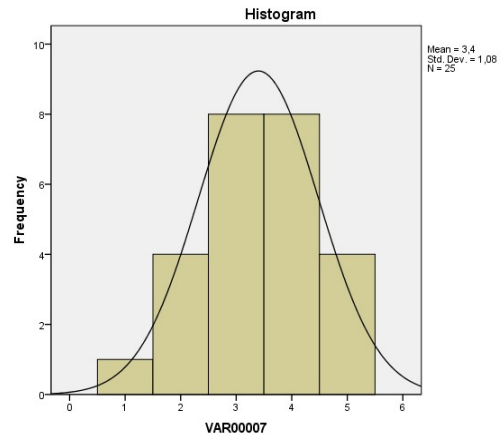
Question 3.a



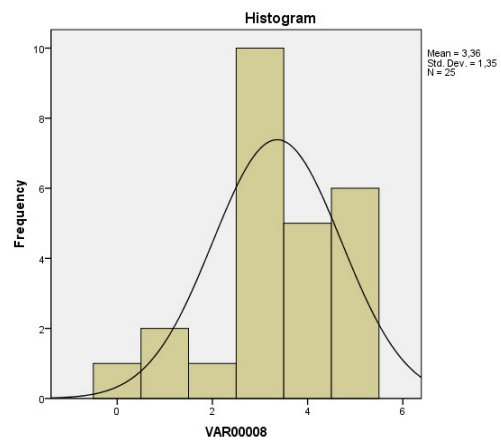
Question 3.b



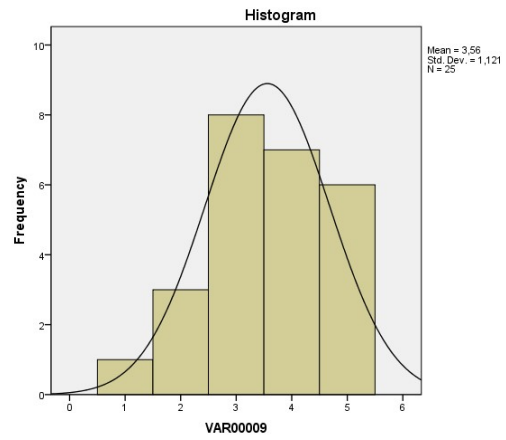
Question 4.a



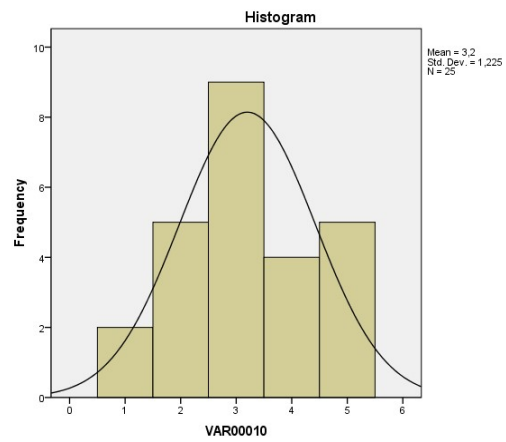
Question 5.a



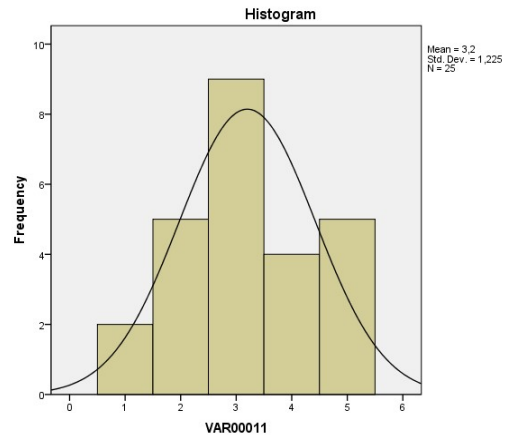
Question 5.b



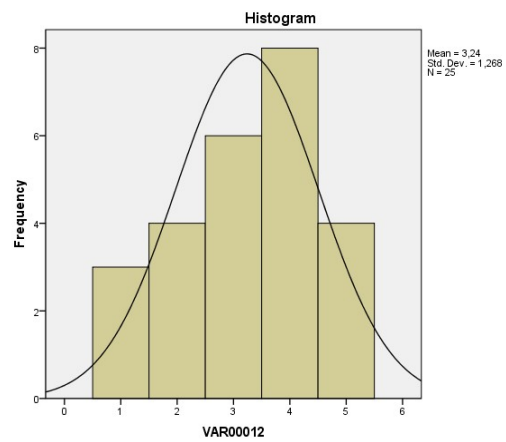
Question 6.a



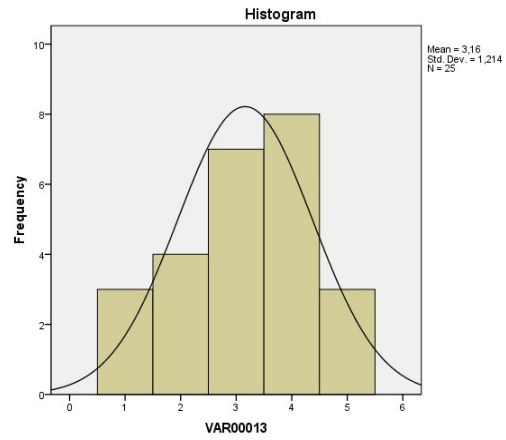
Question 6.b



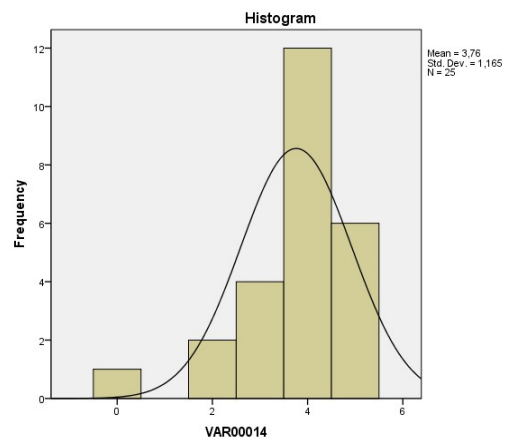
Question 7.a



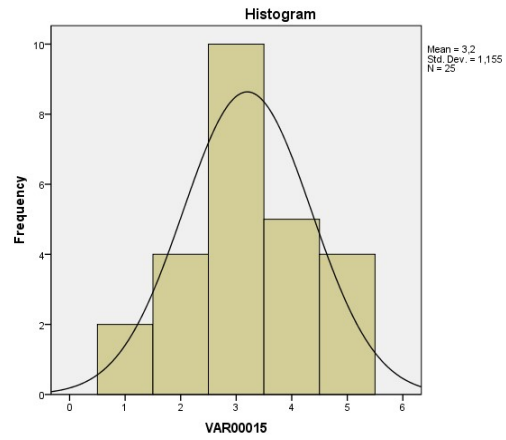
Question 7.b



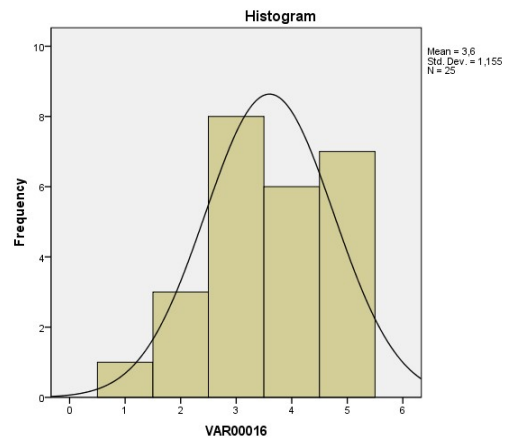
Question 8.a



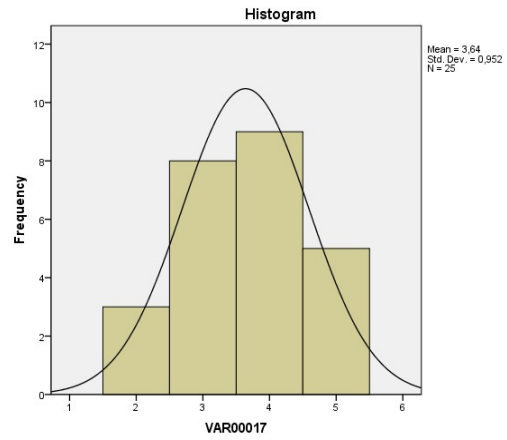
Question 8.b



Question 8.c



Question 8.d



APPENDIX B. Program Code of Tool

```
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.LineBorder;

import com.hp.hpl.jena.datatypes.xsd.XSDDatatype;
import com.hp.hpl.jena.graph.Triple;
import com.hp.hpl.jena.ontology.Individual;
import com.hp.hpl.jena.ontology.OntClass;
import com.hp.hpl.jena.ontology.OntModel;
import com.hp.hpl.jena.ontology.OntModelSpec;
import com.hp.hpl.jena.ontology.OntResource;
import com.hp.hpl.jena.rdf.model.Model;
import com.hp.hpl.jena.rdf.model.ModelFactory;
import com.hp.hpl.jena.rdf.model.Property;
import com.hp.hpl.jena.rdf.model.Resource;
import com.hp.hpl.jena.rdf.model.*;
import com.hp.hpl.jena.sparql.core.Quad;
import com.hp.hpl.jena.tdb.store.DateTimeNode;
import com.hp.hpl.jena.util.FileManager;
