KADİR HAS UNIVERSITY
SCHOOL OF GRADUATE STUDIES
PROGRAM OF MANAGEMENT INFORMATION SYSTEMS

OPEN DATA PLATFORM FOR AGRICULTURE: MULTI LAYERS PERSPECTIVE

ŞAHİN AYDIN

PH.D. THESIS

ISTANBUL, AUGUST, 2020

Şahin Aydın

Ph.D. Thesis

2020

OPEN DATA PLATFORM FOR AGRICULTURE: MULTI LAYERS PERSPECTIVE

ŞAHİN AYDIN

PHD THESIS

Submitted to the School of Graduate Studies of Kadir Has University in partial fulfillment of the requirements for the degree of PhD in the Program of Management Information Systems

ISTANBUL, AUGUST, 2020

DECLARATION OF RESEARCH ETHICS /
METHODS OF DISSEMINATION

I, ŞAHİN AYDIN, hereby declare that;

- this PhD Thesis is my own original work and that due references have been appropriately provided on all supporting literature and resources;
- this PhD Thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- I have followed "Kadir Has University Academic Ethics Principles" prepared in accordance with the "The Council of Higher Education's Ethical Conduct Principles"

In addition, I understand that any false claim in respect of this work will result in disciplinary action in accordance with University regulations.

Furthermore, both printed and electronic copies of my work will be kept in Kadir Has Information Center under the following condition as indicated:

☒ The full content of my thesis will not be accessible for 3 years. If no extension is required by the end of this period, the full content of my thesis will be automatically accessible from everywhere by all means.

ŞAHİN AYDIN

____

10/08/2020

KADİR HAS UNIVERSITY

SCHOOL OF GRADUATE STUDIES

ACCEPTANCE AND APPROVAL

This work entitled **OPEN DATA PLATFORM FOR AGRICULTURE: MULTI LAYERS PERSPECTIVE** prepared by **ŞAHİN AYDIN** has been judged to be successful at the defense exam held on **August 10, 2020** and accepted by our jury as **PHD THESIS**.

APPROVED BY:

Assoc. Prof. Dr. Mehmet Nafiz Aydın (Advisor)	Kadir Has University	____________________
Prof. Dr. Hasan Dağ	Kadir Has University	____________________
Assoc. Prof. Dr. Mehmet Aktaş	Yıldız Technical University	____________________
Assoc. Prof. Dr. Bilgin Metin	Boğaziçi University	____________________
Assist. Prof. Dr. Oğuzhan Ceylan	Kadir Has University	____________________

I certify that the above signatures belong to the faculty members named above.

____

Prof. Dr. Emine Füsun Alioğlu

Dean of School of Graduate Studies

DATE OF APPROVAL: (10/08/2020)

TABLE OF CONTENTS

OPEN DATA PLATFORM FOR AGRICULTURE: MULTI LAYERS PERSPECTIVE

ABSTRACT

In recent years, several projects which are supported by information and communications technologies (ICT) have been developed in the agricultural domain to promote more precise agricultural activities. These projects account for different kinds of key ICT terms such as internet of things (IoT), wireless sensors networks (WSN), cloud computing (CC), and semantic web. The implementation of these projects successfully depends on the extent to which various stakeholders provide support by leveraging relevant data, gathered from heterogenous data sources. Agriculture domain has a great number of stakeholders. These stakeholders need sophisticated data and appropriate intelligence to get benefits in order to perform precise agricultural activities.

In this thesis, first, we shall investigate the open data term in an agricultural context, create an open data processing model, and develop an IoT-based solution to gather environmental data from agricultural fields. We show viability of the proposed model by developing an ICT-based solution.

Second, we propose an ontology for hazelnut and examine a variety of ontology evaluation tools and methodologies to assess the ontology developed. In particular, we use a number of the metrics to evaluate the quality of proposed ontology and discuss the implications of proposed Hazelnut Trait Ontology and its quality for both researchers and practitioners.

Third, we propose a generic ontology-based data acquisition model to create data acquisition forms based on model-view-controller (MVC) design pattern, to publish and make use of on the agricultural open data platforms. We develop a tool called OWL2MVC that integrates the Hazelnut Trait Ontology, which illustrates the effectiveness of the proposed model for generating data acquisition forms. OWL2MVC Tool was evaluated in terms of usability considering five different scales such as efficiency, affect, helpfulness, control, and learnability by fifty-three respondents implementing the case-study scenario. Among others the findings show that the tool has satisfactory usability score overall and is promising to provide stakeholders with required support for agricultural open data platforms.

Lastly, we propose an ontology-based data integration approach to demonstrate how to use an agricultural trait dictionary for linking site-specific parameters to sensor measurement values. In addition, we develop an open data platform to justify the viability of the proposed approach. The open data platform has been evaluated in terms of usability considering five different scales such as efficiency, affect, helpfulness, control, and learnability by twenty-seven respondents. According to the usability results, the developed platform has satisfactory scores for each one of them. The score of global usability scale is quite reasonable for the open data platform. Furthermore, this research has shown how to utilize the web services and APIs to carry out the syntactic interoperability of sensor data in agriculture domain.

Keywords: Open data, Open data in agriculture, Open data processing model, Semantic web in agriculture, Agricultural Trait dictionaries, Ontology-based data acquisition model, Ontology-based data acquisition tool, Ontology-based open data platform, ontology-based data integration, Semantic and syntactic interoperability using agricultural trait dictionaries

TARIM İÇİN AÇIK VERİ PLATFORMU: ÇOK KATMANLI BAKIŞAÇISI

ÖZET

Son yıllarda, daha hassas tarımsal faaliyetler gerçekleştirebilmek amacıyla bilgi ve iletişim teknolojileri (BİT) tarafından desteklenen çeşitli projeler geliştirilmekte ve tarım alanında oldukça yoğun bir şekilde kullanılmaktadır. Bu projeler, nesnelerin interneti (IoT), kablosuz sensörler ağları (WSN), bulut bilişim (CC) ve anlamsal ağ gibi farklı temel BİT terimlerini içine alan yenilikçi ve akıllı araçlarla desteklenmektedirler. Bu projelerin başarıyla uygulanması, çeşitli paydaşların farklı türde veri kaynaklarından toplanan ilgili verileri kullanarak ne ölçüde destek sağladıklarına bağlıdır. Tarım alanındaki çok sayıda farklı paydaşın hassas tarımsal faaliyetlerde bulunabilmeleri ve tarımsal üretim sürecinde ürün ile ilgili farklı parametrelerin izlenebilirliğini elde edebilmeleri için karmaşık verilere ve bu verilerle oluşturulacak akıllı sistem uygulamalarına ihtiyaçları vardır.

Bu tezde, ilk olarak, açık veri terimi tarım bağlamında incelenmiş, açık bir veri işleme modeli oluşturulmuş ve tarım alanlarından çevresel verileri toplamak için IoT tabanlı bir çözüm geliştirilmiştir. Ayrıca BİT tabanlı bir çözüm geliştirilerek önerilen modelin uygulanabilirliği de gösterilmiştir.

İkinci olarak, fındık için ürüne özgü özellik ontolojisi geliştirilmiştir. Geliştirilen ontolojiyi değerlendirmek için çeşitli ontoloji değerlendirme araçları ve yöntemleri incelenmiştir. Özellikle, önerilen ontolojinin kalitesini değerlendirmek için bir dizi metrik kullanılmış ve geliştirilen fındık ontolojisinin sonuçları ve kalitesi hem araştırmacılar hem de uygulayıcılar için tartışılmıştır.

Üçüncü bir araştırma konusu olarak, tarımsal açık veri platformları üzerinde yayınlamak ve kullanmak amacıyla model-view-controller (MVC) tasarım desenine dayalı veri toplama formları oluşturmak için genel ontoloji tabanlı bir veri toplama modeli önerilmiştir. Veri toplama formlarının oluşturulması için önerilen modelin etkinliğini

gösteren Fındık Ontolojisini entegre eden OWL2MVC adlı bir araç geliştirilmiş ve bu araçla ilgili detaylar da ilgili bölümde açıklanmıştır. OWL2MVC yazılım aracı, vaka çalışması senaryosunu uygulayan elli üç katılımcı tarafından verimlilik, etkililik, yardımcılık, kontrol edilebilirlik ve öğrenilebilirlik gibi beş farklı ölçek dikkate alınarak kullanılabilirlik açısından değerlendirilmiştir. Elde edilen bulgular, aracın genel olarak tatmin edici kullanılabilirlik puanına sahip olduğunu ve paydaşlara tarımsal açık veri platformları için gerekli desteği vermeyi vaat ettiğini göstermiştir.

Son olarak, sahaya özgü parametreleri sensör ölçüm değerlerine bağlamak için bir tarımsal özellik sözlüğünün nasıl kullanılacağını göstermek amacıyla ontoloji tabanlı bir veri entegrasyonu yaklaşımı önerilmiştir. Ayrıca, önerilen yaklaşımın uygulanabilirliğini kanıtlamak için bir açık veri platformu da geliştirilmiştir. Açık veri platformu, yirmi yedi katılımcı tarafından verimlilik, etkililik, yardımcılık, kontrol edilebilirlik ve öğrenilebilirlik gibi beş farklı ölçek dikkate alınarak kullanılabilirlik açısından değerlendirilmiştir. Kullanılabilirlik sonuçlarına göre, geliştirilen platformun her bir ölçek için tatmin edici puanlara sahip olduğu tespit edilmiştir. Küresel kullanılabilirlik ölçeğinin puanı açık veri platformu için oldukça makul bir değer olarak hesaplanmıştır. Ayrıca, bu araştırmada, tarım alanındaki sensör verilerinin sözdizimsel birlikte çalışabilirliğinin gerçekleştirilebilmesi için web hizmetlerinin ve Web API'lerin nasıl kullanılacağı da geliştirilen açık veri platformu yardımıyla gösterilmiştir.

Anahtar Sözcükler: Açık veri, Tarımda açık veri, Açık veri işleme modeli, Tarımda anlamsal ağ, Tarımsal özellik sözlükleri, Ontoloji tabanlı veri toplama modeli, Ontoloji tabanlı veri toplama aracı, Ontoloji tabanlı açık veri platformu, Ontoloji tabanlı veri entegrasyonu, Tarımsal özellik sözlüklerini kullanarak anlamsal ve sözdizimsel birlikte çalışabilirlik

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my supervisor Assoc. Prof. Dr. Mehmet Nafiz Aydin for the continuous support and his guidance through each stage of my PhD thesis. I would like to express my endless thanks for his patience and effective supervision along my academic journey. To me, he is a great scientist who enlightens people walking on long and tough science way.

The true friend is one of the greatest blessings. Therefore, I would like to thank to my true friend Ertan Çoban very much for his continuous support. Furthermore, I am grateful to him for assisting me technically throughout the development process of software tools and systems for research which was carried out within the scope of my PhD thesis.

I would like to thank my students for their good wishes and prayers throughout my PhD thesis. I would like to send special thanks to my student Selim Ata who helped me for designing graphical user interface of developed software tools within the scope of my research. I am proud of him due to continuously improve himself and being self-motivated and self-disciplined. I would like to thank to him once again for assisting me technically during whole process of developing the software tools.

I would like to express my deepest thanks to my parents and sister. I would never be successful in my PhD researches without their unrequited love, continuous support, and sincere prayers. I am grateful to them. I am so glad I have them.

Şahin Aydın
Istanbul, August 2020

To my sister and parents

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Topic Orientation

In recent years, several projects that are supported by information and communications technologies (ICT) have been developed in the agricultural domain to manage agricultural practices. Each agricultural practice is performed by a lot of stakeholders, including farmers, domain experts, traders, and regulation agencies. These stakeholders need more sophisticated data and much more appropriate intelligence to perform the relevant agricultural practices. The data regarding any agricultural domain are based upon the activities of stakeholders. ICT applications are valuable resources in the agriculture domain to handle data gathered from various data sources such as farmers, sensors, government, researchers, analysts, and the market. ICT applications support gathering, processing, storing, publishing the agricultural data in an efficient way. Hazelnut is one of the valuable agriculture domains that can benefit from implementing ICT applications.

ICT applications might help farmers deal with a number of challenges. For instance, recording and reporting the data with respect to age productivity, morbidity of trees, and the number of trees in orchards. Furthermore, ICT applications provide analyzing observation data for soil fertility and detecting location of orchards (southern slopes or northern slopes). They provide storing, publishing, and reporting the following data gathered from hazelnut farmers as well: the pruning period of hazelnut trees, fertilization techniques used for hazelnut, types of sprayers used in hazelnut orchards, and irrigation methods. Internet of Things (IoT) tools which have particular sensors might improve agricultural practices by gathering data regarding the environmental effects on plant breeding like ICT applications. Sensors which are typically the main components of IoT applications might detect and measure a variety of data with respect to hazelnut. For instance, slope of hazelnut orchards, weather temperature, velocity of wind, rainfall, soil moisture, soil Ph, frost (date of most recent frost, minimum temperature, duration of temperature below 0°C), relative humidity (diurnal and seasonal range), light intensity, and leaf anatomy might be detected and measured using IOT applications. They might also be monitored by implementing ICT applications.

When the data heterogeneity in the agriculture domain is considered, it is essential to provide publishing domain-specific vocabularies while gathering data from heterogeneous sources and merging them. Considering this variety of data sources and users interested in domain-specific information regarding hazelnut, one needs to have an ontology that fulfills the quality requirements. Ontologies enable us to share a common understanding of the structure of information among people or software agents, to reuse domain knowledge, to make domain assumptions explicit, to separate domain knowledge from the operational knowledge, and to analyze domain knowledge (Noy and McGuinness, 2001). Evaluating the quality of ontologies which are used in such critical processes plays a vital role for applying them to the semantic systems in a reliable way. The usage of ontologies with lack of quality requirements might cause complicated issues such as misunderstandings among people and software agents, publishing wrong metadata, and gathering, storing, and processing inappropriate data in the relevant domain. A variety of tools and methodologies are used while evaluating the ontologies to eliminate these problems.

Data acquisition forms mean a kind of application user interface which enables us to gather domain-specific data by stakeholders through web-based platforms. In the context of open data platforms, they are developed with the aim of storing, processing, publishing, freely accessing, reusing, and sharing structured data considering the open standards. It is required that well-organized and standardized data models collect domain-specific structured data through data acquisition forms. The required data models are represented as classes in the object-oriented world. The properties of the classes represented by the data models are the elements of data acquisition forms. Some properties of the classes might only correspond to a value which is defined by the user or domain stakeholder. On the other hand, another property's value might be selected from a set of predefined terms stored in databases or files and listed in an appropriate element on the form. For instance, "Bud Shape" which is a kind of hazelnut trait regarding its characterization might equal to only one of three terms such as "Conical", "Ovoid", and

Globular. In this case, a generated data acquisition form has to provide an appropriate form control which enables users to select one of the aforementioned options. "First Female Bloom Date", which is a phenology descriptor of hazelnut might only get date-time values. In this case, data acquisition form should similarly provide a control element that makes selecting date and time easy for users. On the other hand, "Leaf Width", which is the leaf descriptor of hazelnut might be defined by domain stakeholders manually. Similarly, data acquisition forms should contain a form element which allows us to enter such a value for the "Leaf Width" descriptor. The examples given above have been provided for hazelnut in the context of agriculture. However, each requirement reflected within the examples should be met for other domains as well. Data acquisition forms generated with the aim of collecting structured domain-specific data have to meet the requirements of any domain stakeholders. In other words, it is difficult to collect data in compliance with a certain data model that is of interest to different types of stakeholders within a domain. It is a daunting task, if not impossible, to determine which stakeholder needs what kind of data to be collected. Furthermore, it is quite challenging to identify domain-specific data for users who are outsiders for the domain. Sophisticated data concerning a domain might only be specified by its own experts. Considering that each data acquisition form is required to be based on a data model, and that each domain stakeholder might need to collect various kinds of domain-specific data, there needs to be a system that allows stakeholders to create data acquisition forms dynamically. Collecting domain-specific data through such dynamic and automized web-based data acquisition forms plays an important role in providing rich domain information including agricultural products. On the other hand, it is not adequate to create only data acquisition forms. They should be automatically made available for entire users through web-based platforms without much effort. Furthermore, there are several issues to be considered. It is required that data defined by users using the generated data acquisition forms should be stored in favorable data storage options. In addition, it is important that stakeholders might export stored data in diverse open formats. Another challenge with data acquisition forms is to meet the requirements of domain stakeholders in the context of data collection. It seems not completely probable to know which stakeholder needs to collect what sort of data concerning the relevant domain. In addition, generating data acquisition forms, publishing, and making them available for stakeholders through web-based platforms require many technical skills.

In recent years, Internet-of-Things (IoT) -based applications have been used in various domains such as health, industry, and agriculture. Considerable amount of data in diverse formats is collected from Wireless Sensor Networks (WSNs) integrated into IoT devices. Semantic interoperability of data gathered from IoT devices are generally being carried out using existing sensor ontologies. However, crop-specific trait ontologies, which include site-specific parameters concerning hazelnut as a particular agricultural product, can be used to make links between domain-specific variables and sensor measurement values as well. Crop-specific trait ontologies can be used for linking site-specific parameters to sensor measurement values. This objective can be achieved by proposing a data integration approach for semantic and syntactic interoperability and developing an open data platform based on this model. There are six objectives of the developed open data platform. It enables to collect domain-specific data concerning particular agricultural products through ontology-based data acquisition forms generated by domain stakeholders using agricultural trait dictionaries. It provides gathering and visualizing stream data concerning site-specific parameters of particular agricultural products through WSNs. It aims to produce domain-specific linked open data using mapping rules constructed by any domain stakeholder using agricultural trait dictionaries. It allows to store semantically annotated agricultural data within the diverse databases and files such as relational databases, graph databases, XML files, RDF files. It provides syntactical interoperability using Web Services and APIs which allow stakeholders share data for a particular agricultural product between different kinds of software applications. It helps to publish well-defined, well-structured, and semantically annotated data concerning a particular agricultural product using open standard in appropriate formats such as RDF/XML, RDF/JSON, N-Triples, Notation 3, Turtle, XML, JSON, HTML, CSV, and Excel.

1.2 Research Objectives and Questions

This work is aimed to address the following research questions and specific chapter:

The research questions addressed within the chapter three:

- What kind of framework is needed to better understand the problem scope and focus of the thesis?
- What are the requirements of the proposed system architecturally considering the necessities of the ecosystem stakeholders?
- What kinds of features (Apps, services, tools etc.) will the system provide to meet the requirements of the stakeholders?

The research questions addressed within the chapter four:

- How to provide common understanding of a particular agricultural product by developing a crop-specific trait ontology?
- How to evaluate an ontology using existing ontology evaluation tools and methodologies?

The research questions addressed within the chapter five:

- What is the appropriate generic model to collect domain-specific data regarding a particular agricultural product using the capabilities of semantic web technologies, contemporary software design patterns, and data storage options to meet data requirements of miscellaneous software applications?
- How could web-based data acquisition forms be published automatically through open data platforms to collect more sophisticated and structured data, concerning an agricultural product, using trait dictionaries and supplementary ontologies, which were utilized as conceptual models to design user interfaces?
- How to collect domain-specific data concerning particular agricultural products through ontology-based data acquisition forms generated by domain stakeholders using crop-specific trait ontologies?
- What is the usability of developed tools based on a proposed model from domain stakeholders' point of view?