import com.hp.hpl.jena.vocabulary.*;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.OutputStream;
import java.io.IOException.*;
import org.apache.jena.riot.writer.WriterStreamRDFBase;
import java.io.*;

import java.awt.*;
import java.awt.event.*;
import java.util.Iterator;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*;
import javax.swing.border.LineBorder;

import javax.swing.JTextArea;
```

```

import javax.swing.ImageIcon;

class ImagePanel extends JPanel {

private Image img;

public ImagePanel(String img) {
this(new ImageIcon(img).getImage());
}

public ImagePanel(Image img) {
this.img = img;
Dimension size = new Dimension(img.getWidth(null), img.getHeight(null));
setPreferredSize(size);
setMinimumSize(size);
setMaximumSize(size);
setSize(size);
setLayout(null);
}

public void paintComponent(Graphics g) {
g.drawImage(img, 0, 0, null);
}

}

public class MyClass
{
static String inputFileName;
static String personURI;

static final String relationshipURI = "http://purl.org/vocab/relationship/";

static Model model = ModelFactory.createDefaultModel();

static Property isA;
static Property hasA;
static Property subclassOf;
static Property superClassOf;
static Property locatedIn;
static Property specifies;
static Property actAs;
static Property isAssigned;
static Property performedAt;

static List<String> relationNameList = new ArrayList<String>();

```

```

// Field members
static JPanel panel = new JPanel ();
static JPanel panel1 = new JPanel();
static JPanel panel2 = new JPanel();
static JPanel panel2_1 = new JPanel();
static JPanel panel3 = new JPanel();
static Integer indexer = 0;
static List<JLabel> listOfLabels = new ArrayList<JLabel>();
static List<JLabel> listOfLabelsValue = new ArrayList<JLabel>();
static List<JTextField> listOfTextFields = new ArrayList<JTextField>();
static List<JTextField> listOfTextFieldsValue = new ArrayList<JTextField>();

private static JTextField txtSONUC;
private static JTextField txtMetaModel;
private static JTextField txtModelName;

static Integer pvindex = 1;

static Integer lblindex = 0;
static JLabel lbl = new JLabel();

static List<String> listRelSet1 = new ArrayList<String>();
static List<String> listRelSet2 = new ArrayList<String>();
static List<String> listRelSet3 = new ArrayList<String>();

static List<String> listRelSet1_gostermek = new ArrayList<String>();
static List<String> listRelSet2_gostermek = new ArrayList<String>();

static JPanel panel2_1_relations = new JPanel();

static int lblRelConstrY = 0;

//List<Resource> resList = new ArrayList<Resource>();

//static Model model = ModelFactory.createOntologyModel();
//static Resource res;

static List<Model> modellist = new ArrayList<Model>();

static List<String> resourceNameList = new ArrayList<String>();
static List<String> resourceOnlyNameList = new ArrayList<String>();
static JComboBox res1ComboBox = new JComboBox();
static JComboBox res2ComboBox = new JComboBox();
static JComboBox relTypeComboBox = new JComboBox();

//SONUC RDF INI GOSTEREN TEXTAREA
private static JTextArea rdfoutput = new JTextArea();

public static void main(String[] args)
{
//RELATION TIPLERINI GIR
Resource resourceNamespace = model.createResource(relationshipURI);
model.setNsPrefix("rela", relationshipURI);
isA = model.createProperty(relationshipURI,"isA");

```

```

hasA = model.createProperty(relationshipURI,"hasA");
subClassOf = model.createProperty(relationshipURI,"subClassOf");
superClassOf = model.createProperty(relationshipURI,"superClassOf");
locatedIn = model.createProperty(relationshipURI,"locatedIn");
specifies = model.createProperty(relationshipURI,"specifies");
actAs = model.createProperty(relationshipURI,"actAs");
IsAssigned = model.createProperty(relationshipURI,"IsAssigned");
performedAt = model.createProperty(relationshipURI,"performedAt");

```

```

relationNameList.add("isA");
relationNameList.add("hasA");
relationNameList.add("subClassOf");
relationNameList.add("superClassOf");
relationNameList.add("locatedIn");
relationNameList.add("specifies");
relationNameList.add("actAs");
relationNameList.add("IsAssigned");
relationNameList.add("performedAt");

```

```

//PANEL 1
panel1.setPreferredSize(new Dimension(600, 600));
//panel1.setLayout(new GridBagLayout());
panel1.setBorder(BorderFactory.createBlackLineBorder());

```

```

JPanel panel_baslik = new JPanel();
panel_baslik.add(new JLabel("ONTOLOGY CREATION TOOL"));
panel1.add(panel_baslik, BorderLayout.NORTH);

```

```

JPanel panel_ice = new JPanel();
txtMetaModel = new JTextField("",30);
panel_ice.add(new JLabel("Create Ontology About:"));
panel_ice.add(txtMetaModel);

```

```

panel1.add(panel_ice, BorderLayout.SOUTH);

```

```

JButton next1 = new JButton("Next>>");
next1.addActionListener(new Next1_ButtonListener());
panel1.add(next1);

```

```

//PANEL2
BorderLayout layout = new BorderLayout ();
panel2.setLayout(layout);
panel2.setPreferredSize(new Dimension(600, 600));
panel2.setBorder(BorderFactory.createBlackLineBorder());

```

```

JPanel pnl2_ust = new JPanel();
pnl2_ust.setLayout(new FlowLayout());

```

```

JButton clearButton = new JButton("Clear");
clearButton.addActionListener(new Clear_ButtonListener());
pnl2_ust.add(clearButton);

```

```

JButton next2 = new JButton("Next>>");

```

```

next2.addActionListener(new Next2_ButtonListener());
pnl2_ust.add(next2);

panel2.add(pnl2_ust, BorderLayout.NORTH);

JPanel pnl2_orta = new JPanel();
pnl2_orta.setLayout(new FlowLayout());

txtModelName = new JTextField("", 15);
pnl2_orta.add(new JLabel("Resource Name:"));

pnl2_orta.add(txtModelName);

JButton addButton = new JButton("Add Property");
addButton.addActionListener(new Add_ButtonListener());
pnl2_orta.add(addButton);

JButton showButton = new JButton("Add Resource to Ontology");
showButton.addActionListener(new Add_Resource_to_Ontology_ButtonListener());
pnl2_orta.add(showButton);

pnl2_orta.add(lbl);

pnl2_orta.add(panel);

panel2.add(pnl2_orta, BorderLayout.CENTER);

panel.setPreferredSize(new Dimension(590, 500));
panel.setLayout(new GridBagLayout());
//panel.setBorder(BorderFactory.createBlackLineBorder());

panel2.setVisible(false);

//PANEL 2_1 RELATION EKLEME
panel2_1.setPreferredSize(new Dimension(800, 600));
panel2_1.setLayout(new BorderLayout(2, 2));
panel2_1.setBorder(BorderFactory.createBlackLineBorder());

JLabel lblres1 = new JLabel("1. Resource");
JLabel lblres2 = new JLabel("2. Resource");
JLabel lblrel = new JLabel("Relation Type");
JButton relSet = new JButton("Set Relation");
relSet.addActionListener(new relSet_ButtonListener());

JPanel panel_next2_1 = new JPanel();
panel_next2_1.setLayout(new FlowLayout());
JButton next2_1 = new JButton("Next>>");
next2_1.addActionListener(new Next2_1_ButtonListener());
panel_next2_1.add(next2_1);

panel2_1.add(panel_next2_1, BorderLayout.BEFORE_FIRST_LINE);