The research questions addressed within the chapter six:

- What is the appropriate approach to provide semantic and syntactic interoperability by using crop-specific trait ontologies?
- How to use crop-specific trait ontologies for linking site-specific parameters to sensor measurement values?
- How to utilize the web services and APIs to carry out the syntactic interoperability of sensor data in the agriculture domain?
- How to gather and visualize stream data concerning site-specific parameters of particular agricultural products through WSNs?
- How to produce domain-specific linked open data using mapping rules constructed by any domain stakeholder using crop-specific trait ontologies?
- How to store semantically annotated agricultural data within the diverse databases and files such as relational databases, graph databases, XML (extensible markup language) files, and RDF (resource description framework) files?
- How to provide syntactical interoperability using Web Services and APIs (application programming interface), which allow stakeholders to share data for a particular agricultural product between different kinds of software applications?
- How to publish and export well-defined, well-structured, and semantically annotated data concerning a particular agricultural product using open standard in appropriate formats such as RDF/XML, RDF/JSON, N-Triples, Notation 3, Turtle, XML, JSON, HTML, CSV, and Excel?
- What is the usability of Open Data Platform based on the proposed ontology-based data integration approach from domain stakeholders' point of view?

1.3 The Structure of This Work

The overall structure of the thesis takes the form of eight chapters, including this introductory chapter.

The purpose of the chapter two is to describe and discuss the methodology used in this thesis. In addition, the detailed information concerning the methodology used is provided.

Different studies applying the Design Science Research Methodology (DSRM) have been presented with the aim of indicating the importance of the methodology used. Lastly, chapter two has demonstrated how DSRM is applied for produced the artifacts within the scope of this thesis.

Chapter three begins by establishing the theoretical dimensions of the “Open Data” and articulates the design aspect of a generic multi-layered open data processing model in the context of agriculture domain. In addition, a typical wireless sensor network has been established for demonstrating how to collect environmental data in compliance with the proposed model.

The fourth chapter of the thesis discuses an ontology development for hazelnut to support sustainable farming, and an evaluation of the proposed ontology by using different tools and methodologies. Furthermore, the proposed ontology might be used as the main component of semantic annotation layer within the scope of multi-layer agricultural open data processing model. The research which is carried out within the scope of fourth chapter of the thesis makes noteworthy contributions to the current literature in two ways. First, we review the research conducted on existing agricultural ontologies; then we propose an ontology for hazelnut. Second, we examine a variety of ontology evaluation tools and methodologies to validate the ontology developed. Then, we select three tools and one methodology that cover many quality criteria to evaluate the proposed ontology. Lastly, the findings based on the metrics to evaluate the quality of proposed ontology are examined, and their implications for both researchers and practitioners are discussed as well.

The fifth chapter proposes a generic ontology-based data acquisition model, focusing on the three key themes that are Model-View-Controller design pattern, semantic web, and agricultural trait dictionaries. A dynamically web-based data acquisition form generator tool called OWL2MVC which integrates the Hazelnut Trait Ontology has been developed to prove the validity of the proposed model. Furthermore, OWL2MVC Tool has been evaluated in terms of usability by fifty-three respondents implementing a case-study scenario which is explained in detail within the fifth chapter.

The sixth chapter of the thesis seeks to address how to use an agricultural trait dictionary for linking site-specific parameters to sensor measurement values. An ontology-based data integration approach has been proposed to achieve this objective and an open data platform has been developed to justify the viability of the proposed approach. Furthermore, this section of the thesis has shown how to utilize the web services and APIs to carry out the syntactic interoperability of sensor data in agriculture domain.

The seventh chapter introduces the papers which have been produced and published within the scope of the research. We show the contributions of each paper to the research area and application domain.

The final chapter draws upon the entire thesis, tying up the various theoretical and empirical strands in order to conclude whole studies provided within the thesis and includes a discussion of the implication of the findings to future research into this area.

2. RESEARCH APPROACH AND METHODOLOGY

2.1 Introduction

The purpose of this chapter is to describe and discuss the research approach and the methodology used in this thesis. The research approach has been presented within the second part of this chapter. The detailed information concerning the methodology used has been provided within the third part of this chapter. Furthermore, it has been presented what particular method fragments are adopted at different phases of this research with the aim of indicating the importance of the methodology used. The last part of this chapter has demonstrated how we applied the Design Science Research Methodology (DSRM) for the produced artifacts within the scope of this thesis.

2.2 Research Approach

One of the most important concern of this thesis is to investigate open data term and open data related technologies in different aspects for agriculture domain. The initial focus area of the thesis is to design a multi-layered open data processing model for agriculture and also to develop an agricultural open data platform based on this model. Development of an open data platform is a way of publishing and managing domain-specific data in machine-readable and open formats. On the other hand, open data platforms or portals can be used by individual software developers and ICT companies with the aim of developing and delivering innovative software applications and services based on the data (Kostovski et al., 2012). Generally, software applications and systems have been developed based on a particular architectural model, architectural concept, or a contextual approach. Open data platforms or portals, which are a kind of web-based software applications developed in various domains such as smart city, health, education, and agriculture, have their own architectural development models as well. Several authors have proposed their own platforms or architectural models based on open data. Kostovski et al., have developed an open data platform with its own system architecture based on semantic web technologies with the aim of providing manageable, publishable, linkable, and consumable open data in machine-readable format (Kostovski et al., 2012). Sowe and Zettsu, proposed an open data development model which demonstrates how data gathered from open repositories can be managed, used, transformed, and published by varied stakeholders (Sowe and Zettsu, 2015). According to their proposed open data development model, open data management process consists of the following five steps: checking-out, accessing, using and reusing open data; creating new open dataset; appraising and selecting open data; transforming open data; checking-in or publishing open data (Sowe and Zettsu, 2015). Pflügler et al., have proposed a concept model for the architecture of an open platform, which allows developing mobility solutions using modular mobility services for smart cities (Pflügler et al., 2016). Their concept model has been developed based on design science research approach considering the following elements: data sources, modular services layer, integration layer and solutions (Pflügler et al., 2016). Jayaratne et al. have proposed an open data integration platform as a three-tier web application with the aim of integrating patient, clinical, medical, and historical data gathered from varied health information systems (Jayaratne et al., 2019). Furthermore, they proposed a general-purpose data integration model considering heterogeneous health information sources, healthcare ontologies, and different stakeholders (Jayaratne et al., 2019). Bonina and Eaton have developed a conceptual and theoretical model with the purpose of providing a description with respect to governing and cultivating the demand and the supply side of the open data platform ecosystem by its owner (Bonina and Eaton, 2020). Their proposed governance model for open data platform ecosystem focuses on controlling the activities of contributors on the supply side and developers on the demand side (Bonina and Eaton, 2020). Considering the valuable contributions of the above initiatives concerning open data in platform and architectural model levels, a sustainable multi-layered open data processing model has been proposed for agriculture domain in this thesis. There have been several distinctive features which distinguish our proposed model from other models and system architectures. The most important feature of proposed model is to be designed as multi-layers which are interconnected to each other. The other distinctive features of proposed model have been provided as follows. First, it has been designed for agriculture domain considering heterogeneous data sources such as farmers, IoT devices, government statistical data, and market data. Second, it strongly recommends storing raw data gathered from heterogeneous data sources into graph databases, relational databases, cloud databases, RDF files and/or XML files, due to the fact that domain stakeholders might need to handle such data by their own. Third, the proposed model has been essentially built on semantic web technologies and ontologies which are as much important as open data term. Therefore, agricultural ontologies and supplementary ontologies have critical role in terms of data interchange. Fourth, web services and APIs are quite important for providing interoperability in terms of proposed model. The proposed model has recommended wide range of web services and APIs such as XML web services, restful services, web APIs, mobile services, analysis services, reporting services, and SPARQL query services. Fifth, the proposed model enables users to develop innovative software application by exploiting the open data. Thus, varied stakeholders such as farmers, experts, researchers, and analysts can get benefit concerning the relevant agricultural domain by using developed software applications. Consequently, the proposed multi-layered open data processing model has been designed with the aim of providing sustainable and more precise agricultural activities. In chapter three, more detailed information concerning the proposed multi-layered open data processing model has been provided.

When one examines the general purpose of the proposed model's layers, one can see that collecting domain-specific data from varied data sources has been crucial in terms of performing sustainable and precise agricultural activities. Therefore, there need to be software platforms which are developed based on a crop-specific trait ontology with the aim of carrying out the semantic annotation and semantically data integration. In chapter four, a crop-specific trait ontology belonging to hazelnut, which is called "Hazelnut Trait Ontology", has been developed to meet aforementioned requirements.

In view of all that has been mentioned so far, one may suppose that crop-specific trait ontologies can be used to collect domain-specific data from heterogenous data sources. In chapter five, an ontology-based data acquisition model has been developed for agricultural open data platforms to support this assumption. Furthermore, a tool called "OWL2MVC" which integrates the Hazelnut Trait Ontology for generating data acquisition forms, has been developed based on this model with the aim of illustrating the effectiveness of the proposed model. In chapter five, the proposed model has been compared with similar ones by conducting a systematic literature review and also detailed information concerning the developed tool has been provided.

Throughout this thesis, semantic web technologies and ontologies have been main pillars of each proposed models and developed software tools. In particular, crop-specific trait ontologies have played a vital role in bringing together all parts of eventual platform. In chapter six, open data platform which consists of several components and tools based on semantic web technologies and the crop-specific trait ontologies, has been described. On the other hand, in this chapter, it has been examined how to use crop-specific trait ontologies for linking site-specific parameters to sensor measurement values. A data-integration approach for semantic and syntactic interoperability has been proposed to achieve this objective. Furthermore, in this chapter, it has been shown how to use web services and APIs to carry out the syntactic interoperability of sensor data in agriculture domain. Similarly, a systematic literature review has been conducted in this chapter as well.

2.3 Design Science Research Methodology

Design Science Research (DSR) is a well-established research paradigm and scientific method applied within the information systems (IS) discipline with the aim of creating innovative artifacts which produce solutions for practical problems of general interest (Johannesson and Perjons, 2014) (Costa et al., 2020).

DSR is defined by Hevner and Chatterjee as follows (Hevner and Chatterjee, 2010):
"*Design science research is a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence. The designed artifacts are both useful and fundamental in understanding that problem*".

They determined the first principle of DSR as follows (Hevner and Chatterjee, 2010):
"*The fundamental principle of design science research is that knowledge and understanding of a design problem and its solution are acquired in the building and application of an artifact*".

Generally, artifacts are described as follows (Hevner et al., 2004):

- Constructs as vocabulary and symbols
- Models as abstractions and representations
- Methods as algorithms and practices
- Instantiations as implemented and prototype systems

Hevner et al. proposes seven guidelines with the aim of helping to improve practitioners' understanding with respect to the needs for applying design-science effectively (Hevner et al., 2004). The Information Systems Research Framework is shown in Figure 2.1.

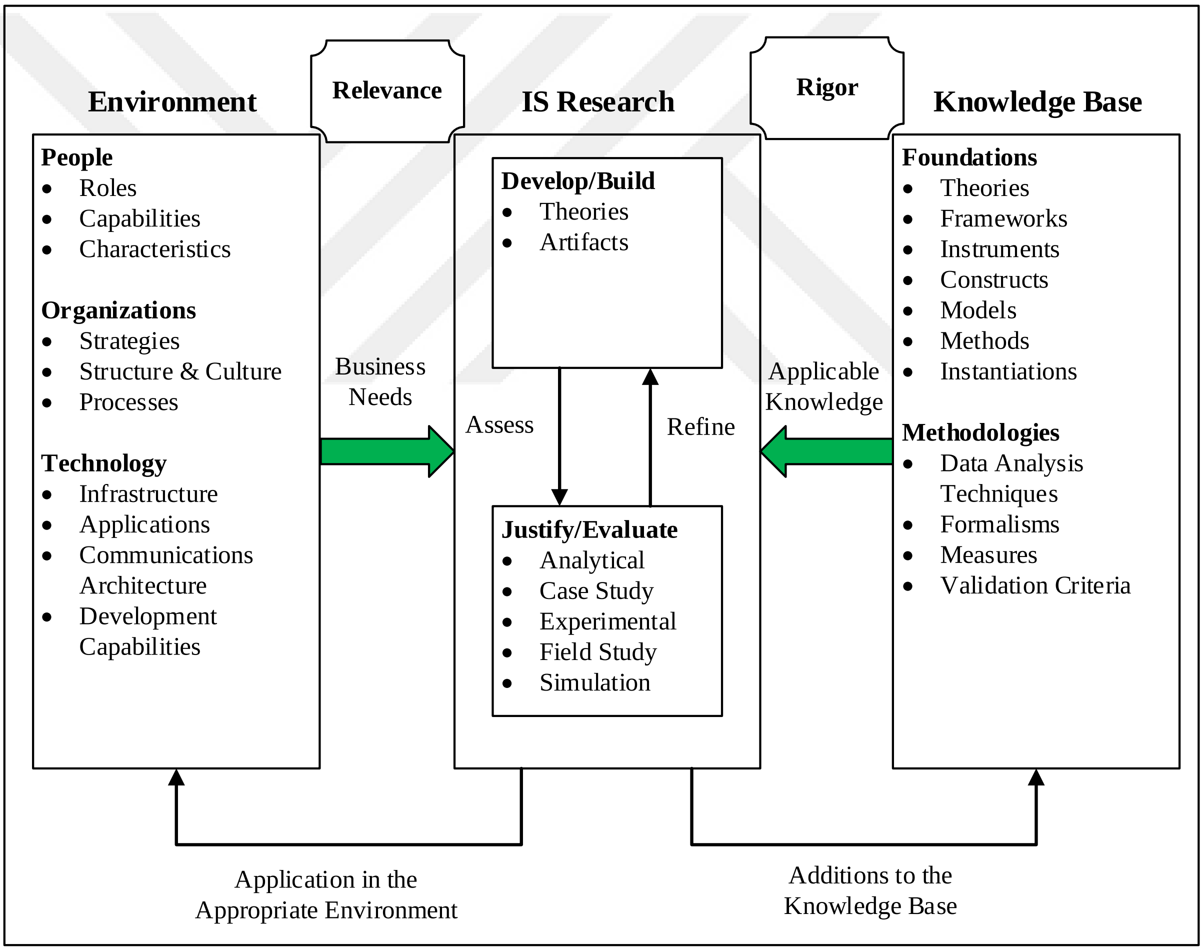


Figure 2.1 Information Systems Research Framework (Adapted from (Hevner et al., 2004))

Their proposed DSR guidelines are as follows (Hevner et al., 2004):

- Guideline 1 - Design as an Artifact: producing a viable artifact

- Guideline 2 - Problem Relevance: developing technology-based solutions
- Guideline 3 - Design Evaluation: evaluating the design of artifact in terms of utility, quality, and efficacy
- Guideline 4 - Research Contributions: making apparent and valid contributions in the areas of the design artifact, design foundations, and/or design methodologies
- Guideline 5 - Research Rigor: applying rigorous methods in constructing and evaluating the design artifact
- Guideline 6 - Design as a Search Process: employing the existing means with the aim of attaining an effective artifact
- Guideline 7 - Communication of Research: making effective presentation of DSR to technology-oriented and management-oriented audiences

Peffers et al., developed a methodology for design science (DS) by introducing a DS process. Figure 2.2 illustrates each step of DSRM process model.
Their process model consists of the following six activities in a nominal sequence (Peffers et al., 2007):

- Problem identification and motivation: definition of the research problem and justification of a solution's value.
- Objectives of a solution: inferring the purposes of a solution from the definition of the problem
- Design and development: creation of the artifactual solution
- Demonstration: demonstration of the artifact's efficacy for solving the problem
- Evaluation: Observing and measuring how well the artifact supports a solution to the problem
- Communication: Communicate the problem and its importance, the artifact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and other relevant audiences

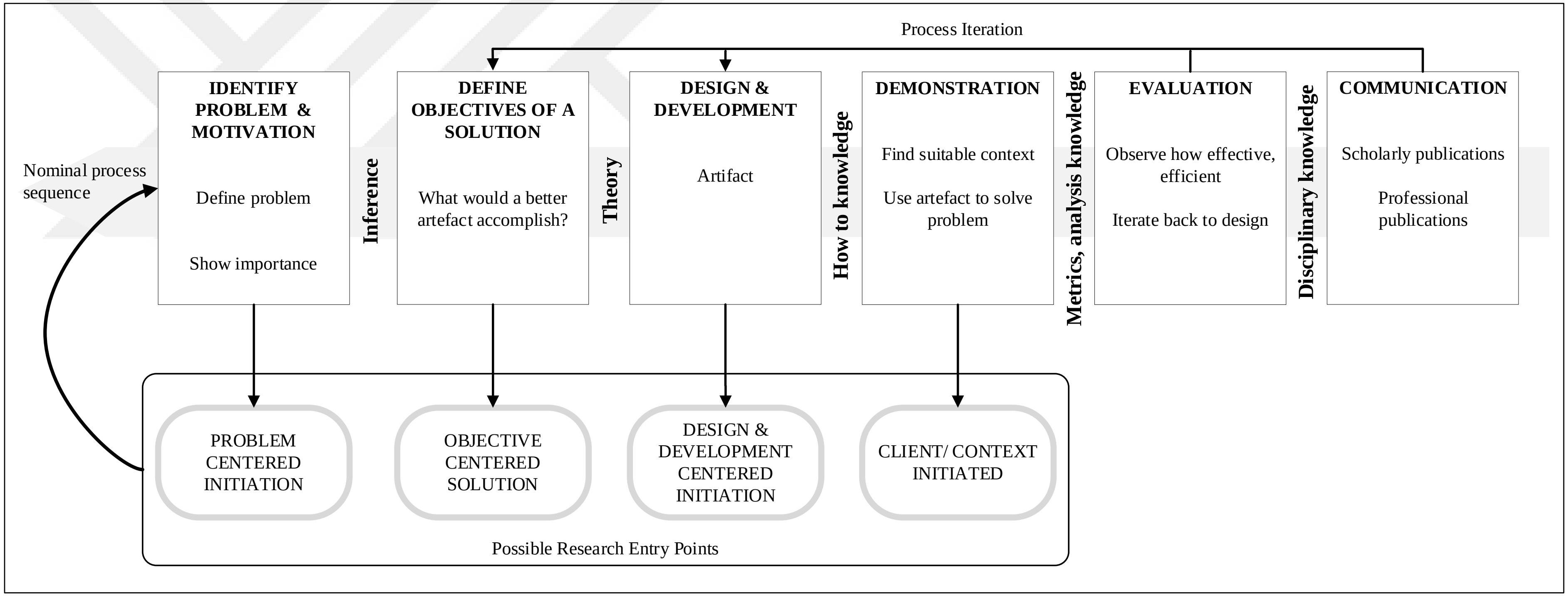


Figure 2.2 Design Science Research Methodology (DSRM) Process Model (Adapted from Peffers et al., 2007)

Today, Design Science Research Methodology (DSRM) is widely used in IS discipline. It has been adopted in development processes of many applications, services, approach, solutions and frameworks such as accounting information systems (Geerts, 2011), web-based knowledge mapping system (Zhang et al., 2011), performance-oriented approach (Wang et al., 2011), mobile phone-based agriculture market information service (Islam and Grönlund, 2012), ICT-enabled service (Hsu and Tsaih, 2014), HR planning framework (Pournader et al., 2014), solution for extracting user requirements (De Silva et al., 2014), business intelligence (BI) in the cloud (Mwili et al., 2016), hospital-based business intelligent system (Kao et al., 2016), knowledge transfer analysis framework (Inan and Beydoun, 2017), real-time information system concerning emergency department crowding (Martin et al., 2018), UI prototypes for evaluations in practice-oriented research (Walter, 2018), agroecosystem for agriculture industry (Nimmagadda et al., 2019), contextualized digital health innovation ecosystem framework (Iyawa et al., 2019), forest management and virtual reality (Holopainen et al., 2020).