```

```

JPanel panel2_1_inner = new JPanel();
panel2_1_inner.setLayout(new FlowLayout());

relTypeComboBox.addItem("isA");
relTypeComboBox.addItem("hasA");
relTypeComboBox.addItem("subClassOf");
relTypeComboBox.addItem("superClassOf");
relTypeComboBox.addItem("locatedIn");
relTypeComboBox.addItem("specifies");
relTypeComboBox.addItem("actAs");
relTypeComboBox.addItem("IsAssigned");
relTypeComboBox.addItem("performedAt");

panel2_1_inner.add(lblres1);
panel2_1_inner.add(res1ComboBox);
panel2_1_inner.add(lblrel);
panel2_1_inner.add(relTypeComboBox);
panel2_1_inner.add(lblres2);
panel2_1_inner.add(res2ComboBox);
panel2_1_relations.setLayout(new GridBagLayout());
//panel2_1_inner.add(panel2_1_relations);

panel2_1_inner.add(relSet);

//panel2_1_relations.add(new JLabel("aaaaaaaaaaaaa"));
panel2_1.add(panel2_1_inner, BorderLayout.CENTER);
panel2_1.add(panel2_1_relations, BorderLayout.SOUTH);

panel2_1.setVisible(false);

//PANEL 3
panel3.setPreferredSize(new Dimension(600, 600));
panel3.setLayout(new BorderLayout(5,5));
panel3.setBorder(LineBorder.createBlackLineBorder());

JPanel jpanel3butonlar = new JPanel();

JButton rdfButton = new JButton("Generate RDF");
rdfButton.addActionListener(new Generate_RDF_ButtonListener());
jpanel3butonlar.add(rdfButton);

JButton newrdfButton = new JButton("New RDF");
newrdfButton.addActionListener(new newrdfButton_ButtonListener());
jpanel3butonlar.add(newrdfButton);

JButton filerdfButton = new JButton("Write RDF to a File");
filerdfButton.addActionListener(new filerdfButton_ButtonListener());
jpanel3butonlar.add(filerdfButton);

panel3.add(jpanel3butonlar, BorderLayout.BEFORE_FIRST_LINE);

```

```
rdfoutput.setSize(500, 500);
panel3.add(rdfoutput, BorderLayout.CENTER);
panel3.setVisible(false);
```

```
FlowLayout experimentLayout = new FlowLayout();
// Construct frame
JFrame frame = new JFrame();
frame.setLayout(experimentLayout);
frame.setPreferredSize(new Dimension(990, 990));
frame.setTitle("Ontology Creation Automation System");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Frame constraints
GridBagConstraints frameConstraints = new GridBagConstraints();

// Add panel to frame
frameConstraints.gridx = 0;
frameConstraints.gridy = 1;
frameConstraints.weighty = 1;
frame.add(panel1);
frame.add(panel2);
frame.add(panel2_1);
frame.add(panel3);

// Pack frame
frame.pack();

// Make frame visible
frame.setVisible(true);

JMenuBar menubar = new JMenuBar();
JMenu help = new JMenu("Help");
//JMenu onto = new JMenu("Ontology");

JMenuItem eMenuItem1 = new JMenuItem("Resource");
eMenuItem1.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent event) {
ImagePanel panel_Resource = new ImagePanel(new
ImageIcon(this.getClass().getResource("/images/Resource.png")).getImage());

JFrame frame1 = new JFrame();
frame1.setSize(640, 395);
frame1.setTitle("Resource");
frame1.getContentPane().add(panel_Resource);
frame1.setVisible(true);
}
});
```

```

JMenuItem eMenuItem2 = new JMenuItem("Property");
eMenuItem2.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent event) {

ImagePanel panel_Property = new ImagePanel(new
ImageIcon(this.getClass().getResource("/images/Property.png")).getImage());

JFrame frame2 = new JFrame();
frame2.setSize(620, 310);
frame2.setTitle("Property");
frame2.getContentPane().add(panel_Property);

frame2.setVisible(true);
}
});

JMenuItem eMenuItem3 = new JMenuItem("Relation");
eMenuItem3.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent event) {
ImagePanel panel_Relation = new ImagePanel(new
ImageIcon(this.getClass().getResource("/images/Relation.png")).getImage());

JFrame frame3 = new JFrame();
frame3.setSize(615, 458);
frame3.setTitle("Relation");
frame3.getContentPane().add(panel_Relation);
frame3.setVisible(true);
}
});
JMenuItem onto = new JMenuItem("What is an Ontology?");
onto.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent event) {
ImagePanel panel_Ontology = new ImagePanel(new
ImageIcon(this.getClass().getResource("/images/Ontology.png")).getImage());

JFrame frame4 = new JFrame();
frame4.setSize(605, 158);
frame4.setTitle("What is an Ontology?");
frame4.getContentPane().add(panel_Ontology);
frame4.setVisible(true);
}
});

help.add(onto);
help.add(eMenuItem1);
help.add(eMenuItem2);
help.add(eMenuItem3);

//menubar.add(onto);