The DSRP model has been applied to this work due to it is an explicit guideline for helping researchers while carrying out the design science research in IS (Peffers et al., 2006).

2.4 Applying Design Science Research Methodology to This Work

The six-steps process DSRM, which consists of problem identification and motivation, objectives for a solution, design and development, evaluation, and communication (Peffers et al., 2006) has been used for the production of artifacts within the thesis. There are four different artifacts produced within this thesis:

1. Designing and developing a sustainable multi-layered open data processing model for agriculture
2. Developing an ontology regarding hazelnut called Hazelnut Ontology
3. Developing a tool called OWL2MVC, which creates data acquisition forms based on model-view-controller (MVC) design pattern with the aim of publishing on the agricultural open data platforms using the Hazelnut Ontology while creating. The tool is developed based on a generic ontology-based data acquisition model.
4. Developing an open data platform to justify the viability of the proposed data integration approach for semantic and syntactic interoperability for agricultural open data platforms in the context of IoT using crop-specific trait ontology

Figure 2.3, Figure 2.4, Figure 2.5, and Figure 2.6 illustrate how to apply DSRM process for Artifact 1, Artifact 2, Artifact 3, and Artifact 4, respectively.

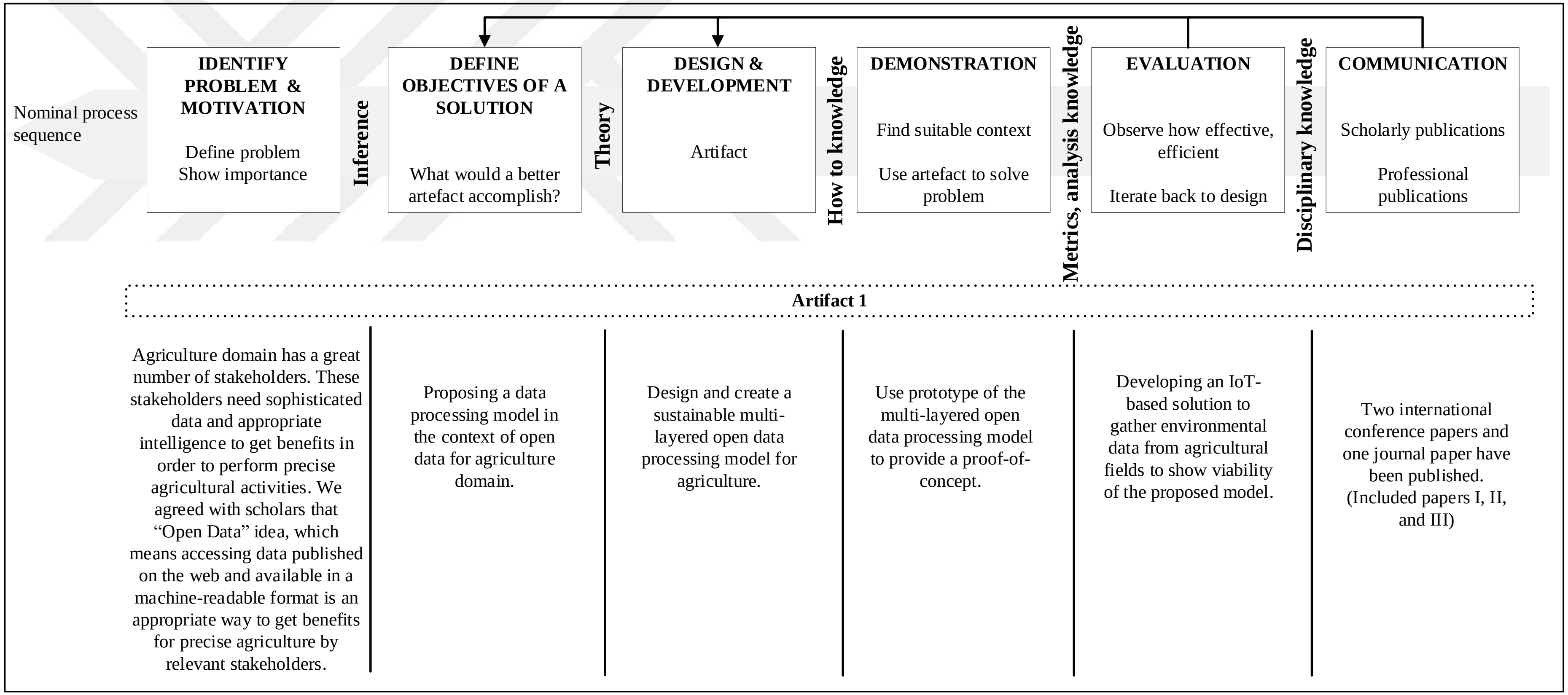


Figure 2.3 DSRM Process for the Artifact 1

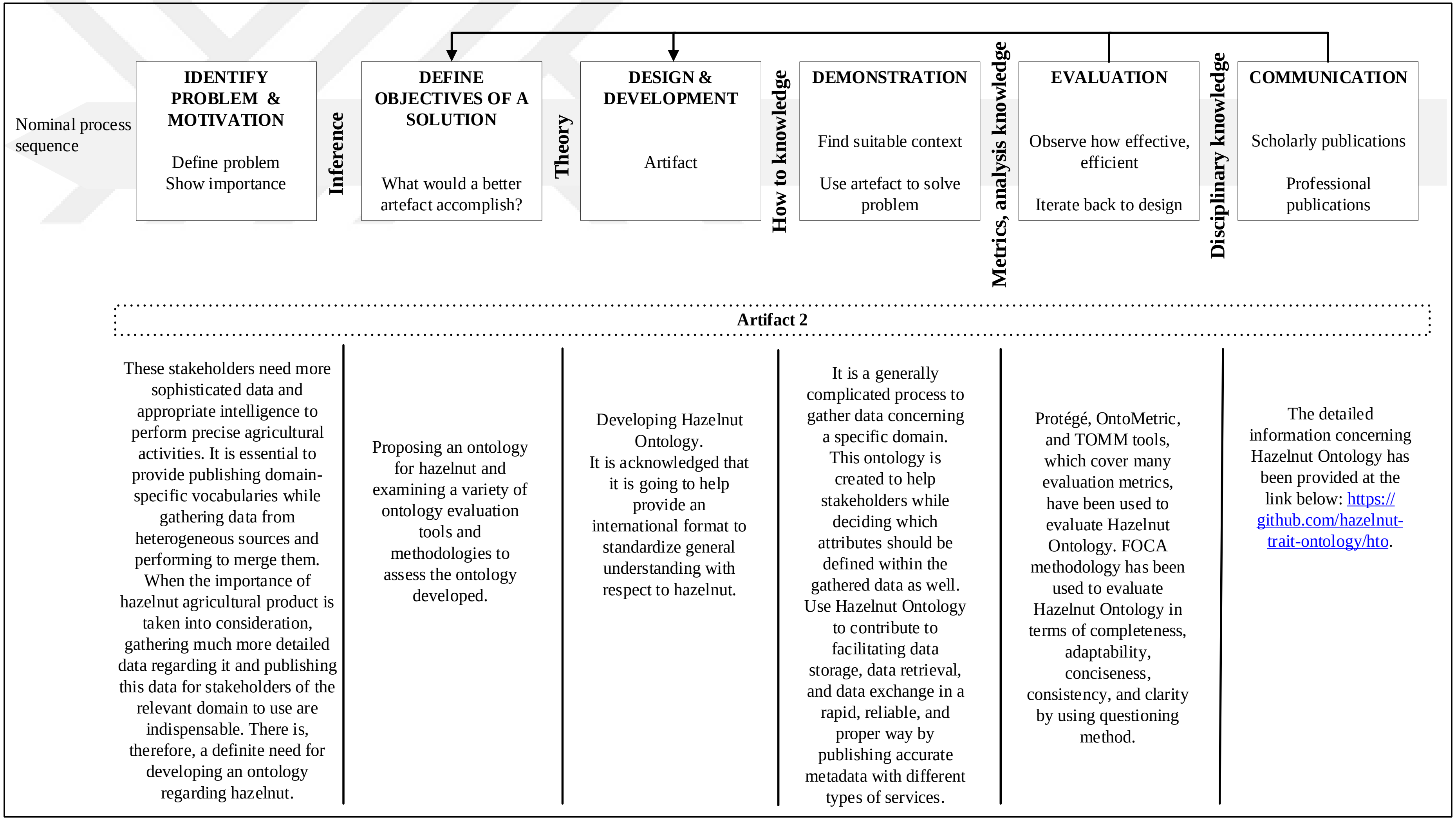


Figure 2.4 DSRM Process for the Artifact 2

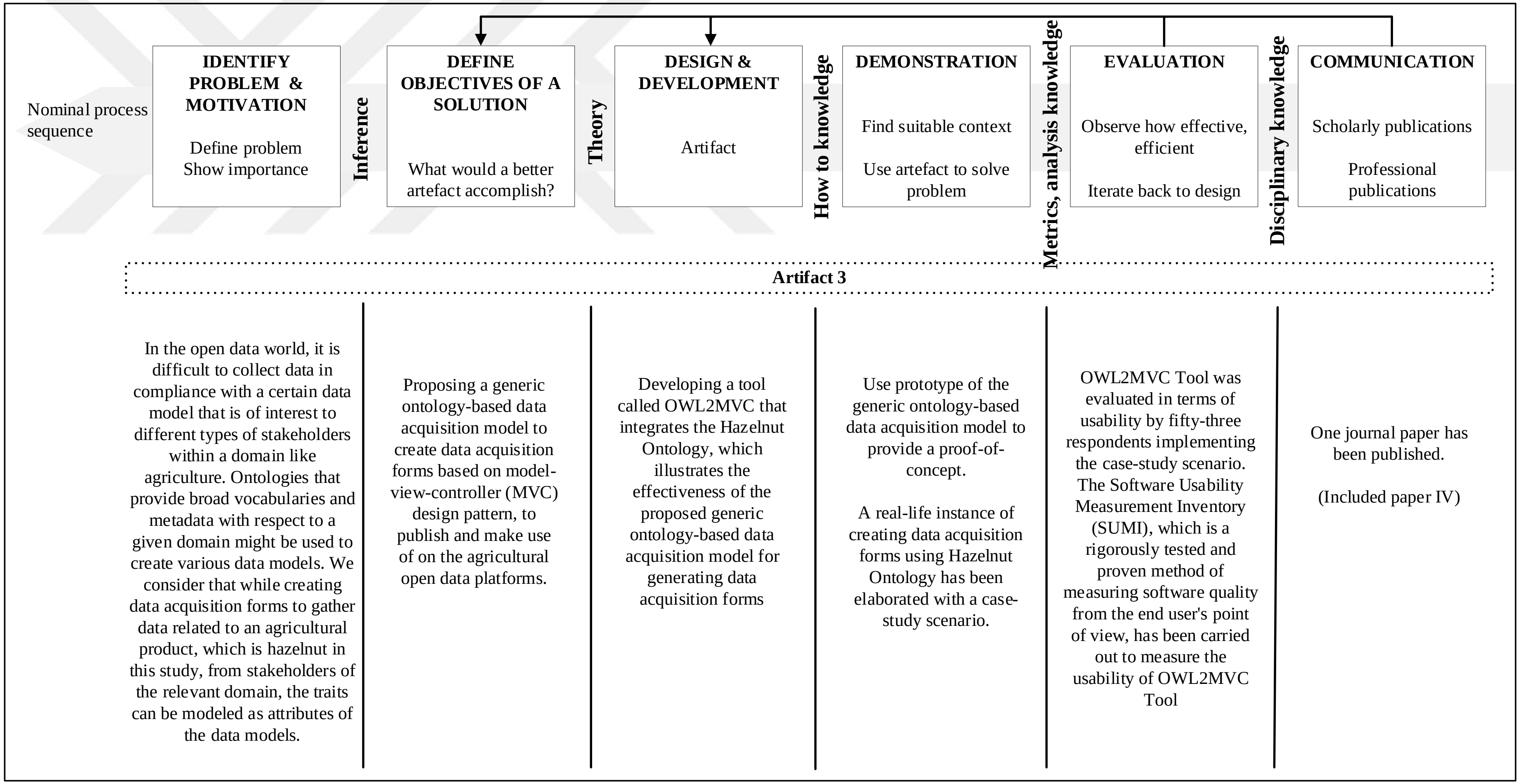


Figure 2.5 DSRM Process for the Artifact 3

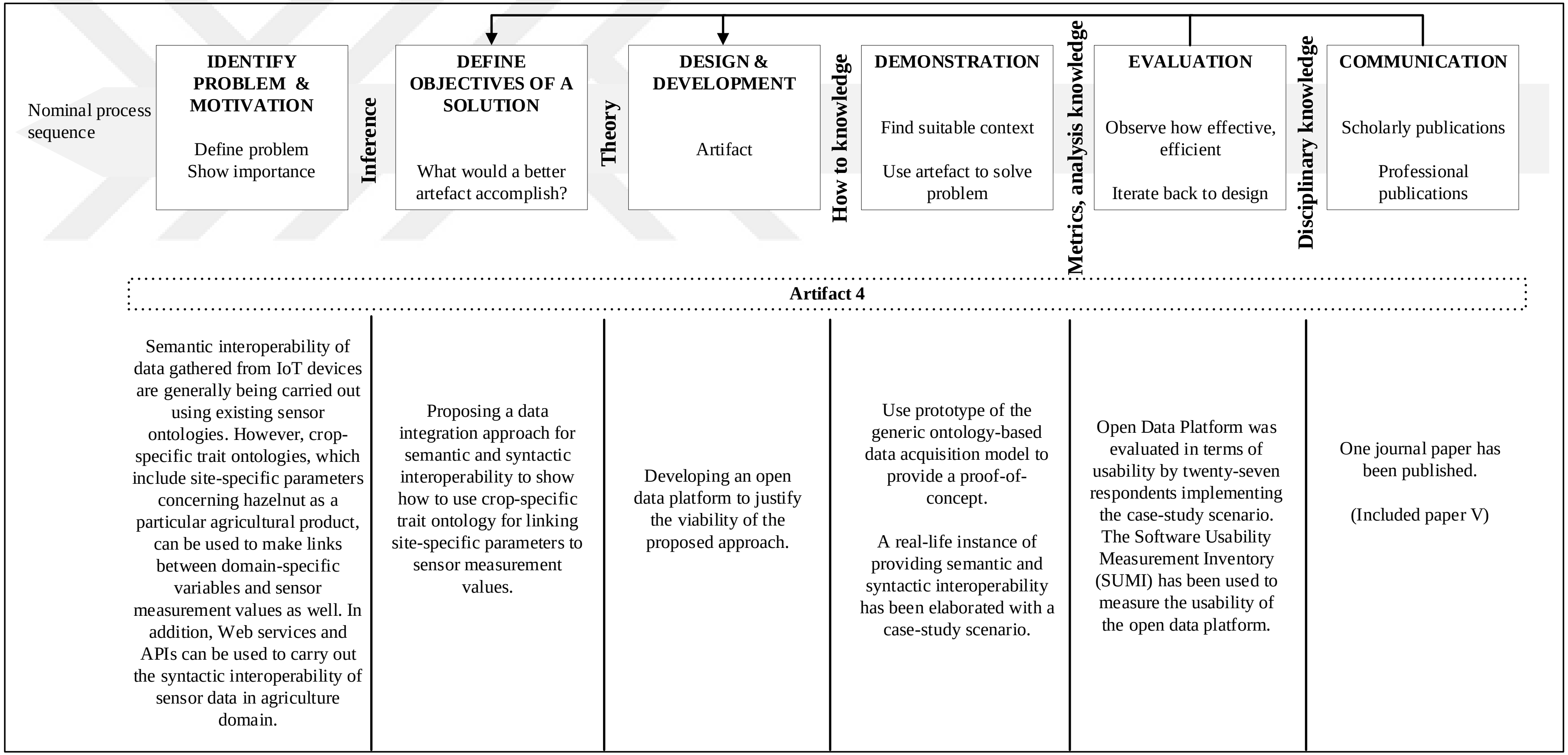


Figure 2.6 DSRM Process for the Artifact 4

Applying DSRM to some of the works done within the scope of this thesis has been carried out with some additional iterations. For some artifacts, there needed to turn back from some steps to previous ones.

The first open data processing model consisted of only three different layers called “types of agricultural data sources”, “cloud hub”, and “users”, respectively. However, this model needed to extend by including new particular layers considering the feedbacks received from scholars. In addition, it was necessary to design an open data processing model which is applicable not only hazelnut but also for all agricultural products. While applying DSRM to this work, some iterations have been repeated to meet the above requirements. After the last step was completed and feedbacks regarding the open data processing model were received, it was returned to second step of DSRM. Thus, objectives were redefined, and the suitable model was redesigned as multi-layer including additional particular layers. As shown in Figure 2.3, each step of DSRM were carried out for Artifact 1 which is multi-layered open data processing model.

In the same way, it has been needed returned from “evaluation” step to “design and development” step while applying DSRM for the Artifact 2 – Hazelnut Trait Ontology. On one hand, some classes of the Hazelnut Trait Ontology were redundant, and they should have been removed from the ontology. On the other hand, there needed to create several additional classes to the ontology with the aim of meeting new software application requirements. Furthermore, it was considered that storing the Hazelnut Trait Ontology in GitHub enables the ontologists make new contributions. Therefore, the Hazelnut Trait Ontology was made available in GitHub. The steps of DSRM applied for Artifact 2 are set out in Figure 2.4.

Applying DSRM for the Artifact 3 and Artifact 4 has been carried out with no additional iteration. The steps of DSRM applied for Artifact 3 and Artifact 4 are shown in Figure 2.5 and Figure 2.6, respectively.

3. OPEN DATA AND OPEN DATA PROCESSING MODEL

3.1 Open Data

The term Open Data is frequently used in various domains such as government, health, science, and agriculture by many stakeholders. Each stakeholder of these domains might access open data published in different types of format. The generally accepted definition of the term "Open Data" is data published on the web in machine-readable format and freely accessible, sharable, and reusable for any users or stakeholders (Carolan et al., 2015). Open Data is easily accessible, available in an appropriate format at any time, usable, and sharable, so unlike the other types of data, it is much more beneficial for any stakeholders of any domains in terms of ensuring effective contribution to the relevant domain. Creating links between interrelated data enables stakeholders to access much more sophisticated data in relevant domains. Accessing such valuable data has a positive effect on making valuable contributions to the relevant domain by stakeholders. Uniform Resource Identifiers (URIs), which are used as the name of the things represented in Linked Data (in other words, Web of Data) (Bizer et al., 2009), render open data accessible and controllable using the strength of semantic web technologies.