```

```

menubar.add(help);
frame.setJMenuBar(menubar);
}

static class Add_ButtonListener implements ActionListener
{
@Override
public void actionPerformed(ActionEvent arg0)
{
// Clear panel
panel.removeAll();

// Create label and text field
JTextField jTextField = new JTextField();
jTextField.setSize(100, 100);
listOfTextFields.add(jTextField);
listOfLabels.add(new JLabel("Property :" + pvindex));

jTextField = new JTextField();
jTextField.setSize(100, 100);
listOfTextFieldsValue.add(jTextField);
listOfLabelsValue.add(new JLabel("Value :" + pvindex));

pvindex++;

// Create constraints
GridBagConstraints textFieldConstraints = new GridBagConstraints();
GridBagConstraints labelConstraints = new GridBagConstraints();

GridBagConstraints textFieldConstraintsValue = new GridBagConstraints();
GridBagConstraints labelConstraintsValue = new GridBagConstraints();

// Add labels and text fields
for(int i = 0; i <= indexer; i++)
{
// Text field constraints
textFieldConstraints.gridx = 1;
textFieldConstraints.fill = GridBagConstraints.HORIZONTAL;
textFieldConstraints.weightx = 0.5;
textFieldConstraints.insets = new Insets(10, 10, 10, 10);
textFieldConstraints.gridy = i;

textFieldConstraintsValue.gridx = 3;
textFieldConstraintsValue.fill = GridBagConstraints.HORIZONTAL;
textFieldConstraintsValue.weightx = 0.5;
textFieldConstraintsValue.insets = new Insets(10, 10, 10, 10);
textFieldConstraintsValue.gridy = i;

// Label constraints
labelConstraints.gridx = 0;
labelConstraints.gridy = i;
labelConstraints.insets = new Insets(10, 10, 10, 10);
}
}

```

```

labelConstraintsValue.gridx = 2;
labelConstraintsValue.gridy = i;
labelConstraintsValue.insets = new Insets(10, 10, 10, 10);

// Add them to panel
panel.add(listOfLabels.get(i), labelConstraints);
panel.add(listOfTextFields.get(i), textFieldConstraints);

panel.add(listOfLabelsValue.get(i), labelConstraintsValue);
panel.add(listOfTextFieldsValue.get(i), textFieldConstraintsValue);
}

// Align components top-to-bottom
GridBagConstraints c = new GridBagConstraints();
c.gridx = 0;
c.gridy = indexer+1;
c.weighty = 1;
panel.add(new JLabel(), c);

// Increment indexer
indexer++;
panel.updateUI();

}
}

static class Add_Resource_to_Ontology_ButtonListener implements ActionListener
{
@Override
public void actionPerformed(ActionEvent arg0)
{

JTextField tf;
JTextField tfValue;

personURI = "http://" + txtMetaModel.getText() + "/" + txtModelName.getText(); //ontology
about text buraya
resourceNameList.add(personURI);
resourceOnlyNameList.add(txtModelName.getText());
Resource res = model.createResource(personURI);

for(int i = 0; i <= indexer-1 ; i++)
{
tf = (JTextField)listOfTextFields.get(i);
tfValue = (JTextField)listOfTextFieldsValue.get(i);

res.addProperty(RDFS.label, model.createLiteral(tfValue.getText(), tf.getText()));

```

```

}

modellist.add(model);

txtModelName.setText("");

panel.removeAll();
panel.updateUI();

indexer = 0;
listOfLabels.clear();
listOfTextFields.clear();
listOfLabelsValue.clear();
listOfTextFieldsValue.clear();

lblindex++;
lbl.setText("Resource " + lblindex + " added.");

pvindex=1;
}
}

static class Clear_ButtonListener implements ActionListener
{
@Override
public void actionPerformed(ActionEvent arg0)
{
//txtMetaModel.setText("");
txtModelName.setText("");

panel.removeAll();
panel.updateUI();

indexer = 0;
listOfLabels.clear();
listOfTextFields.clear();
listOfLabelsValue.clear();
listOfTextFieldsValue.clear();
pvindex=1;
}
}

static class Generate_RDF_ButtonListener implements ActionListener
{
@Override
public void actionPerformed(ActionEvent arg0)
{
if(modellist.size()>=2)
{
Model model1 = (Model)modellist.get(0);
Model model2 = (Model)modellist.get(1);
Model modeli;
Model modelsonuc = model1.union(model2);

```

```

for(int i=2;i<modellist.size();i++)
{
modeli = modellist.get(i);
modelsonuc = modelsonuc.union(modeli);
}

modelsonuc.write(System.out);

String sss = "";
Writer wr = new StringWriter();
modelsonuc.write(wr);
sss = wr.toString();
rdfoutput.setText(sss);
}
else if(modellist.size()>=1)
{
Model model1 = (Model)modellist.get(0);

model1.write(System.out);

String sss = "";
Writer wr = new StringWriter();
model1.write(wr);
sss = wr.toString();
rdfoutput.setText(sss);

}
}
}

static class Next1_ButtonListener implements ActionListener
{
@Override
public void actionPerformed(ActionEvent arg0)
{
panel1.setVisible(false);
panel2.setVisible(true);
}
}

static class Next2_ButtonListener implements ActionListener
{
@Override
public void actionPerformed(ActionEvent arg0)
{
for(int i = 0;i<resourceOnlyNameList.size();i++)
{
res1ComboBox.addItem((String)resourceOnlyNameList.get(i));
res2ComboBox.addItem((String)resourceOnlyNameList.get(i));
}
panel2.setVisible(false);
panel2_1.setVisible(true);
}
}

```

```
}  
}
```

```
static class Next2_1_ButtonListener implements ActionListener  
{  
    @Override  
    public void actionPerformed(ActionEvent arg0)  
    {  
        String rel = "";  
        String res1 = "";  
        String res2 = "";  
  
        //relationlari set et  
        for(int i = 0;i<listRelSet1.size();i++)  
        {  
            res1 = listRelSet1.get(i);  
            res2 = listRelSet2.get(i);  
            rel = listRelSet3.get(i);  
  
            Resource r1 = model.getResource(res1);  
            Resource r2 = model.getResource(res2);  
  
            Statement st;  
            if(rel == "isA")  
                st = model.createStatement(r1, isA, r2);  
            else if(rel == "hasA")  
                st = model.createStatement(r1, hasA, r2);  
            else if(rel == "subClassOf")  
                st = model.createStatement(r1, subClassOf, r2);  
            else if(rel == "locatedIn")  
                st = model.createStatement(r1, locatedIn, r2);  
            else if(rel == "specifies")  
                st = model.createStatement(r1, specifies, r2);  
            else if(rel == "actAs")  
                st = model.createStatement(r1, actAs, r2);  
            else if(rel == "IsAssigned")  
                st = model.createStatement(r1, IsAssigned, r2);  
            else if(rel == "performedAt")  
                st = model.createStatement(r1, performedAt, r2);  
            else  
                st = model.createStatement(r1, superClassOf, r2);  
            model.add(st);  
        }  
  
        listRelSet1.clear();  
        listRelSet2.clear();  
        listRelSet3.clear();  
  
        //sonraki sayfaya git  
        panel2_1.setVisible(false);  
        panel3.setVisible(true);  
    }  
}
```

```

}

static class relSet_ButtonListener implements ActionListener
{
@Override
public void actionPerformed(ActionEvent arg0)
{
int rint1 = res1ComboBox.getSelectedIndex();
int rint2 = res2ComboBox.getSelectedIndex();
String rel = (String)relTypeComboBox.getSelectedItem();

String res1 = resourceNameList.get(rint1);
String res2 = resourceNameList.get(rint2);

//listeye ekle
listRelSet1.add(res1);
listRelSet2.add(res2);
listRelSet3.add(rel);

//relation i gostermek icin
String rstr1 = (String)res1ComboBox.getItemAt(rint1);
String rstr2 = (String)res2ComboBox.getItemAt(rint2);
listRelSet1_gostermek.add(rstr1);
listRelSet2_gostermek.add(rstr2);

//paneli ve listeyi guncelle
FillAndUpdateRelationsPanelAndList();

res1ComboBox.setSelectedIndex(0);
res2ComboBox.setSelectedIndex(0);
relTypeComboBox.setSelectedIndex(0);
}
}

static class relSil_ButtonListener implements ActionListener
{
@Override
public void actionPerformed(ActionEvent e)
{
JButton o = (JButton)e.getSource();
String name = o.getName();
String indexstr = name.substring(4);
int index = Integer.parseInt(indexstr);

//index i listelerden cikart
listRelSet1.remove(index);
listRelSet2.remove(index);
listRelSet3.remove(index);

listRelSet1_gostermek.remove(index);
listRelSet2_gostermek.remove(index);

//paneli ve listeyi guncelle

```

```

FillAndUpdateRelationsPanelAndList();
}
}

private static void FillAndUpdateRelationsPanelAndList()
{
String rel;
String res1;
String res2;

String relationStr;

//paneldekileri sil hepsini tekrar yukle
panel2_1_relations.removeAll();
GridBagConstraints labelConstraints = new GridBagConstraints();
JButton btn = new JButton("Sil");
btn.addActionListener(new relSil_ButtonListener());
labelConstraints.gridx=0;
labelConstraints.gridy=lblRelConstrY;
panel2_1_relations.add(new JLabel("RELATION LIST:"), labelConstraints);
lblRelConstrY++;

for(int i = 0;i<listRelSet1.size();i++)
{
res1 = listRelSet1_gosterme.get(i);
res2 = listRelSet2_gosterme.get(i);
rel = listRelSet3.get(i);

relationStr = (i+1) + ". Relation: " + res1 + " " + rel + " " + res2 + " ";

btn = new JButton("Sil");
btn.setName("btn_" + i);
btn.addActionListener(new relSil_ButtonListener());

labelConstraints.gridx=0;
labelConstraints.gridy=lblRelConstrY;
panel2_1_relations.add(new JLabel(relationStr), labelConstraints);

labelConstraints.gridx=1;
labelConstraints.gridy=lblRelConstrY;
panel2_1_relations.add(btn, labelConstraints);

//panel2_1_relations.updateUI();
lblRelConstrY++;
}
panel2_1_relations.updateUI();
}

static class newrdfButton_ButtonListener implements ActionListener
{
@Override
public void actionPerformed(ActionEvent arg0)
{