3.2 Focus Domain: Hazelnut Production

Agriculture increases employment that covers more than third of the world population via exportation and domestic income which makes it one of the leading sectors in the world (Perry, 2017). Turkey which is the leader country in terms of hazelnut production in the world, has an important role for contributing more information with respect to hazelnut production cycle from this aspect. As well as Turkey is the leader of hazelnut production, it is the leading producer of dried apricots, raisins, and dried figs (Agriculture and Food, 2018). Since having such a convenient weather quality which only a few countries could have for hazelnut production, Turkey, has 75% of overall production and 70-75% of the exportation. Furthermore, considering that there have been 4 million people correspond with almost 5% of the overall population of Turkey, who are concerned with hazelnut produced over an area of 550-600 thousand hectares, this makes hazelnut quite important in terms of socio-economic aspect (Turkey's Hazelnut, 2018).

3.3 Agricultural Data as Open Data

In Turkey, statistics concerning agricultural products, including hazelnut are presented as only hypertext markup language (HTML) pages, spreadsheets, and comma-separated files (CSV) by Turkish Statistical Institute (TUIK). The context of hazelnut data presented by TUIK contains just a certain type of statistical information. However, the domain stakeholders such as farmers, experts, researchers, and analysts might generate more detailed and sophisticated data regarding hazelnut so that gaining knowledge from it must be straightforward and obvious. In order to achieve that, the data must include basic parameters used for the general management of the accession and environmental and field-specific parameters. This kind of data, freely available to use and republish and in machine-readable format, might give a chance to stakeholders for removing the obstacles of implementation of sustainable hazelnut production. Examples of issues regarding hazelnut, which might be commonly encountered by domain stakeholders, might be indicated as follows: limited information or uncertainty about supply and demand side. In addition, there are several problems regarding hazelnut production such as aging hazelnut gardens, land element and position, soil texture, blights, fertilizing, treatment, less pollination, less pruning, agricultural pests, price fluctuation, increase of hazelnut production areas uncoordinatedly, and lack of persistent exportation policies (Fındık Veriminde ve Üretiminde Temel Sorunlar, 2018). Furthermore, there exist correlations between environmental factors and hazelnut production. For instance, a sudden drop in the temperature negatively affects hazelnut production. It is not generally possible to access this type of data for such a particular agricultural product. However, it might be a solution to leverage the reference body of knowledge and analyze the reports of scientific experiments by accessing this data. The data obtained from scientific experiments are not usually allowed for using by any stakeholders out of the relevant communities (Murray-Rust, 2008). However, considering the presence of barriers for accessing such kind of data, most of organizations belonging to government or private sectors are in charge of removing them and making the data freely accessible. Nevertheless, aforementioned permission barriers should be removed for hazelnut stakeholders to meet requirement accessing, reusing, and publishing sophisticated data regarding hazelnut as well.

In the part 3.4 of this section, we shall look in the details of processing open data gathered from different data sources and a designed model, considering very exclusive layers from stakeholders' perspectives.

This section of the thesis begins with an introduction section in which the focus domain is explained. Second section gives a brief overview of the agricultural data and open data. In the further section, open data processing model is described in detail. Lastly, a case study, which is developed as an exclusive system to demonstrate the model viability, was explained in the part of this section.

3.4 Open Data Processing Model

As mentioned in the previous sections, a multi-layered open data processing model has been created, and focused on the hazelnut agricultural product. Even though this model focuses on a sustainable agricultural production lifecycle of hazelnut in this study, it is convenient for all agricultural products as well. This model is created by considering data sources, data processing, semantic annotations, data storage, services, applications, and users. The proposed open data model which demonstrates how to process data obtained from heterogeneous agricultural data sources and designed as multi-layered includes varied layers such as types of data sources (IoT devices, statistical data from government, market data, farmers, and other data sources); storing and processing raw data; semantic annotation layer (agricultural product ontology, other ontologies and data interchange engine); data storage layer (graph databases, relational databases, cloud databases, RDF and XML file-based storage options and database service); services layer (XML web services, REST web services, web APIs, mobile services, analysis services, reporting services and SPARQL query services); applications layer (unified web based data platform based on open formats, mobile applications and desktop applications); and lastly, end users layer. The overall model and its each layer, proposed in this part, are being demonstrated in Figure 3.1.

The data are generally gathered from heterogeneous sources such as farmers, sensors, government statistical data, and market data in the agriculture domain. There is no doubt that the most important one of these heterogeneous data sources is "farmers". As a significant data source in terms of the model, farmers, play a vital role for performing agricultural production life cycle.

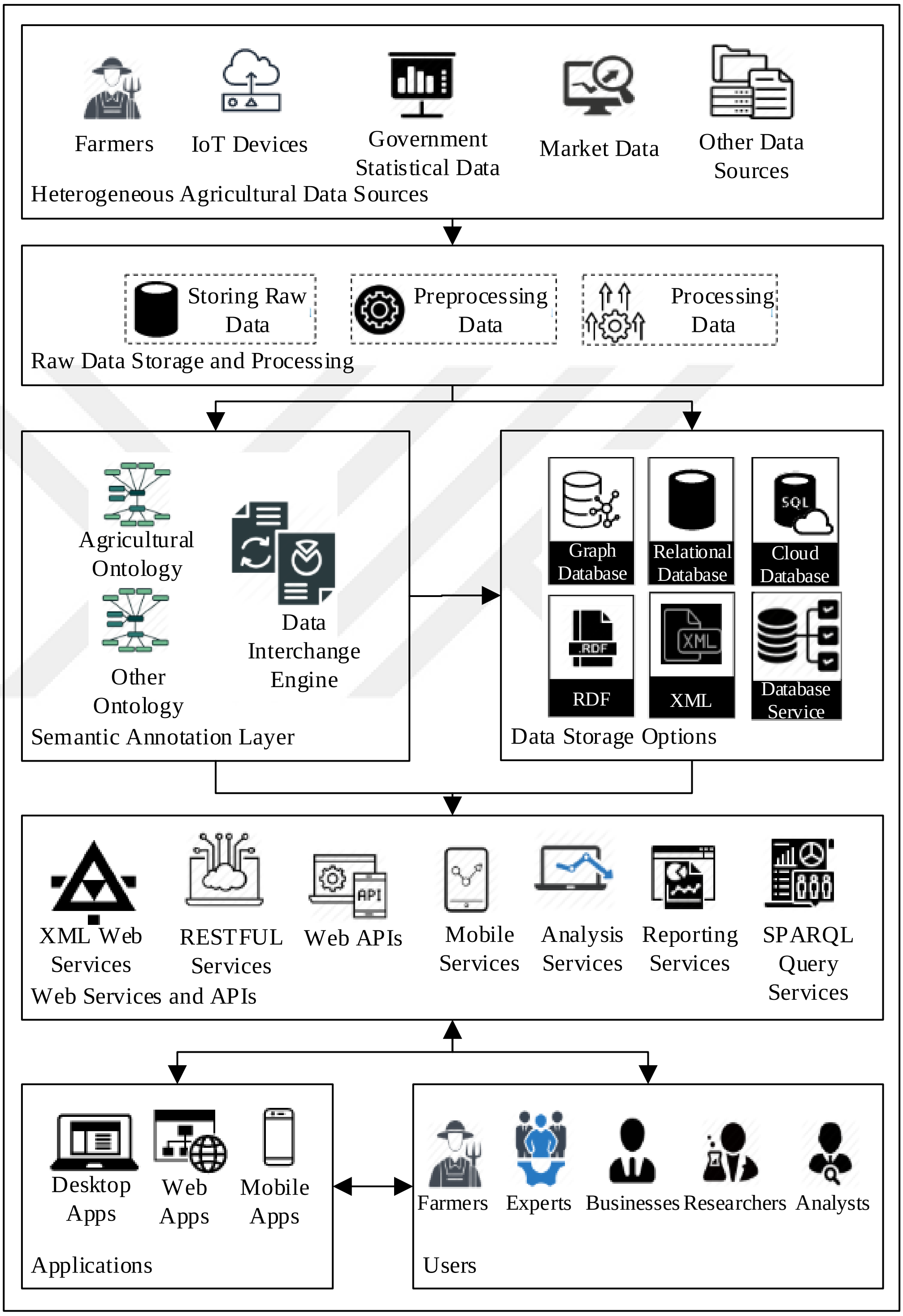


Figure 3.1 Overall Architecture of Open Data Model

Considering the hazelnut production cycle, there is much varied information regarding hazelnut might be collected by consulting of farmers: age productivity and morbidity of trees, observation for soil fertility, the number of trees in orchards, location of orchards (southern slopes or northern slopes), pruning hazelnut trees, fertilization, agricultural spraying, cropping system, propagation method, irrigation method, overall vegetation surrounding the collecting site, stoniness, rockiness etc.

Today, IoT devices have been frequently used for varied domains such as smart cities, industries, education, health, and agriculture. These devices generally consist of a wide range of sensors which are used to collect specific data pertaining to environmental characteristics of site. Collecting and processing data with respect to these characteristics facilitates performing precise agricultural activities. Sensors might provide sensitive measurement values concerning slope, weather temperature, velocity of wind, rainfall, soil moisture, soil Ph, frost (date of most recent frost, minimum temperature, duration of temperature below 0°C), relative humidity (diurnal and seasonal range), light intensity, leaf anatomy, etc.

Agricultural production stats, which are a sort of official statistics data, are recorded by government agencies. The open data model takes into consideration the importance of statistical data concerning agricultural production so, it adopts them as an element of overall architecture.

The statistical records regarding many varied domains in Turkey are compiled, evaluated, and published by TUIK (TUIK Duties and Authorities, 2018). The data with respect to agriculture domain are generally published by TUIK as spreadsheet, comma-separated values, and HTML files, which might be accessed through TUIK web platform.

Marketing operations in agriculture domain include several procedures such as production plan, burgeoning, reaping, classifying, packaging, shipping, managing storehouse, transforming agricultural products into food, delivery, promote, and sale, required for movement of a product from farmers to buyers (Agricultural Marketing, 2018). It is a very crucial cycle for any agricultural product, so the proposed model contains it as a data source element. The experts, researchers, and analysts who are interested in relevant agricultural products might contribute to the system by loading some exclusive data. So, these kinds of data sources are defined as other data sources in the open data model.

According to NIST (National Institute of Standards and Technology), “cloud computing (CC) is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources such as networks, servers, storage, applications, and services, that can be rapidly provisioned and released with minimal management effort or service provider interaction” (Mell and Tim, 2018). Considering the strength and robust characteristics of CC, it has been recommended that data storages, services and applications of the proposed model should serve on the cloud. Raw Data Storage and Processing Layer (RDSandPL) comprises component of raw data storing (SRD), component of data preprocessing (PPD), and component of data processing (PD).

Data obtained from heterogeneous agricultural data sources are stored into varied data storage options such as relational databases, graph databases, cloud databases, RDF files, and XML files via SRD. It is worth bearing in mind that any stakeholder might need to handle data obtained from heterogeneous sources in a raw form. That's why, it is significant storing data in a raw format using recommended data storage options according to an open data model point of view.

Considering the data gathered from heterogeneous agricultural data sources might be incomplete, incorrect, or irrelevant, it is important to emphasize the requirement of a mechanism to detect and correct them. According to open data model, this mechanism is being corresponded with a PPD component. Preprocessing the data ensures removing incompleteness and inconsistency with respect to the raw data and transforming such data into meaningful format to analyze effectively (Siddiqa et al., 2016). As well as, PPD decomposes raw data, it aims to apply exclusive techniques concerning preprocessing to such data. Applying preprocessing techniques such as data integration, data enhancements, data enrichment, data transformation, data reduction, data discretization and data cleansing boosts quality of the data (Taleb et al., 2015). Some of the traditional data preprocessing methods such as K-neighbors Classifier, Logistic Regression, Gaussian Naïve Bayes, Decision Tree, and Support Vector Machine might be used for sensor data, but it should be noted that the major purpose of development of these methods are general, not to specific to sensor data (Zhong et al., 2019). This component is also responsible for semantic annotation of streaming data obtained from sensors, using existing or new ontologies. By this means, semantic annotation of streaming sensor data facilitates data integration processes between varied data sources and open data platforms. In addition, this component solves noisy and redundant problems of sensor data streams, solves missing value problems, detects anomalies related to data, and manages data heterogeneity using the strength of semantic web technologies (Jung, 2011).

The aim of carrying out the preprocessing techniques and methods to the sensor stream data is to build a robust technical infrastructure for data processing component of the proposed model. Data processing means that acquiring meaningful information from preprocessed raw data. Acquiring meaningful information enables users to make estimations for the future via implementing varied analyze techniques and allows creating analytical solutions. Thus, it is important to bear in mind that having such a component is essential in terms of open data model. The design principles proposed by Begoli and Horey, which allows using varied analytical techniques and substantially interested in maximizing the controllable factors, provide easily exploring and analyzing the data by researchers (Begoli and Horey, 2012). The practical analytical solutions are ensured corresponding with these design principles for stakeholders by component of data processing of open data model.

When implementation of relevant methods and techniques is completed by the components of RDSandPL, it is required transforming data into RDF datasets. The data interchange component of Semantic Annotation Layer (SAL) is in charge of transforming data into RDF datasets. The proposed model recommends using agricultural product ontologies to transform data into RDF datasets.

The focus of this section of the thesis is on the development of open data processing model. Therefore, Hazelnut Descriptor Ontology in other words, Hazelnut Trait Dictionary shall be introduced within the scope of the further section of the thesis.

Data gathered from different data sources and processed by RDSandPL might be stored in graph databases, relational databases, cloud databases, XML files, and RDF files, which are recommended as data storage options in the open data model. The proposed open data model is noted the importance of converting raw data into RDF datasets. Therefore, transforming data gathered from varied sources into triple store which means subject, predicate, and object, has been built as a component of semantic annotation layer. Due to RDF datasets have robust features in terms of storage and indexing (Luo et al., 2012), they have been considered as a storage option in the proposed open data model. Relational databases component of data storage layer (DSL) is in charge of storing relational data. Graph databases component of DSL is in charge of storing data as graphs. Cloud databases might store data as both graphs and relational. Database services, which is a component of DSL, provide a secure connection, session management, user mapping, user authentication, and authorization between components of services layer (SL).

Another essential layer of proposed model is the services layer, which consists of seven different types of services: XML web services, REST web services, web APIs, mobile services, analysis services, reporting services, and SPARQL query services. According to W3Schools, “web services are defined as self-contained and self-describing application components that can be discovered using UDDI, be used by other applications, and they communicate using open protocols“ (XML Web Services, 2018). Web Services and APIs Layer (WSandAL), which is in charge of providing interoperability between mobile, desktop, and embedded software applications and databases, is a significant part of the proposed model. While providing interoperability by WSandAL, the RESTful Services component of this layer plays an important role. Software agents, tools and applications, which are being developed using different or similar development technologies, utilize agricultural data by consuming RESTful Services. Another important feature of the proposed model is mobility. To meet this requirement, it strongly recommends developing mobile applications. It is, therefore, needed to generate mobile services to develop an efficient infrastructure for mobile devices. Mobile services, which provide connection to different kinds of mobile devices must take part in the proposed model. Stakeholders might be limited by lack of knowledge to apply data analysis methods to agricultural data. The proposed model recommends building analysis services to solve this problem.

Designing and publishing detailed reports are required to develop usable software systems which aims to process data. Reporting Services component of open data model aims to meet these requirements. It enables domain stakeholders create dynamic, well-designed and detailed reports. As mentioned in previous part of this section, SPARQL is the language to query RDF datasets which are stored in files or graph databases. In the proposed open data model, creating queries using this particular language and compiling them are provided and facilitated by service component named SPARQL Query Service.

Applications Layer (AL) allows end users to access agricultural data obtained from heterogeneous data sources. AL consists of web, mobile and desktop applications which enable us to import, export, report, analyze, and query data.

There are a number of varied end users belonging to proposed open data model, called domain stakeholders such as farmers, researchers, businesses, analysts, and domain experts. Domain stakeholders are provided to utilize data which are gathered from heterogeneous agricultural data sources and might be consumed through the WSandAL and processed by applications layer. Considering the importance of accessing, reusing, and processing such data, it allows domain stakeholders of hazelnut agricultural product construct common knowledge with respect to performing agricultural activities in an efficient way. Every year, several exploratory investigations are performed to estimate total amount of hazelnut production by different institutions of Turkey. However, these current exploratory investigations to find out the hazelnut production of a further year are limited because of unforeseen circumstances with respect to hazelnut harvest. Unfortunately, some discrepancies are seen among the estimations which are carried out by different government agencies every year. Due to the hazelnut price per kilogram is being affected from these discrepancies negatively, rapid price fluctuations are being seen for hazelnut in Turkey. When one takes into consideration the problems for hazelnut market, each user in other words domain stakeholder, play a vital role in terms of the proposed open data model.

The proposed open data model reveals that how each of stakeholders might exploit the processed and published agricultural data for removing the obstacles and problems concerning any particular agricultural product. The proposed open data model by its data processing perspective might enable farmers who have a key role in agriculture domain plan their production cycle effectively. The domain experts and researchers might take preventive measures to prevent further spread of the agricultural diseases by producing the meaningful information using the applications and services developed based on the proposed open data model. Varied kinds of statistical analysis with respect to market, investment, and employment might be undertaken by analysts utilizing the conferred data by proposed open data model. Statistical analysis presented by analysts might be used to support making effective decisions concerning market share of agricultural businesses for the future.

3.5 Implementation

The proposed multi-layered open data processing model is introduced within the previous sections. In this part of the section, a prototype of the electronic system is developed by establishing a wireless sensor network (WSN), which consists of varied sensors and other electronic devices in order to test the viability of the proposed model. This implementation is only focused on streaming data from IoT devices to the databases through web services. Other aspects of the proposed model shall be elaborated in different studies in future studies. There are four sensor nodes named router 1, router 2, router 3 and router 4 in the developed system, respectively. Figure D.2, Figure D.3, Figure D.4, and Figure D.5 represent the router 1, router 2, router 3 and router 4, respectively. These nodes transmit measured data to the coordinator representing in Figure D.1. This system structure is an implementation of ordinary WSN. Figure 3.2 presents the architecture of the developed wireless sensor network.

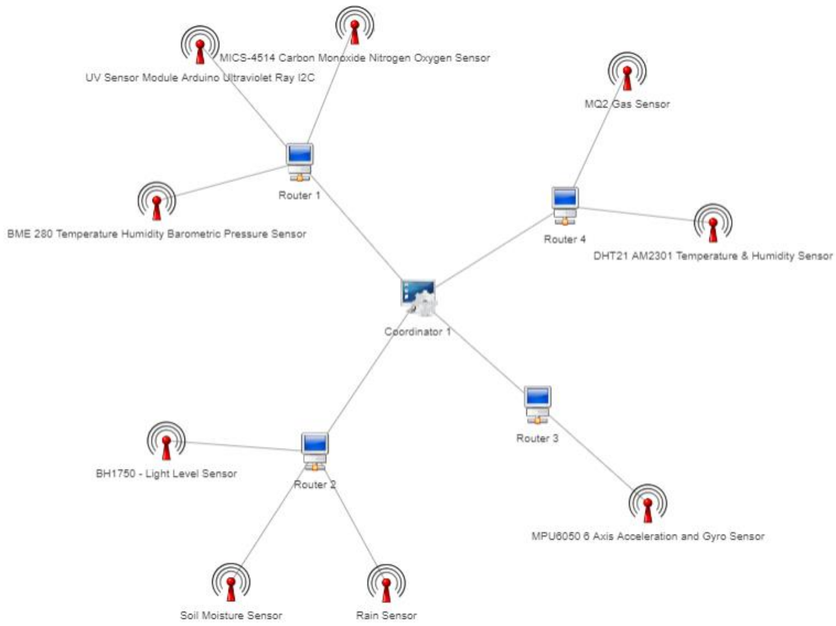


Figure 3.2 Established Wireless Sensor Network

Weather temperature, weather humidity, weather pressure, carbon monoxide, nitrogen, oxygen, and ultraviolet detection sensors have been plugged on router 1; digital light intensity, soil moisture, and precipitation detection sensors have been plugged on router 2; accelerometer and gyroscope sensor have been plugged on router 3; MQ2 gas sensor and DHT21 temperature-humidity sensor have been plugged on router 4. Table 3.1 illustrates which sensor measures what type of measurements, and which sensor is plugged on which router.

While some of these sensors measure different types of environmental events or changes at the same time, some of them measure only one event or change. Weather pressure, temperature, humidity changes have been detected by only one sensor named BME280. Likewise, rates of carbon monoxide, hydrocarbons, and oxidizing gases such as NO2 have been measured by only one sensor named MICS-4514. Arduino Uno microcontroller boards based on ATmega328P have been used to plug the sensors and measure their values in the system. XBee is one of the most popular modules which allows devices connect to each other through wireless. So, XBee Pro S2C modules have been seen an appropriate embedded solution for established wireless sensor network to transmit data from routers to the coordinator.

Table 3.1 Sensors plugged on routers

Router Name	Sensor Name	Measurements
Router 1	BME280 Temperature Humidity Barometric Pressure Sensor	Temperature
		Humidity
		Pressure
	GUVA-S12SD UV Sensor Module Arduino Ultraviolet Ray I2C	UV radiation in sunlight
	MICS-4514 Sensor	CO concentration
		Hydrocarbons
		Oxidizing gases
Router 2	Soil Moisture Sensor	Soil Moisture
	BH1750 - Light Level Sensor	Light Intensity
	Rain Sensor	Rain detection and rainfall intensity measurement
Router 3	MPU6050 (Gyroscope + Accelerometer + Temperature) Sensor Module	Accelerometer Gyroscope Temperature
Router 4	DHT221 Temperature and Humidity Sensor	Temperature
		Humidity
	MQ2 Gas Sensor	Hydrogen Concentration
		LPG Concentration
		Carbon monoxide Concentration
		Alcohol Concentration
		Propane Concentration

The coordinator consists of a Raspberry Pi 3 small single-board computer, an Arduino Uno, and an XBee Pro S2C module. Raspberry Pi has been included in WSN to run a universal windows application, which is in charge of reading, visualizing, and storing real-time data stream obtained from sensors. This application communicates with the coordinator device to read the data stream through serial port. The real-time data stream, which is visualized in favorable visualization objects by universal application, are stored in databases through web services. Figure 3.3 illustrates how sensor measurements are visualized by universal windows application on Raspberry Pi.

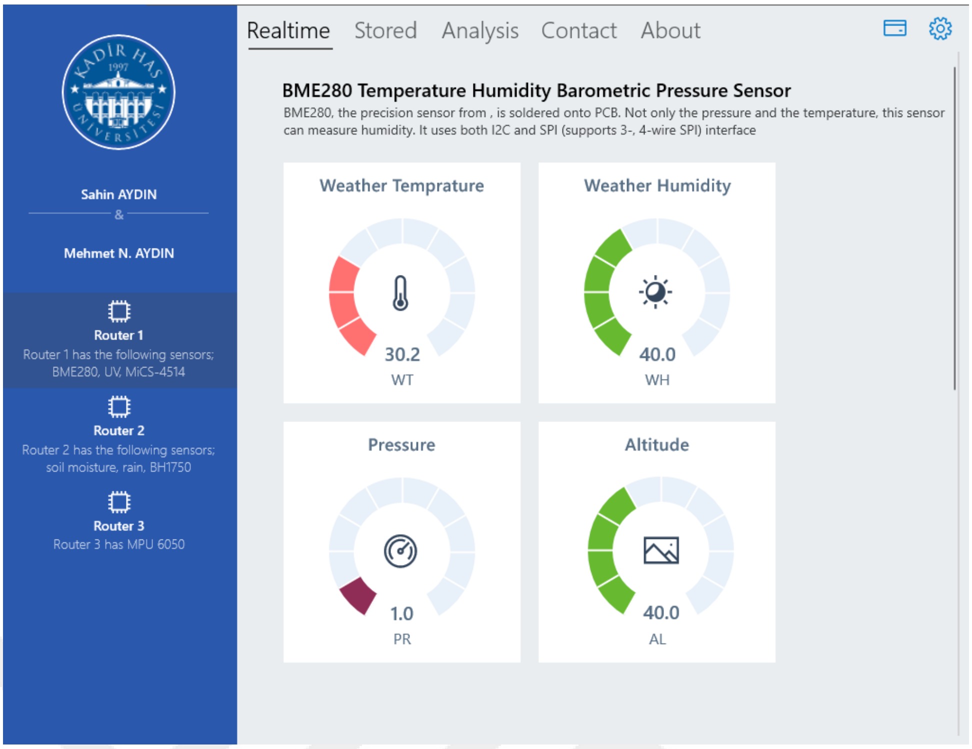


Figure 3.3 Screenshot from Universal App run on Raspberry Pi

ZigBee is an international open standard and communication protocol, which depends on IEEE 802.15.4 and ensures low-power, low-cost, low-data-rate, and wireless mesh networking (DIGI International Inc, 2018). Zigbee has three kinds of node types, such as coordinator, router, and end device. However, the developed WSN does not include any end device. Each of the XBee modules belonging to WSN were configured to API operating mode, which provides data transmission to multiple destinations with no requirement, using command mode. Receiving success and failure status of each transmitted RF packet is facilitated in this mode. In addition, the operation of identifying the source address of each received packet is readily provided by this mode. The routers transmit data, which is gathered via sensors, in an exclusive format named frame, represented as a ZigBee packet to the coordinator. The structure of the frame data; in other words, a ZigBee packet, which is used by the routers to transmit sensor data stream towards the coordinator, is demonstrated in Figure 3.4. The structure of frame data differs in compliance with the objectives of the API frame. The frame data structure of the established wireless sensor network, represented in Figure 3.4, consists of the fields, such as start delimiter, length, frame type, frame id, 64-bit destination address, options, RF data, and checksum.

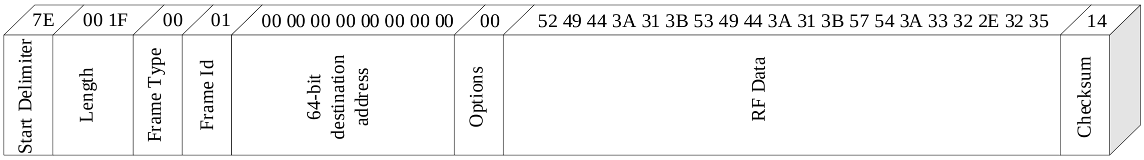


Figure 3.4 Frame Data Format

The sensor data are transmitted from the routers to the coordinator using key-value pair identifiers. For instance, the frame data, represented in Figure 3.4, refers to “RID:1;SID:1;WT:32.25”. The key-value pairs of this frame are RID:1, SID:1, and WT:32.25, respectively. Likewise, the values are 1, 1, and 32.25 as well. The keys, RID, SID, and WT stand for router id, sensor id, and temperature, respectively.

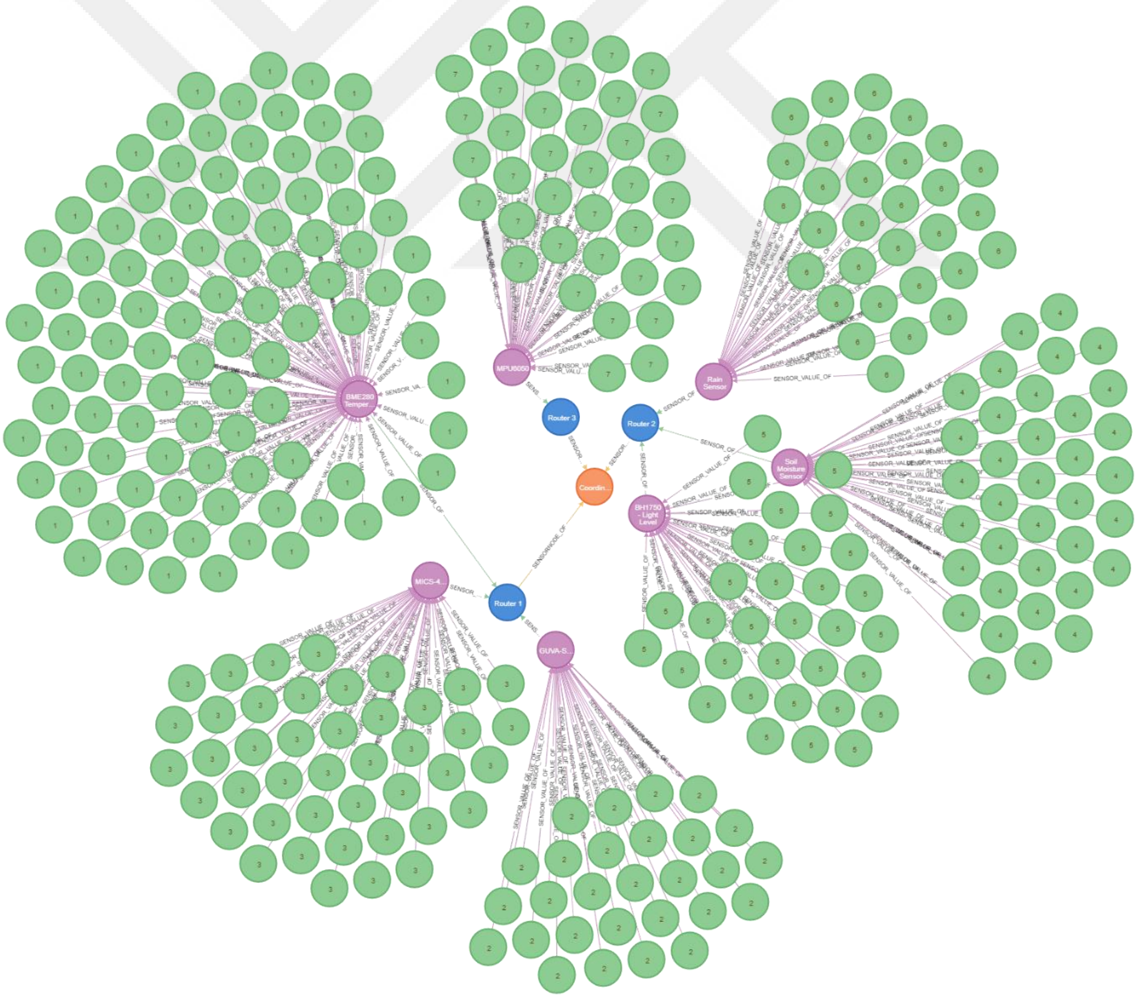


Figure 3.5 Storing sensor data in Neo4j Graph Database

When the data are transmitted to the coordinator and transformed into a meaningful format by universal application, storage operations are performed by DPL. Data are stored in two different types of database management systems, such as Microsoft MS SQL Server and Neo4j graph database system, which are a component of data storage layer. Neo4j, which is a graph database management system, increases the system performance, provides flexibility and agility. Figure 3.5 presents the data stored in Neo4j.

Table 3.2 Example of JSON data


```
[{"Id": 2,
   "SensorDataAddedDatetime": "\/Date(1580635128430+0300)\/",
   "SensorDataType": 2,
   "SensorDataValue": "32.45",
   "SensorId": 1000 }]
```


RDF file-based data storage option is used to store and export sensor data for this case study. Table 3.3, Table 3.4, Table 3.5 represent RDF/XML, Notation 3, and N-Triples datasets, respectively. These datasets can be created using the sensor listing UI of the open data platform which shall be introduced in further sections.

Table 3.3 Example of RDF/XML dataset


```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE rdf:RDF [
        <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
        <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
        <!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
        <!ENTITY sensorData 'http://www.opendatainagriculture.com/sensors#'>
]>
<rdf:RDF xml:base="http://www.opendatainagriculture.com/sensors"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:sensorData="http://www.opendatainagriculture.com/sensors#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description
rdf:about="http://www.opendatainagriculture.com/sensors/datatype#WeatherTemperature">
    <sensorData:DataTypeId>2</sensorData:DataTypeId>
    <sensorData:Datetime>2/2/2020 10:04:07 PM</sensorData:Datetime>
    <sensorData:SensorTypeId>1000</sensorData:SensorTypeId>
    <sensorData:SensorTypeName>BME 280 Temperature Humidity Barometric Pressure
Sensor</sensorData:SensorTypeName>
    <sensorData:TypeDef>WT</sensorData:TypeDef>
    <sensorData:TypeName>Weather Temprature</sensorData:TypeName>
    <sensorData:Value>32.434</sensorData:Value>
  </rdf:Description>
</rdf:RDF>
```


The data gathered from sensors and stored on databases must be published in a suitable format for applications running on different platforms. Therefore, REST web services were developed using Windows Communication Foundation (WCF), which is a framework for building service-oriented applications. Furthermore, web services for Windows platforms were developed in the same way. Table 3.2 illustrates the formatted data in JSON (JavaScript Object Notation) format. Developed web services are accessible to stakeholders.

Table 3.4 Example of Notation 3 dataset

```
@base <http://www.opendatainagriculture.com/sensors>.

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix sensorData: <http://www.opendatainagriculture.com/sensors#>.

<http://www.opendatainagriculture.com/sensors/datatype#WeatherTemperature>
sensorData:DataTypeId "2";
                              sensorData:Datetime "2/2/2020 10:04:07 PM";
                              sensorData:SensorTypeId "1000";
                              sensorData:SensorTypeName "BME 280 Temperature Humidity
Barometric Pressure Sensor";
                              sensorData:TypeDef "WT";
                              sensorData:TypeName "Weather Temprature";
                              sensorData:Value "32.434".
```

Table 3.5 Example of N-Triples dataset

```
<http://www.opendatainagriculture.com/sensors/datatype#WeatherTemperature>
<http://www.opendatainagriculture.com/sensors#DataTypeId> "2”.
<http://www.opendatainagriculture.com/sensors/datatype#WeatherTemperature>
<http://www.opendatainagriculture.com/sensors#SensorTypeId> "1000”.
<http://www.opendatainagriculture.com/sensors/datatype#WeatherTemperature>
<http://www.opendatainagriculture.com/sensors#SensorTypeName> "BME 280 Temperature
Humidity Barometric Pressure Sensor".
<http://www.opendatainagriculture.com/sensors/datatype#WeatherTemperature>
<http://www.opendatainagriculture.com/sensors#TypeName> "Weather Temperature”.
<http://www.opendatainagriculture.com/sensors/datatype#WeatherTemperature>
<http://www.opendatainagriculture.com/sensors#TypeDef> "WT”.
<http://www.opendatainagriculture.com/sensors/datatype#WeatherTemperature>
<http://www.opendatainagriculture.com/sensors#Value> "32.434”.
<http://www.opendatainagriculture.com/sensors/datatype#WeatherTemperature>
<http://www.opendatainagriculture.com/sensors#Datetime> "2/2/2020 10:04:07 PM”.
```

Created RDF datasets might be manipulated using SPARQL query language through SPARQL query service. This service filters files in compliance with the “.RDF” extension and creates buttons for each of them. Stakeholders should click to load the relevant dataset on SPARQL query service engine. After the loading process is completed, it should be written to execute the query into the text area. SPARQL query service shall be introduced in further section of the thesis.

In this section of the thesis, a multi-layered open data processing system that consists of different layers and components has been proposed. All layers and its components were explained; and an implementation regarding sensors was developed and summarized in the implementation section. Only one aspect has been examined within the scope of this implementation. However, there are four other aspects in terms of heterogeneous data sources. These other aspects are worth discussing and examining in the future. Also, it is tried to justify the suitability of the data life cycle in terms of the proposed multi-layered model by implementing the wireless sensor network system. This section enhances academic understanding of creating a multi-layered open data processing model. However, the life cycle of other data from other data sources should be examined and justified for viability. An ontology for hazelnut shall be introduced and adapted to the model. In addition, an open data platform based on the proposed model shall be introduced in further sections of the thesis. Stakeholders will be able to create data acquisition forms using agricultural trait dictionaries to gather structured data regarding the relevant agricultural product without requiring strong technical skills. The platform will allow obtaining environmental data from heterogenous sensors and will be able to map these sensor measurements with relevant classes of agricultural trait dictionaries. By this means, domain-specific data will be able to be published in varied open formats using the developed platform.

3.6 Limitations

In this chapter, a multi-layered open data processing model has been proposed for agriculture domain considering heterogeneous data sources, data processing, data storage, semantic annotation, web services / APIs, software applications and domain stakeholders (or users). On one hand, the validation of the proposed model has been demonstrated in the implementation part of this chapter for only one aspect by establishing an IoT platform which provides environmental data. On the other hand, data gathered from farmers, government statistical data, and market data have not been examined in this thesis.

There are some limitations in terms of demonstrating how to collect data from farmers and how farmers utilize processed data. For instance, required data may not be collected from farmers due to their lack of knowledge concerning technology usage. Similarly, they may not believe that the ICT applications developed based on the proposed model can solve their real-life problems concerning the agricultural production. Herein, it should be noted that technology adoption on agricultural activities is quite important in terms of broadening the usage of agricultural technologies. On the other hand, defining the boundaries of gathering market data can be a limitation. Due to it is required much more efforts achieving how to collect, process, annotate agricultural market data and how to make such data available for entire domain stakeholders, handling market data is another limitation. It is worth bearing in mind that the diversity of data sources in agriculture domain will be increase in future. However, it does not seem possible to determine these data sources for today. So, specifying other data sources is another limitation of the proposed model.

4. SEMANTIC WEB AND ONTOLOGIES IN AGRICULTURE

4.1 Semantic Web and Ontologies

Given the fact that agricultural data are gathered from various data sources, in different kinds of formats, and by using different metadata, it is complicated to use the relevant data in an efficient way. Eliminating heterogeneity of data sources in the agricultural domain is possible by using semantic web technologies. Semantic web targets ensure significant contents' format, which could be processed by both humans and machines (Szilaghyi and Wira, 2016). The inventor of World Wide Web (WWW), Tim Berners-Lee, defines semantic web as an extension of the current one which enhances straightforward information regarding the things on the web; and also, it enables human and machine collaboration (Berners-Lee et al., 2001). The major objective of the semantic web is to provide semantically annotated, structured, and semi-structured data into particular formats which are processable by both humans and machines (Szilaghyi and Wira, 2016) (Charalampidis and Keramopoulos, 2018). Several standards belonging to the semantic web, such as RDF, OWL (web ontology language), and SPARQL, ease the creation of data stores, construction of vocabularies concerning certain domains, and generation of rules to handle data (W3C, 2001). RDF, OWL, and SPARQL are well-known and widely used common standards of the semantic web. Resource Description Framework (RDF) is the base language of semantic web and represents well-defined, interrelated information concerning the web resources in triple format which consists of subject, object, and predicate (Manola and Miller, 2004) (Pollock, 2009) (Zimmermann et al., 2011). Furthermore, it is a standard data and modelling specification used to encode metadata and digital information. Web Ontology Language (OWL), which builds on and extends RDF, adds some particular terminologies such as classes, facts concerning the classes, and relationships between the classes or instances (Pollock, 2009).