```

```

txtModelName.setText("");
txtMetaModel.setText("");

panel.removeAll();
panel.updateUI();

indexer = 0;
listOfLabels.clear();
listOfTextFields.clear();
listOfLabelsValue.clear();
listOfTextFieldsValue.clear();

modellist.clear();

lblindex = 0;
lbl.setText("");

//relations lari sifirla
listRelSet1.clear();
listRelSet2.clear();
listRelSet3.clear();
listRelSet1_gostermek.clear();
listRelSet2_gostermek.clear();
FillAndUpdateRelationsPanelAndList();

panel2.setVisible(false);
panel3.setVisible(false);
panel1.setVisible(true);
}
}
static class filerdfButton_ButtonListener implements ActionListener
{
@Override
public void actionPerformed(ActionEvent arg0)
{

String fileName = "Ekrem.rdf";
FileWriter out = null;
try {
out = new FileWriter( fileName );
} catch (IOException ex) {
Logger.getLogger(MyClass.class.getName()).log(Level.SEVERE, null, ex);
}
try
{
model.write( out, "RDF/XML-ABBREV" );
}
} finally
{
try
{

```

```
out.close();
}
catch (IOException closeException)
{
// ignore
}
}

/*
txtModelName.setText("");
txtMetaModel.setText("");

panel.removeAll();
panel.updateUI();

indexer = 0;
listOfLabels.clear();
listOfTextFields.clear();
listOfLabelsValue.clear();
listOfTextFieldsValue.clear();

modellist.clear();

lblindex = 0;
lbl.setText("");

panel2.setVisible(false);
panel3.setVisible(false);
panel1.setVisible(true);
*/
}

}
}
```

APPENDIX C. Statistical Analysis Results

- A: 1-5 years of working experience
- B: 6-10 years of working experience
- C: 10 years and above working experience

Individual Comparison:

A-B, Users: 1-19

→ NPar Tests

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	34	3,2059	1,32068	,00	5,00	2,0000	3,0000	4,0000
VAR00003	34	1,5000	,50752	1,00	2,00	1,0000	1,5000	2,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	1,00	17	23,41	398,00
	2,00	17	11,59	197,00
	Total	34		

Test Statistics^b

	VAR00002
Mann-Whitney U	44,000
Wilcoxon W	197,000
Z	-3,556
Asymp. Sig. (2-tailed)	,000
Exact Sig. [2*(1-tailed Sig.)]	,000 ^a

a. Not corrected for ties.

Individual Comparison: A-C, Users: 1-23

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00001	34	3,5882	1,13131	,00	5,00	3,0000	4,0000	4,0000
VAR00002	34	2,0000	1,01504	1,00	3,00	1,0000	2,0000	3,0000

Mann-Whitney Test

Ranks

	VAR00002	N	Mean Rank	Sum of Ranks
VAR00001	1,00	17	20,71	352,00
	3,00	17	14,29	243,00
	Total	34		

Test Statistics^b

	VAR00001
Mann-Whitney U	90,000
Wilcoxon W	243,000
Z	-1,961
Asymp. Sig. (2-tailed)	,050
Exact Sig. [2*(1-tailed Sig.)]	,062 ^a

a. Not corrected for ties.

Individual Comparison: B-C, User: 17-25

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	34	3,1176	1,17460	1,00	5,00	2,0000	3,0000	4,0000
VAR00003	34	2,5000	,50752	2,00	3,00	2,0000	2,5000	3,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	2,00	17	21,44	364,50
	3,00	17	13,56	230,50
	Total	34		

Test Statistics^b

	VAR00002
Mann-Whitney U	77,500
Wilcoxon W	230,500
Z	-2,391
Asymp. Sig. (2-tailed)	,017
Exact Sig. [2*(1-tailed Sig.)]	,020 ^a

a. Not corrected for ties.

Understandability group averages:

A-B

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,2708	,65243	2,25	4,25	2,8125	3,1250	3,8750
VAR00003	12	1,5000	,52223	1,00	2,00	1,0000	1,5000	2,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	1,00	6	9,17	55,00
	2,00	6	3,83	23,00
	Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	2,000
Wilcoxon W	23,000
Z	-2,589
Asymp. Sig. (2-tailed)	,010
Exact Sig. [2*(1-tailed Sig.)]	,009 ^a

a. Not corrected for ties.

A-C

→ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,4375	,57530	2,50	4,25	3,0000	3,3750	4,0000
VAR00003	12	2,0000	1,04447	1,00	3,00	1,0000	2,0000	3,0000

Mann-Whitney Test

Ranks

VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	1,00	6	8,50
	3,00	6	4,50
Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	6,000
Wilcoxon W	27,000
Z	-1,967
Asymp. Sig. (2-tailed)	,049
Exact Sig. [2*(1-tailed Sig.)]	,065 ^a

a. Not corrected for ties.

B-C

→ NPar Tests

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,2917	,42417	2,50	4,00	3,0000	3,2500	3,6875
VAR00003	12	2,5000	,52223	2,00	3,00	2,0000	2,5000	3,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	2,00	6	8,00	48,00
	3,00	6	5,00	30,00
	Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	9,000
Wilcoxon W	30,000
Z	-1,475
Asymp. Sig. (2-tailed)	,140
Exact Sig. [2*(1-tailed Sig.)]	,180 ^a

a. Not corrected for ties.

Usability

A-B

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,4725	,78527	2,00	4,67	3,0000	3,3300	4,0000
VAR00003	12	1,5000	,52223	1,00	2,00	1,0000	1,5000	2,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	1,00	6	9,33	56,00
	2,00	6	3,67	22,00
	Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	1,000
Wilcoxon W	22,000
Z	-2,756
Asymp. Sig. (2-tailed)	,006
Exact Sig. [2*(1-tailed Sig.)]	,004 ^a

a. Not corrected for ties.