The strength of OWL is to be a computational logic-based language which means the knowledge specified within it might be utilized by computer programs (W3C OWL, 2001). OWL 2, which is the extension and revision of the previous version published in 2004, is developed by W3C Web Ontology Working Group. Its first edition was published in 2009; and the second edition was published in 2012 (W3C OWL, 2001). There exist three sublanguages of OWL to meet different requirements such as OWL Full, OWL DL, and OWL Lite (Antoniou and Harmelen, 2008) (Allemang and Hendler, 2011). OWL is the common, well-known, and standard language for developing ontologies which might be identified as OWL documents. Ontology is a practical and systematic way of the conceptualization to enhance a common understanding of represented things (Gruber, 1993). Conceptualization means combining a group of objects, concepts, and various entities used to explain knowledge by providing the association between them (Noy and Hafner, 1997). Ontologies are not only used to express a particular domain or knowledge. As conceptual models, ontologies also provide interoperability between the diverse software applications (Cristina and Cuel, 2005). Creating ontologies in the field of agriculture are generally performed to provide the common knowledge of a particular agricultural product. Hazelnut Trait Ontology, which is created to represent the metadata regarding hazelnut, to carry out the interoperability between heterogeneous agricultural data sources and shall be elaborated later on is developed by using OWL.

The aim of generating ontologies is not merely to identify the conceptualization or point out the widespread savvy of a particular domain. In addition, they are utilized as conceptual models which provide common understanding to facilitate communicating between varied types of software application systems and among people (Cristina et al., 2005). Nowadays, there exist several ontologies which are still being in use, concerning the agriculture domain.

There exist diverse ontologies concerning agriculture domain. In the further part of this section, they shall be introduced.

4.2 Agricultural Ontologies

There exist diverse ontologies concerning agriculture domain such as AGROVOC, which was initially published in the early 1980s by the Food and Agriculture Organization (FAO) of the United Nations. Furthermore, it is the most popular and well-known agricultural thesaurus all over the world. Today, AGROVOC is available as an SKOS-

XL (Simple Knowledge Organization System eXtension for Labels) concept scheme. It is also published as a Linked Data (LD) set composed of 35,000+ concepts available in up to 29 languages (About AGROVOC, 2019). SKOS (Simple Knowledge Organization System) was developed by the Institute for Learning & Research Technology to provide a means for representing knowledge organization systems in a distributed and linkable way (Allemang and Hendler, 2008). SKOS aims providing linked and distributed knowledge organization systems such as controlled vocabularies, thesauri, taxonomies, and folksonomies (Allemang and Hendler, 2011). SKOS-XL defines an extension for the SKOS, providing additional support for describing and linking lexical entities (SKOS-XL Namespace Document, 2019).

Another prominent representative of agricultural ontology is Crop Ontology (CO). The CO is designed to provide a structured, controlled vocabulary for the phenotype of important crops for food and agriculture; and it is collectively developed by various Crop Communities, associated with the centers of the Consultative Group on International Agricultural Research (CGIAR) (Arnaud et al., 2012). It is the creation of the Generation Challenge Programme (GCP), which understood from its inception the importance of controlled vocabularies and ontologies for the digital annotation of data (Crop Ontology Curation Tool, 2019). CO has generally two resources OBO (Open Biological and Biomedical Ontology) ontologies and trait dictionaries. These resources are collected under five various groups, which are General Germplasm Ontology, Phonotype and Trait Ontology, Structural and Functional Genomic Ontology, Location and Environmental Ontology, and Plant Anatomy & Development Ontology. Some of the trait dictionaries and ontologies of CO were examined and analyzed to see whether they are suitable for hazelnut domain or not by the guidance of domain experts. However, the arguments proposed by domain experts have proved that any ontology is not appropriate for hazelnut. Even though there exist various agricultural ontologies created by many authors apart from these popular ones, when taking into consideration agricultural products' descriptors such as passport, management, characterization, evaluation, environment and site, they are not favorable to use them for hazelnut domain. Before creating Hazelnut Trait Ontology, it was considered that whether the trait dictionaries and phenotypes published on the web site of CO, are suitable for hazelnut or not.

Table 4.1 Trait dictionaries and phenotypes published by CO (Crop Ontology Curation Tool, 2019)

Phenotype and Trait Ontology	Number of Variables / Traits	Abiotic stress	Agronomical	Biochemical	Biotic stress	Morphological	Phenological	Physiological	Quality	Fertility	Other Traits
Bambra grounnut	134 variables	√	√	√	√	√	√	√	√	X	
Banana	370 variables	X	√	√	√	√	√	√	√	X	
Barley	148 variables	√	√	X	√	√	√	√	√	X	
Barley Trait POLAPGEN Ontology	148 traits	X	√	√	X	√	X	X	X	X	
Beet Ontology	385 variables	√	√	X	√	√	X	√	√	√	
Brachiaria	82 variables	√	√	√	√	√	X	√	√	X	
Brassica	155 variables	√	√	√	√	√	√	X	√	X	
Cassava f	245 variables	X	√	X	√	√	X	√	√	X	
Castor bean	75 variables	√	√	X	√	√	√	X	√	X	
Chickpea	89 variables	√	√	X	√	√	X	√	√	X	
Common Bean	184 traits	X	√	√	√	√	√	√	√	X	
Cotton	282 variables	√	√	√	√	√	√	√	√	X	Processing
Cowpea	204 variables	√	√	X	√	√	√	X	√	X	
Fababean	94 variables	√	√	X	√	√	√	√	√	X	
Groundnut	101 variables	X	√	X	√	√	X	√	√	X	
Lentil	68 variables	√	√	X	√	√	√	√	√	X	
Maize	352 variables	√	√	X	√	√	√	√	√	X	
Mungbean	89 variables	√	√	X	√	√	√	√	√	X	
Oat	228 variables	√	√	√	√	√	√	√	√	X	
Pearl millet	52 variables	X	√	X	√	√	√	X	X	√	
Pigeonpea	73 variables	√	√	X	√	√	√	√	√	X	
Potato	197 variables	√	√	X	√	√	X	X	√	X	Crop research

Rice	405 variables	√	√	√	√	√	√	√	√	√	
Sorghum	179 variables	X	√	X	√	√	√	X	√	X	
SoyBase Soybean Trait Ontology	678 terms	√		√	√	√			√	√	
Soybean	90 variables	√	√	√	√	√	√	X	X	X	
Sugar Kelp trait	46 variables	X	√	√	X	√	√	√	X	X	
Sunflower	353 variables	√	X	√	√	X	√	√	√	X	
Sweet Potato	303 variables	√	√	√	√	√	X	√	√	X	Crop research and Harvest
Vitis	273 variables	√	√	√	√	√	√	X	X	X	Technological
Wheat	498 variables	√	√	X	√	√	√	√	√	X	
Woody Plant Ontology	384 variables	√	√	√	√	√	√	√	√	√	
Yam	191 variables	X	√	X	√	√	√	√	√	X	

However, the traits which are examined by these ontologies do not meet the requirements of stakeholders in hazelnut domain. Table 4.1 represents all the trait dictionaries and phenotypes published by CO. As Table 4.1 shows, CO considers traits and variables into nine different groups and almost each of trait dictionaries include all of these groups. However, while creating Hazelnut Trait Ontology the following categories are used Passport, Management, Environment and Site, Characterization, and Evaluation. These categories are broader top categories used by Bioversity for defining genetic resource. They also include the nine groups illustrated on Table 4.1 as well. Considering the needs of stakeholders of hazelnut domain, it was inevitable to create a comprehensive ontology. So, we decided to create Hazelnut Trait Ontology, and introduce it in the next chapter of thesis.

4.3 Hazelnut Trait Ontology

One of the major contributions of thesis is the development of an agricultural ontology which provides common understanding and vocabulary concerning hazelnut. Therefore, a comprehensive hazelnut trait dictionary has been developed within the scope of the thesis. It has been used for collecting domain-specific data and providing appropriate intelligence in the context of precise agriculture. In addition, it has been utilized to accomplish the following purposes: (i) to be a reference (knowledge) model to develop web-based dynamic data acquisition forms for the hazelnut domain; (ii) to provide making inferences concerning hazelnut agricultural product; (iii) to provide and enhance data integration from heterogeneous data sources towards knowledge-based systems; (iv) to produce linked open data in open formats using open standards.

Agriculture is a vital sector due to the contribution to employment, exportation, and domestic income; providing raw materials sources for industry. According to Statistical Yearbook of the Food and Agriculture Organization for the United Nations, more than 3 billion people–almost half of the world’s population–live in rural areas; and roughly 2.5 billion of these rural people derive their livelihoods from agriculture (FAO, 2014). One of the much significant agricultural products is hazelnut as well in the world. Hazelnut is produced almost one billion tons per year dominantly in Turkey, Italy, the USA, Azerbaijan, Georgia, and Spain. In the world, there are many stakeholders directly or indirectly related to hazelnut such as farmers, researchers, analysts, domain experts and exporters. When the importance of hazelnut agricultural product is taken into consideration, gathering much more detailed data regarding it and publishing this data for stakeholders of the relevant domain to use are indispensable. There is, therefore, a definite need for developing an ontology regarding hazelnut. The main purpose of developing Hazelnut Trait Ontology is to share a common vocabulary. On the other hand, Hazelnut Trait Ontology is acknowledged that it is going to help provide an international format to standardize general understanding with respect to hazelnut. Another purpose of creating an ontology regarding hazelnut provides a generally accepted common language for hazelnut. It is a generally complicated process to gather data concerning a specific domain. This ontology is created to help stakeholders while deciding which attributes should be defined within the gathered data as well. Hazelnut Trait Ontology is created to contribute to facilitating data storage, data retrieval, and data exchange in a rapid, reliable, and proper way by publishing accurate metadata with different types of services. While developing Hazelnut Trait Ontology we have examined three types of approaches such as top-down, bottom-up, and middle-out. According to the top-down approach, the initial work for building ontology is to begin the process by modelling top level concepts (Francesconi et al., 2010). The bottom-up approach assumes that the documents include concepts and conceptual structures of the domain as well as the needed terminology to express them (Francesconi et al., 2010). In the middle-out approach, the process of building the ontology started with the most important concepts first, and defining higher level concepts in terms of these, as the higher-level categories naturally arise (Uschold and Gruninger, 1996). Relevance of these approaches to ontology development is to identify the concepts as follows: from the most concrete to the most abstract (bottom-up), from the most abstract to the most concrete (top-down), or from the most relevant to the most abstract and most concrete (middle-out) (Fernandez-Lopez, 1999). We will examine a number of methodologies adopting the approaches for the purpose of assessing the Hazelnut Trait Ontology in detail.

DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering), which is a foundational ontologies library, acts as a starting point for comparing and elucidating the relationships with other future modules of the library, and also clarifying the hidden assumptions underlying existing ontologies or linguistic resources (Masolo et al., 2002). On-To-Knowledge project known as OTK Methodology develops, explores sophisticated methods and tools for Knowledge Management, provides infrastructure for the Semantic Web, and is a process-oriented methodology for introducing and maintaining ontology-based knowledge management solutions in enterprises (Sure and Studer, 2002). METHONTOLOGY is a popular and well-structured methodology used for building ontologies from scratch. METHONTOLOGY guides in how to carry out the whole ontology development through the specification, the conceptualization, the formalization, the implementation and the maintenance of the ontology; and it also identifies management activities (schedule, control, and quality assurance) and support activities (knowledge acquisition, integration, evaluation, documentation, and configuration management) (Corcho et al., 2005). TOVE (Toronto Virtual Enterprise) project aims to build ontologies that model both commercial and public enterprises. Also the goals of TOVE are to create a shared representation (aka ontology) of the enterprise which each agent in the distributed enterprise can jointly understand and use; to define the meaning of each description (aka semantics); to implement the semantics in a set of axioms that will enable TOVE to automatically deduce the answer to many common sense questions about the enterprise; and to define a symbology for depicting a concept in a graphical context (Fox, 1992). DILIGENT (Distributed, Loosely-controlled and Evolving Engineering of Ontologies) methodology provides a process template suitable for distributed engineering of knowledge structures that is planned to extend towards a fully worked out and repeatedly tested methodology in the long run; and it comprises five main activities of ontology engineering: building, local adaptation, analysis, revision, and local update (Denny et al., 2005). Business Object Ontology is used in (Izumi and Yamaguchi, 2001) to propose an integrated support methodology for constructing business models, including employing new business models, transplanting existing business activities to computers, and decision-making support in employing a new environment of computers. A Natural Language Interface Generator (GISE), which is designed to provide a natural language access to a variety of Expert Systems, generates natural language interfaces that can support dialogues around the ontology (Gatius and Rodríguez, 1996). It uses three steps for developing of natural language interfaces and ontologies. There are many more methodologies such as Enterprise Model Approach, KBSI IDEF5, Ontolingua, CommonKADS and KACTUS, Plinus, Onions, Mikrokosmos, MENELAS, PHYSSYS, and lastly SENSUS for building ontologies (Jones et al., 1998). However, we shall not give the definitions of these methodologies in this thesis.

The methodology of Ontology Development 101 was created considering an iterative approach for building ontologies and has seven steps. It has been preferred to make use of developing Hazelnut Trait Ontology using the Protégé tool within the scope of this research. These steps are as follows: determining the domain and scope of the ontology, considering to reuse existing ontologies, enumerating important terms in the ontology, defining the classes and the class hierarchy, defining the properties of classes—slots, defining the facets of the slots, and creating instances (Noy and McGuinness, 2001). The main purpose of this study is to develop an agricultural domain ontology and evaluate the quality of this ontology. That's why, for the first step of the relevant methodology, agriculture has been determined as the domain of creating ontology. Furthermore, this section of the thesis aims to contribute to Hazelnut Farming area of research by developing an ontology. As the existing ontologies are not appropriate for hazelnut agricultural product, we could not make use of these existing ontologies while developing the ontology. The most comprehensive document to determine the metadata regarding hazelnut is "*Descriptors for Hazelnut*". Therefore, this document is used to enumerate important terms in the ontology. According to this document, Bioversity International (formerly known as IPGRI), which is an independent international scientific organization, uses the following five types of descriptors: passport, management, environment and site, characterization and evaluation descriptors (Koksal and Gunes, 2008). These descriptors are classes of the ontology, and also, they designate the class hierarchy with the sub descriptors. Meanwhile, it should be noted that each descriptor has several sub descriptors. Some of these sub descriptors are classes of the ontology, but the others are the properties of classes-slot. The definitions and number of sub descriptors of each main descriptor are demonstrated on Table 4.2.

The descriptors demonstrated on Table 4.2 are in a hierarchical structure that means subclass-superclass relationships within the ontology, so while creating Hazelnut Trait Ontology, this hierarchy should be considered. Table 4.2 presents an overview of the major descriptors of hazelnut. However, there are many descriptors regarding these major descriptors within the document. The most general concept of Hazelnut Trait Ontology is Descriptor; and it has five types of general top-level concepts: Passport, Characterization, Environment and Site, Evaluation and Management. The other concepts that belonged to top level concepts are described as middle level; and the concepts related to middle level ones are named the bottom level concepts. Figure 4.1 demonstrates top level concepts of the Hazelnut Trait Ontology.

Ontology creation is based on iterative design, and complicated and evolutionary process. There is no doubt that many domain experts shall contribute to Hazelnut Trait Ontology. This part of the thesis begins by laying out introducing the Hazelnut Trait Ontology and looks at why we need to create it. The next part is concerned with the tools and methodologies used to evaluate the quality for ontologies and analyzing the meanings of the results calculated by these tools and methodologies.

4.3.1 Entities and Metrics of Hazelnut Trait Ontology

Hazelnut Trait Ontology consists of 2255 axioms, 1406 logical axioms, 740 declaration axioms, 304 classes, 45 object properties, 19 data properties, and 371 individuals. Hazelnut Trait Ontology aims to provide common descriptors concerning hazelnut for domain stakeholders. Therefore, the top-level concept is defined as “Descriptor”. The top-level concept named as “Descriptor” consists of five sub-classes named as “Passport”, “Characterization”, “EnvironementAndSite”, “Evaluation”, and “Management”. Figure 4.1 illustrates the hierarchy between the top-level concepts of Hazelnut Trait Ontology.

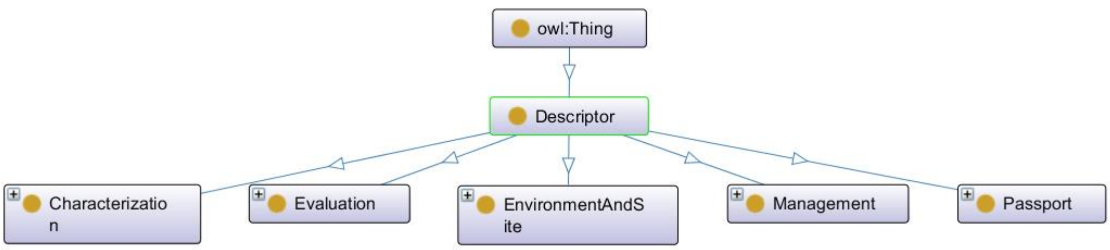


Figure 4.1 Top-level concepts of Hazelnut Trait Dictionary

Table 4.2 represents the top-level classes as Turtle syntax and their explanations.

Table 4.2 Descriptions of Top-Level Concepts

:Descriptor rdf:type owl:Class ;	As a top-level concept "*Descriptor*" is the root class of Hazelnut Trait Dictionary. It consists of five different sub-classes which describe hazelnut agricultural product in different context. The creators of hazelnut descriptors are Professor Dr. A. Ilhami Koksal and Dr. Nurdan Tuna Gunes. According to them "*descriptors should be used when they are useful to the curator for the management and maintenance of the collection or to the users of the plant genetic resources, or both*" (Koksal and Gunes, 2008).
:Characterization rdf:type owl:Class ; rdfs:subClassOf :Descriptor ;	The "*Characterization*" class is one of the top-level concepts of Hazelnut Trait Dictionary. The "*Characterization*" class with its sub-classes are used to provide specifying the differences between phenotypes in an easy way (Koksal and Gunes, 2008).
:Evaluation rdf:type owl:Class ; rdfs:subClassOf :Descriptor ;	The "*Evaluation*" class and its sub-classes might be accepted as the most interesting descriptors in terms of improving crop (Koksal and Gunes, 2008).
:EnvironmentAndSite rdf:type owl:Class ; rdfs:subClassOf :Descriptor ;	In order to identify the environmental and site-specific parameters concerning hazelnut agricultural product, the "*EnvironmentAndSite* " class and its sub-classes are used (Koksal and Gunes, 2008).
:Management rdf:type owl:Class ; rdfs:subClassOf :Descriptor ;	The "*Management*" class and its sub-classes are used to manage accessions within the genebank and also, they are of assistance for multiplication and regeneration of the accessions (Koksal and Gunes, 2008).
:Passport rdf:type owl:Class ; rdfs:subClassOf :Descriptor ;	The "*Passport*" class and its sub-classes are the basic information to generally manage the accession and also, they are used to characterize parameters which should be observed when the

	accession is originally collected (Koksal and Gunes, 2008).

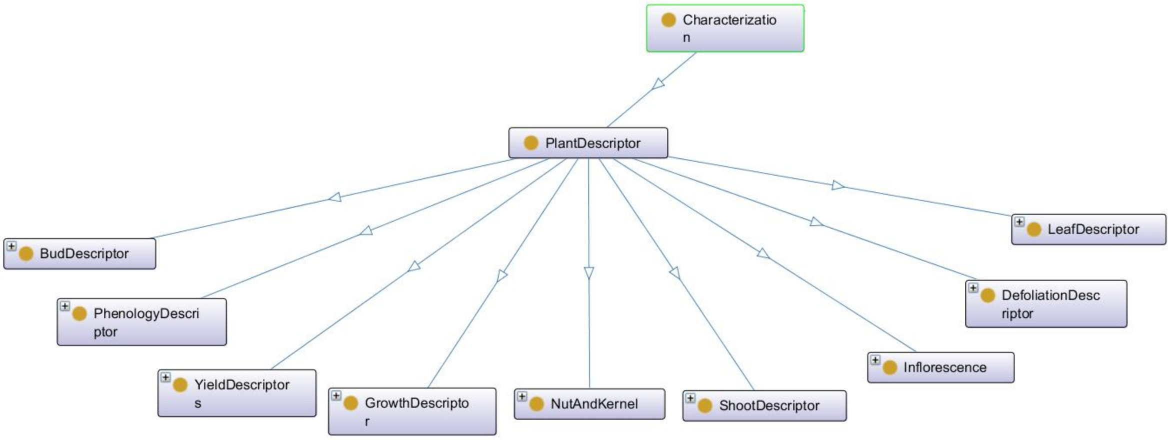


Figure 4.2 Characterization class and its sub-classes

The descriptor named "*Characterization*", which is the sub-class of "*Descriptor*" consists of only one main descriptor named "*PlantDescriptor*". The "*PlantDescriptor*" class has nine sub-classes. These sub-classes of "PlantDescriptor" and its sub-classes are represented in Table 4.3.

Table 4.3 The :PlantDescriptor Class and its sub-classes

:BudDescriptor rdf:type owl:Class ; rdfs:subClassOf :PlantDescriptor.	:BudShape rdf:type owl:Class ; rdfs:subClassOf :BudDescriptor . :BudColour rdf:type owl:Class ; rdfs:subClassOf :BudDescriptor .
:PhenologyDescriptor rdf:type owl:Class ; rdfs:subClassOf :PlantDescriptor .	:BloomingReferenceStandard rdf:type owl:Class ; rdfs:subClassOf :PhenologyDescriptor ; :DateOfVegetativeBudbreak rdf:type owl:Class ; rdfs:subClassOf :PhenologyDescriptor :Dichogamy rdf:type owl:Class ; rdfs:subClassOf :PhenologyDescriptor . :FemalePeakBloomDate rdf:type owl:Class ; rdfs:subClassOf :PhenologyDescriptor :FirstFemaleBloomDate rdf:type owl:Class ; rdfs:subClassOf :PhenologyDescriptor :FirstMaleBloomDate rdf:type owl:Class ; rdfs:subClassOf :PhenologyDescriptor

	:LastFemaleBloomDate rdf:type owl:Class ; rdfs:subClassOf :PhenologyDescriptor . :LastMaleBloomDate rdf:type owl:Class ; rdfs:subClassOf :PhenologyDescriptor :MalePeakBloomDate rdf:type owl:Class ; rdfs:subClassOf :PhenologyDescriptor :ReferenceStandard rdf:type owl:Class ; rdfs:subClassOf :PhenologyDescriptor
:YieldDescriptors rdf:type owl:Class ; rdfs:subClassOf :PlantDescriptor.	:HomogeneityOfNutRipening rdf:type owl:Class; rdfs:subClassOf :YieldDescriptors. :NutFalling rdf:type owl:Class; rdfs:subClassOf :YieldDescriptors. :NutMaturityDate rdf:type owl:Class; rdfs:subClassOf :YieldDescriptors. :TendencyToAlternateBearing rdf:type owl:Class; rdfs:subClassOf :YieldDescriptors. :YearsFromSuckerPlantingGraftOrSeedToFirstYield rdf:type owl:Class ; rdfs:subClassOf :YieldDescriptors.
:GrowthDescriptor rdf:type owl:Class ; rdfs:subClassOf :PlantDescriptor.	:BranchingDensity rdf:type owl:Class; rdfs:subClassOf :GrowthDescriptor. :Suckering rdf:type owl:Class; rdfs:subClassOf :GrowthDescriptor. :TreeGrowthHabit rdf:type owl:Class; rdfs:subClassOf :GrowthDescriptor. :TreeVigour rdf:type owl:Class ; rdfs:subClassOf :GrowthDescriptor.
:NutAndKernel rdf:type owl:Class; rdfs:subClassOf :PlantDescriptor.	100-KernelWeight rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel , 100-NutWeight rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel :BlankProduction rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel :BrownSpotsInKernelCavity rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel :CurvatureOfNutBasalScar rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel . :DoubleOrTwinKernels rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel :HairinessOfNutApex rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel .

```
:InvolucreConstriction rdf:type owl:Class ;
                    rdfs:subClassOf :NutAndKernel .

:InvolucreHairinessDensity rdf:type owl:Class ;
                        rdfs:subClassOf :NutAndKernel .

:InvolucreIndentation rdf:type owl:Class ;
                   rdfs:subClassOf :NutAndKernel .

:InvolucreLengthComparedToNutLength rdf:type ow
l:Class ;
             rdfs:subClassOf :NutAndKernel .

:InvolucreThicknessAtBaseOfInvolucre rdf:type owl:
Class ;
              rdfs:subClassOf :NutAndKernel .

:JointingOfBractsOnInvolucre rdf:type owl:Class ;
               rdfs:subClassOf :NutAndKernel .

:KernelBlanching rdf:type owl:Class ;
              rdfs:subClassOf :NutAndKernel .

KernelDryWeightNutDryWeight×100 rdf:type owl:C
lass ;
         rdfs:subClassOf :NutAndKernel

:KernelFibreTexture rdf:type owl:Class ;
                rdfs:subClassOf :NutAndKernel .

:KernelFlavour rdf:type owl:Class ;
             rdfs:subClassOf :NutAndKernel .

:KernelLength rdf:type owl:Class ;
            rdfs:subClassOf :NutAndKernel.

:KernelPlumpness rdf:type owl:Class ;
              rdfs:subClassOf :NutAndKernel.

:KernelShape rdf:type owl:Class ;
           rdfs:subClassOf :NutAndKernel .

:KernelThickness rdf:type owl:Class ;
              rdfs:subClassOf :NutAndKernel.

:KernelWidth rdf:type owl:Class ;
           rdfs:subClassOf :NutAndKernel.

:NumberOfNutsIn100g rdf:type owl:Class ;
                rdfs:subClassOf :NutAndKernel.

:NutApexProminence rdf:type owl:Class ;
               rdfs:subClassOf :NutAndKernel .

:NutLength rdf:type owl:Class ;
         rdfs:subClassOf :NutAndKernel.

:NutShape rdf:type owl:Class ;
```

	rdfs:subClassOf :NutAndKernel. :NutShapeOfCrossSection rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel . :NutShellColour rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel . :NutThickness rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel. :NutWidth rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel. :PredominantNutNumberPerCluster rdf:type owl:Cla ss ; rdfs:subClassOf :NutAndKernel. :SerrationOfIndentationsOnTheInvolucre rdf:type ow l:Class ; rdfs:subClassOf :NutAndKernel . :ShapeOfNutApex rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel . :ShellStriping rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel . :SizeOfInternalCavityOfKernel rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel . :SizeOfNutBasalScarInRelationToNutSize rdf:type o wl:Class ; rdfs:subClassOf :NutAndKernel . :SizeOfPistilScar rdf:type owl:Class ; rdfs:subClassOf :NutAndKernel .
:ShootDescriptor rdf:type owl:Class; rdfs:subClassOf :PlantDescriptor.	:OneYearOldShootDensityOfLenticels rdf:type owl: Class ; rdfs:subClassOf :ShootDescriptor . :OneYearOldShootHairiness rdf:type owl:Class ; rdfs:subClassOf :ShootDescriptor . :OneYearOldShootThickness rdf:type owl:Class ; rdfs:subClassOf :ShootDescriptor .
:Inflorescence rdf:type owl:Class ; rdfs:subClassOf :PlantDescriptor.	:CatkinAbundance rdf:type owl:Class ; rdfs:subClassOf :Inflorescence . :FemaleFlowerAbundance rdf:type owl:Class ; rdfs:subClassOf :Inflorescence . :FloweringPrecocity rdf:type owl:Class ; rdfs:subClassOf :Inflorescence. :InflorescenceBudDryWeight rdf:type owl:Class ; rdfs:subClassOf :Inflorescence.

	:StigmaColourOfYoungFlowers rdf:type owl:Class ; rdfs:subClassOf :Inflorescence .
:DefoliationDescriptor rdf:type owl:Class; rdfs:subClassOf :PlantDescriptor.	:BeginningOfDefoliation rdf:type owl:Class ; rdfs:subClassOf :DefoliationDescriptor . :DefoliationDate rdf:type owl:Class ; rdfs:subClassOf :DefoliationDescriptor .
:LeafDescriptor rdf:type owl:Class ; rdfs:subClassOf :PlantDescriptor.	:LeafBladeShape rdf:type owl:Class ; rdfs:subClassOf :LeafDescriptor. :LeafColour rdf:type owl:Class ; rdfs:subClassOf :LeafDescriptor. :LeafLength rdf:type owl:Class ; rdfs:subClassOf :LeafDescriptor. :LeafWidth rdf:type owl:Class ; rdfs:subClassOf :LeafDescriptor.

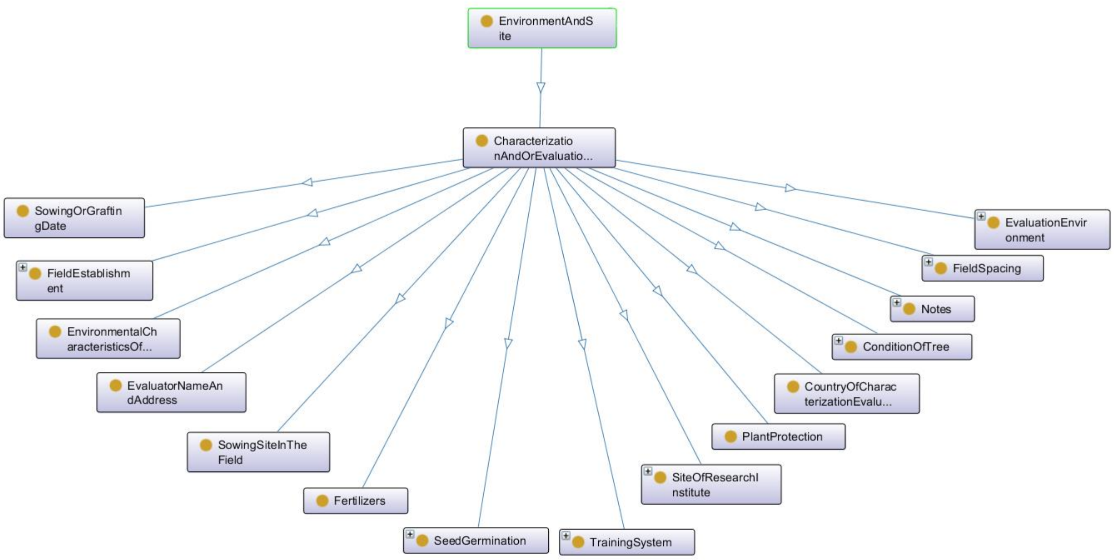


Figure 4.3 :EnvironmentAndSite class and its sub-classes

The descriptor named "*EnvironmentAndSite*", which is the sub-class of "*Descriptor*" consists of two main descriptors named "*CharacterizationAndOrEvaluationSiteDescriptor*" and "*CollectingAndOrCharacterizationEvaluationSiteEnvironmentDescriptor*".

The sub-classes of "*CharacterizationAndOrEvaluationSiteDescriptor*" class are represented in Table 4.4.

Table 4.4 The definitions of sub-classes of ":CharacterizationAndOrEvaluationSiteDescriptor"

:ConditionOfTree rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluationSiteDescriptor.	-
:CountryOfCharacterizationEvaluation rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluationSiteDescriptor.	-
:EnvironmentalCharacteristicsOfSite rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluationSiteDescriptor.	-
:EvaluationEnvironment rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluationSiteDescriptor .	-
:EvaluatorNameAndAddress rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluationSiteDescriptor .	-
:Fertilizers rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluationSiteDescriptor.	-
:FieldEstablishment rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluationSiteDescriptor.	:NumberOfDays rdf:type owl:Class ; rdfs:subClassOf :FieldEstablishment.
:FieldSpacing rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluationSiteDescriptor .	:DistanceBetweenRows rdf:type owl:Class ; rdfs:subClassOf :FieldSpacing :DistanceBetweenTreesInaRow rdf:type owl:Class ; rdfs:subClassOf :FieldSpacing .
:Notes rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluationSiteDescriptor.	-
:PlantProtection rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluationSiteDescriptor.	-
:SeedGermination rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluationSiteDescriptor.	:GerminationPercentage rdf:type owl:Class ; rdfs:subClassOf :SeedGermination. :NumberOfDays rdf:type owl:Class ; rdfs:subClassOf : SeedGermination.
:SiteOfResearchInstitute rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluationSiteDescriptor.	:Elevation rdf:type owl:Class ; rdfs:subClassOf :SiteOfResearchInstitute. :Latitude rdf:type owl:Class ; rdfs:subClassOf :SiteOfResearchInstitute. :Longitude rdf:type owl:Class ; rdfs:subClassOf :SiteOfResearchInstitute. :NameOfFarmOrInstitute rdf:type owl:Class ;

	rdfs:subClassOf :SiteOfResearc hInstitute.
:SowingOrGraftingDate rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluati onSiteDescriptor .	-
:SowingSiteInTheField rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluati onSiteDescriptor.	-
:TrainingSystem rdf:type owl:Class ; rdfs:subClassOf :CharacterizationAndOrEvaluati onSiteDescriptor.	-

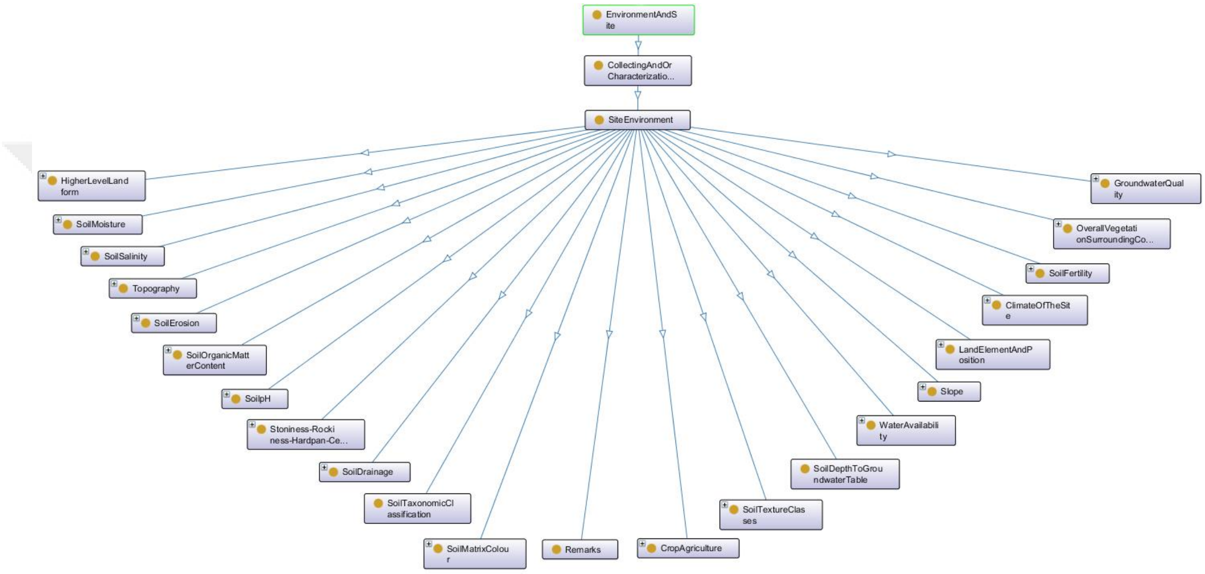


Figure 4.4 The sub-classes of ":SiteEnvironment"

The descriptor named "CollectingAndOrCharacterizationEvaluationSiteEnvironmentDescriptor", which is the sub-class of "EnvironmentAndSite" consists of one main descriptor named "*SiteEnvironment*".

The sub-classes of "*SiteEnvironment*" class are represented in Table 4.5.

Table 4.5 The definitions of sub-classes of ":SiteEnvironment" Class

:ClimateOfTheSite rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	:AtmosphericPressure rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite. :CarbonmonoxideConcentration rdf:type ow l:Class ; rdfs:subClassOf :ClimateOfTheSite.

	:NitrogendioxideConcentration rdf:type owl :Class ; rdfs:subClassOf :ClimateOfTheSite. :Ultraviolet rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite. :DayLength rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite. :Frost rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite . :Light rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite. :Rainfall rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite. :RelativeHumidity rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite. :Temperature rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite. :Wind rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite.
:CropAgriculture rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-
:GroundwaterQuality rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-
:HigherLevelLandform rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment	-
:LandElementAndPosition rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment	-
:OverallVegetationSurroundingCollectingSite rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-
:Remarks rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-
:Slope rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-
:SoilDepthToGroundwaterTable rdf:type owl:Class; rdfs:subClassOf :SiteEnvironment.	-
:SoilDrainage rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-
:SoilErosion rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-
:SoilFertility rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-
:SoilMatrixColour rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-
:SoilMoisture rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-
:SoilOrganicMatterContent rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-

:SoilpH rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	:RootDepth rdf:type owl:Class ; rdfs:subClassOf :SoilpH.
:SoilSalinity rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment .	:DissolvedSalts rdf:type owl:Class ; rdfs:subClassOf :SoilSalinity. :Electro-conductivity rdf:type owl:Class ; rdfs:subClassOf :SoilSalinity.
:SoilTaxonomicClassification rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment .	-
:SoilTextureClasses rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	:SoilParticleSizeClasses rdf:type owl:Class ; rdfs:subClassOf :SoilTextureClasses.
:Stoniness-Rockiness-Hardpan-Cementation rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-
:Topography rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment.	-
:WaterAvailability rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment .	-

The "*:Frost*" class consists of three sub-classes named "*:DataOfMostRecentFrost*", "*:DurationOfTemperatureBelowZeroDegree*", and "*:MinumumTemperature*". On the other hand, the "*:RelativeHumdity*" class has following two classes: "*:DiurnalRange*" and "*:SeasonalRange*". The "*:Wind*" class has similarly one sub-class named "*:AnnualMaxWindVelocity*".