A-C

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,6950	,82324	2,33	5,00	3,0825	3,6700	4,5025
VAR00003	12	2,0000	1,04447	1,00	3,00	1,0000	2,0000	3,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	1,00	6	8,17	49,00
	3,00	6	4,83	29,00
	Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	8,000
Wilcoxon W	29,000
Z	-1,613
Asymp. Sig. (2-tailed)	,107
Exact Sig. [2*(1-tailed Sig.)]	,132 ^a

a. Not corrected for ties.

B-C

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,5825	,75368	2,33	5,00	3,0825	3,5000	4,2475
VAR00003	12	2,5000	,52223	2,00	3,00	2,0000	2,5000	3,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	2,00	6	7,92	47,50
	3,00	6	5,08	30,50
	Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	9,500
Wilcoxon W	30,500
Z	-1,376
Asymp. Sig. (2-tailed)	,169
Exact Sig. [2*(1-tailed Sig.)]	,180 ^a

a. Not corrected for ties.

Consistency

A-B

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,5550	,50025	2,33	4,33	3,3300	3,6700	3,9175
VAR00003	12	1,5000	,52223	1,00	2,00	1,0000	1,5000	2,0000

Mann-Whitney Test

Ranks

VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	1,00	6	8,58
	2,00	6	4,42
Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	5,500
Wilcoxon W	26,500
Z	-2,079
Asymp. Sig. (2-tailed)	,038
Exact Sig. [2*(1-tailed Sig.)]	,041 ^a

a. Not corrected for ties.

A-C

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,2783	,84968	1,67	4,33	3,0000	3,5000	3,9175
VAR00003	12	2,0000	1,04447	1,00	3,00	1,0000	2,0000	3,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	1,00	6	9,08	54,50
	3,00	6	3,92	23,50
	Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	2,500
Wilcoxon W	23,500
Z	-2,517
Asymp. Sig. (2-tailed)	,012
Exact Sig. [2*(1-tailed Sig.)]	,009 ^a

a. Not corrected for ties.

B-C

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,3067	,85720	1,67	4,33	3,0000	3,6700	3,9175
VAR00003	12	2,5000	,52223	2,00	3,00	2,0000	2,5000	3,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	2,00	6	9,25	55,50
	3,00	6	3,75	22,50
	Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	1,500
Wilcoxon W	22,500
Z	-2,704
Asymp. Sig. (2-tailed)	,007
Exact Sig. [2*(1-tailed Sig.)]	,004 ^a

a. Not corrected for ties.

Ease-of-use

A-B

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,3542	,76469	2,25	4,50	2,5625	3,5000	4,0000
VAR00003	12	1,5000	,52223	1,00	2,00	1,0000	1,5000	2,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	1,00	6	9,50	57,00
	2,00	6	3,50	21,00
	Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	,000
Wilcoxon W	21,000
Z	-2,918
Asymp. Sig. (2-tailed)	,004
Exact Sig. [2*(1-tailed Sig.)]	,002 ^a

a. Not corrected for ties.

A-C

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,6458	,58832	2,25	4,50	3,3125	3,7500	4,0000
VAR00003	12	2,0000	1,04447	1,00	3,00	1,0000	2,0000	3,0000

Mann-Whitney Test

Ranks

VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	1,00	6	8,75
	3,00	6	4,25
Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	4,500
Wilcoxon W	25,500
Z	-2,217
Asymp. Sig. (2-tailed)	,027
Exact Sig. [2*(1-tailed Sig.)]	,026 ^a

a. Not corrected for ties.

B-C

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,5208	,57858	2,25	4,25	3,2500	3,5000	4,0000
VAR00003	12	2,5000	,52223	2,00	3,00	2,0000	2,5000	3,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	2,00	6	7,83	47,00
	3,00	6	5,17	31,00
	Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	10,000
Wilcoxon W	31,000
Z	-1,297
Asymp. Sig. (2-tailed)	,195
Exact Sig. [2*(1-tailed Sig.)]	,240 ^a

a. Not corrected for ties.

Productivity

A-B

→ NPar Tests

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,6117	,69278	1,67	4,33	3,4150	3,6700	4,0000
VAR00003	12	1,5000	,52223	1,00	2,00	1,0000	1,5000	2,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	1,00	6	8,42	50,50
	2,00	6	4,58	27,50
	Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	6,500
Wilcoxon W	27,500
Z	-1,920
Asymp. Sig. (2-tailed)	,055
Exact Sig. [2*(1-tailed Sig.)]	,065 ^a

a. Not corrected for ties.

A-C

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,6675	,69501	2,00	4,33	3,6700	3,6700	4,2475
VAR00003	12	2,0000	1,04447	1,00	3,00	1,0000	2,0000	3,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	1,00	6	7,58	45,50
	3,00	6	5,42	32,50
	Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	11,500
Wilcoxon W	32,500
Z	-1,090
Asymp. Sig. (2-tailed)	,276
Exact Sig. [2*(1-tailed Sig.)]	,310 ^a

a. Not corrected for ties.

B-C

➔ **NPar Tests**

[DataSet0]

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
VAR00002	12	3,6950	,68736	2,00	4,33	3,6700	4,0000	4,0000
VAR00003	12	2,5000	,52223	2,00	3,00	2,0000	2,5000	3,0000

Mann-Whitney Test

Ranks

	VAR00003	N	Mean Rank	Sum of Ranks
VAR00002	2,00	6	7,92	47,50
	3,00	6	5,08	30,50
	Total	12		

Test Statistics^b

	VAR00002
Mann-Whitney U	9,500
Wilcoxon W	30,500
Z	-1,425
Asymp. Sig. (2-tailed)	,154
Exact Sig. [2*(1-tailed Sig.)]	,180 ^a

a. Not corrected for ties.