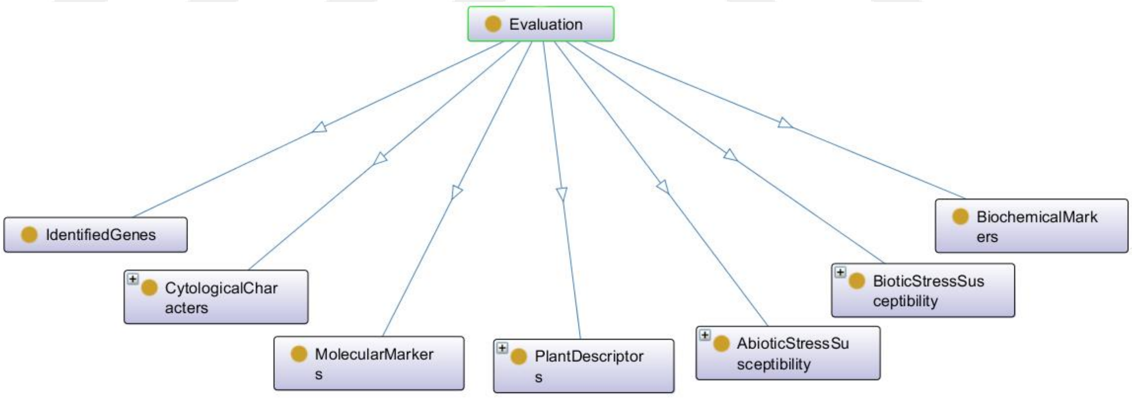


Figure 4.5 The sub-classes of ":Evaluation" Class

Figure 4.5 presents an overview of the sub-classes of "*Evaluation*" class. The table below illustrates the definitions of each sub-classes of "*Evaluation*" class.

Table 4.6 The definitions of sub-classes of ":Evaluation" Class

:AbioticStressSusceptibility rdf:type owl:Class ; rdfs:subClassOf :Evaluation.	:Drought rdf:type owl:Class ; rdfs:subClassOf :AbioticStressSusceptibility.

	:HighTemperature rdf:type owl:Class ; rdfs:subClassOf :AbioticStressSusceptibility. :LowTemperature rdf:type owl:Class ; rdfs:subClassOf :AbioticStressSusceptibility. :MineralDeficiency rdf:type owl:Class ; rdfs:subClassOf :AbioticStressSusceptibility. :MineralToxicity rdf:type owl:Class ; rdfs:subClassOf :AbioticStressSusceptibility. :Salinity rdf:type owl:Class ; rdfs:subClassOf :AbioticStressSusceptibility. :Waterlogging rdf:type owl:Class ; rdfs:subClassOf :AbioticStressSusceptibility.
:BiochemicalMarkers rdf:type owl:Class ; rdfs:subClassOf :Evaluation.	-
:BioticStressSusceptibility rdf:type owl:Cla ss ; rdfs:subClassOf :Evaluation.	:Bacteria rdf:type owl:Class ; rdfs:subClassOf :BioticStressSusceptibility. :Bud-feedingInsectsAndMites rdf:type owl:Class ; rdfs:subClassOf :BioticStressSusceptibility. :Flower-FeedingInsectsAndMites rdf:type owl:Class ; rdfs:subClassOf :BioticStressSusceptibility. :Foliage-FeedingInsectsAndMites rdf:type owl:Class ; rdfs:subClassOf :BioticStressSusceptibility. :Fruit-FeedingInsects rdf:type owl:Class ; rdfs:subClassOf :BioticStressSusceptibility. :Fungi rdf:type owl:Class ; rdfs:subClassOf :BioticStressSusceptibility. :Nematodes rdf:type owl:Class ; rdfs:subClassOf :BioticStressSusceptibility. :StemAndTrunkFeedingInsects rdf:type owl:Class ; rdfs:subClassOf :BioticStressSusceptibility. :VirusesViroidsAndPhytoplasmas rdf:type owl:Class ; rdfs:subClassOf :BioticStressSusceptibility.
:CytologicalCharacters rdf:type owl:Class ; rdfs:subClassOf :Evaluation.	:ChromosomeNumber rdf:type owl:Class ; rdfs:subClassOf :CytologicalCharacters. :MeiosisChromosomeAssociations rdf:type owl:Class ; rdfs:subClassOf :CytologicalCharacters. :OtherCytologicalCharacters rdf:type owl:Class ; rdfs:subClassOf :CytologicalCharacters.

	:PloidyLevel rdf:type owl:Class ; rdfs:subClassOf :CytologicalCharacters.
:IdentifiedGenes rdf:type owl:Class ; rdfs:subClassOf :Evaluation.	-
:MolecularMarkers rdf:type owl:Class ; rdfs:subClassOf :Evaluation.	-
:PlantDescriptors rdf:type owl:Class ; rdfs:subClassOf :Evaluation .	:ChillingRequirementsOfVegetativeBuds rdf:type owl:Class ; rdfs:subClassOf :PlantDescriptors. :Kernel rdf:type owl:Class ; rdfs:subClassOf :PlantDescriptors. :Pollen rdf:type owl:Class ; rdfs:subClassOf :PlantDescriptors. :Yield rdf:type owl:Class ; rdfs:subClassOf :PlantDescriptors.

The "*:HighTemperature*" class which is the sub-class of "*:AbioticStressSusceptibility*" has three sub-classes named "*:SunburnSusceptibilityOfHusk*", "*:SunburnSusceptibilityOfLeaves*", and "*:SunburnSusceptibilityOfTrunk*" respectively. Another sub-class of "*:AbioticStressSusceptibility*" is "*:LowTemperature*". It has two sub-classes named "*:SusceptibilityToFrostDamageInSpring*" and "*:WinterHardiness*". The "*:Kernel*", "*:Pollen*", and "*:Yield*" classes are the sub-classes of "*:PlantDescriptors*". The "*:Pollen*" class has five sub-classes such as ":NormalPollen", ": PollenFertility", "*:PollenIncompatibilityAllelesFormula*", "*:PollenVitality*", and "*:RatioOfNormalAbortedPollenGrains*". The ":Yield" class has three sub-classes similarly. They are "*:AlternateBearing*", "*:CroppingEfficiency*", and "*:EstimatedYield*" respectively. The "*:Kernel*" class consists of two main sub-classes named "*:ChemicalComposition*" and "*:StorageQuality*". The sub-class "*:ChemicalComposition*" has following sub-classes: "*:KernelAshContent*", "*:KernelOilContent*", "*:KernelProteinContent*", "*:RatioBetweenUnsaturatedAndSaturatedFattyAcids*", and *":SolubleSugars*". The class "*:StorageQuality*" has five sub-classes named "*:KernelBitterness*", "*:KernelCrispness*", "*:KernelFirmness*", "*:KernelRancidityPotential*", and "*:KernelSweetness*".

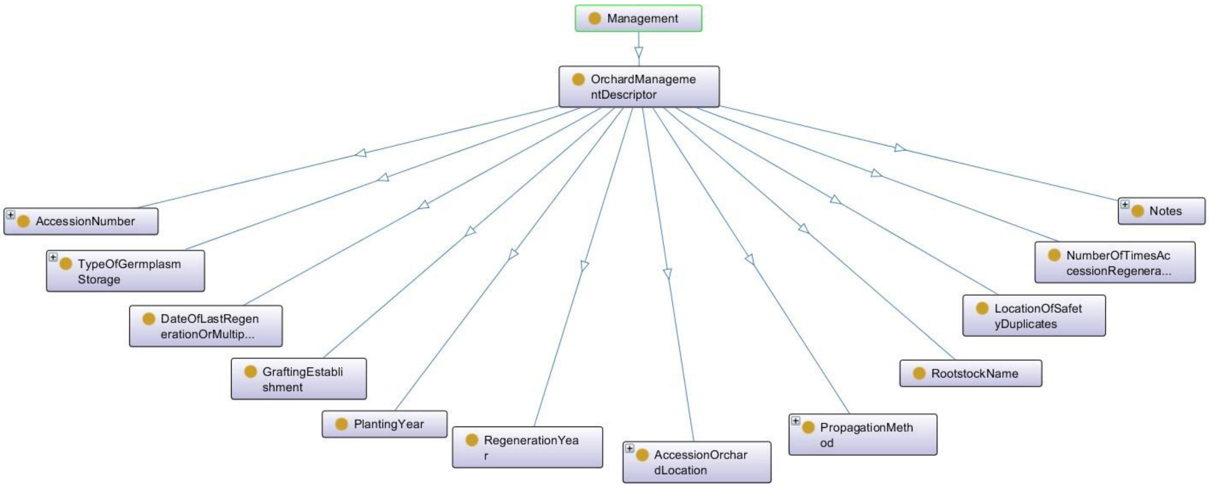


Figure 4.6 The sub-classes of ":OrchardManagementDescriptor" Class

As shown in Figure 4.6 the "*Management*" class consists of only one sub-class named "*OrchardManagementDescriptor*". The definitions of sub-classes of ":*OrchardManagementDescriptor*" class are set out in Table 4.7.

Table 4.7 The definitions of sub-classes of ":OrchardManagementDescriptor" Class

:AccessionNumber rdf:type owl:Class ; rdfs:subClassOf :OrchardManagementDescriptor.	:LocalPlantNumber rdf:type owl:Class ; rdfs:subClassOf :AccessionNumber.
:AccessionOrchardLocation rdf:type owl:Class ; rdfs:subClassOf :OrchardManagementDescriptor.	:BlockDesignation rdf:type owl:Class ; rdfs:subClassOf :AccessionOrchar dLocation . :RowNumber rdf:type owl:Class ; rdfs:subClassOf :AccessionOrchar dLocation . :TreeNumberWithinTheRow rdf:type o wl:Class ; rdfs:subClassOf :AccessionOrchard Location .
:DateOfLastRegenerationOrMultiplication rdf:type owl:Cl ass ; rdfs:subClassOf :OrchardManagementDescriptor.	-
:GraftingEstablishment rdf:type owl:Class ; rdfs:subClassOf :OrchardManagementDescriptor.	-
:LocationOfSafetyDuplicates rdf:type owl:Class ; rdfs:subClassOf :OrchardManagementDescriptor.	-
:Notes rdf:type owl:Class ; rdfs:subClassOf :OrchardManagementDescriptor.	-
:NumberOfTimesAccessionRegenerated rdf:type owl:Clas s ; rdfs:subClassOf :OrchardManagementDescriptor.	-
:PlantingYear rdf:type owl:Class ; rdfs:subClassOf :OrchardManagementDescriptor.	-

:PropagationMethod rdf:type owl:Class ; rdfs:subClassOf :OrchardManagementDescriptor.	-
:RegenerationYear rdf:type owl:Class ; rdfs:subClassOf :OrchardManagementDescriptor.	-
:RootstockName rdf:type owl:Class ; rdfs:subClassOf :OrchardManagementDescriptor.	-
:TypeOfGermplasmStorage rdf:type owl:Class ; rdfs:subClassOf :OrchardManagementDescriptor.	-

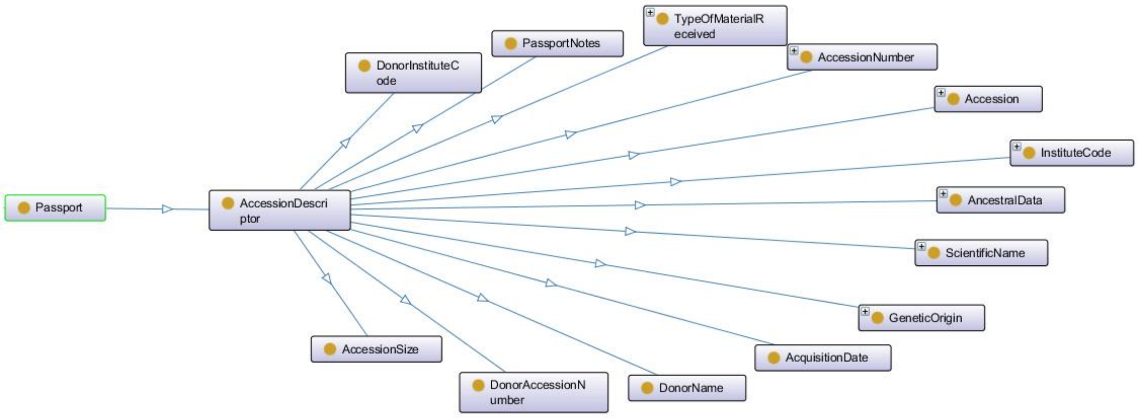


Figure 4.7 The sub-classes of ":AccessionDescriptor" Class

The descriptor named "*Passport*", which is the sub-class of "*Descriptor*" consists of two main descriptors named "*AccessionDescriptor*" and "*CollectingDescriptor*".

As shown in Figure 4.7 the "*AccessionDescriptor*" class has thirteen sub-classes. The sub-classes of "*AccessionDescriptor*" class are represented in Table 4.8.

Table 4.8 The definitions of sub-classes of ":AccessionDescriptor" Class

:Accession rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor.	:AccessionName rdf:type owl:Class ; rdfs:subClassOf :Accession. :CommonCropName rdf:type owl:Class ; rdfs:subClassOf :Accession. :LocalLanguage rdf:type owl:Class ; rdfs:subClassOf :Accession. :Synonyms rdf:type owl:Class ; rdfs:subClassOf :Accession. :TranslationOrTransliteration rdf:type owl: Class ;

	rdfs:subClassOf :Accession. :YearOfReleaseOfAccession rdf:type owl:Class ; rdfs:subClassOf :Accession .
:AccessionNumber rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor.	:LocalPlantNumber rdf:type owl:Class ; rdfs:subClassOf :AccessionNumber.
:AccessionSize rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor.	-
:AcquisitionDate rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor.	-
:AncestralData rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor.	:FemaleParent rdf:type owl:Class ; rdfs:subClassOf :AncestralData . :MaleParent rdf:type owl:Class ; rdfs:subClassOf :AncestralData .
:DonorAccessionNumber rdf:type owl:Class; rdfs:subClassOf :AccessionDescriptor.	-
:DonorInstituteCode rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor.	-
:DonorName rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor.	-
:GeneticOrigin rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor.	-
:InstituteCode rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor.	:CuratorName rdf:type owl:Class ; rdfs:subClassOf :InstituteCode. :SiteWhereMaintained rdf:type owl:Class ; rdfs:subClassOf :InstituteCode .
:PassportNotes rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor .	-
:ScientificName rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor .	:Genus rdf:type owl:Class ; rdfs:subClassOf :ScientificName. :SpeciesAuthority rdf:type owl:Class ; rdfs:subClassOf :Species. :Subtaxa rdf:type owl:Class ; rdfs:subClassOf :ScientificName. :TypeOfMaterialReceived rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor.
:TypeOfMaterialReceived rdf:type owl:Class ; rdfs:subClassOf :AccessionDescriptor .	

The “*:Species*” class which is the sub-class of “*:ScientificAuthority*” has a sub-class named “*:SpeciesAuthority*”. Furthermore, the class named “:Subtaxa” has following sub-classes: “*:RankName*”, “*:SubtaxonAuthority*”, and “*:SubtaxonName*”.

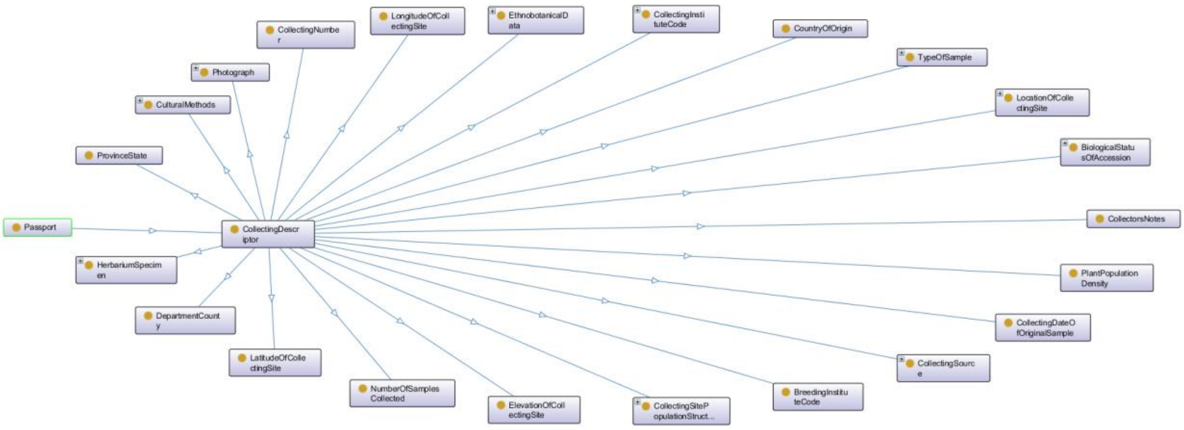


Figure 4.8 The sub-classes of ":CollectingDescriptor" Class

Figure 4.8 provides the “*AccessionDescriptor*” class and its sub-classes. In addition, the sub-classes of “*AccessionDescriptor*” class are represented in Table 4.9.

Table 4.9 The definitions of sub-classes of ":CollectingDescriptor " Class

:BiologicalStatusOfAccession rdf:type owl:Cl ass ; rdfs:subClassOf :CollectingDescripto r.	-
:BreedingInstituteCode rdf:type owl:Class ; rdfs:subClassOf :CollectingDescripto r.	-
:CollectingDateOfOriginalSample rdf:type ow l:Class ; rdfs:subClassOf :CollectingDescriptor.	-
:CollectingInstituteCode rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	:SiteNumber rdf:type owl:Class ; rdfs:subClassOf :CollectingInstituteCode.
:CollectingNumber rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	-
:CollectingSitePopulationStructure rdf:type o wl:Class ; rdfs:subClassOf :CollectingDescriptor .	:AssociatedFlora rdf:type owl:Class ; rdfs:subClassOf :CollectingSitePopulationStruct ure. :AssociatedMycorrhizalFungi rdf:type owl:Class ; rdfs:subClassOf :CollectingSitePopulationStruct ure. :FrequencyOfTheSpeciesAtCollectingSite rdf:type o wl:Class ;

	rdfs:subClassOf :CollectingSitePopulationStructure. :NumberOfTreesSampled rdf:type owl:Class ; rdfs:subClassOf :CollectingSitePopulationStructure .
:CollectingSource rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor .	-
:CollectorsNotes rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	-
:CountryOfOrigin rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	-
:CulturalMethods rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	:CroppingSystem rdf:type owl:Class; rdfs:subClassOf :CulturalMethods. :Irrigation rdf:type owl:Class; rdfs:subClassOf :CulturalMethods. :PropagationMethod rdf:type owl:Class; rdfs:subClassOf :CulturalMethods.
:DepartmentCounty rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	-
:ElevationOfCollectingSite rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor .	-
:EthnobotanicalData rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	:EthnicGroup rdf:type owl:Class ; rdfs:subClassOf :EthnobotanicalData. :LocalVernacularName rdf:type owl:Class ; rdfs:subClassOf :EthnobotanicalData. :PrevailingStresses rdf:type owl:Class ; rdfs:subClassOf :EthnobotanicalData. :UseOfSamplesCollected rdf:type owl:Class ; rdfs:subClassOf :EthnobotanicalData .
:HerbariumSpecimen rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	:HerbariumName rdf:type owl:Class ; rdfs:subClassOf :HerbariumSpecimen. :SpecimenIdentificationNumber rdf:type owl:Class; rdfs:subClassOf :HerbariumSpecimen.
:LatitudeOfCollectingSite rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	-
:LocationOfCollectingSite rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	:DirectionFromNearestPlace rdf:type owl:Class ; rdfs:subClassOf :LocationOfCollectingSite. :DistanceInKm rdf:type owl:Class ; rdfs:subClassOf :LocationOfCollectingSite. :NearestNamedPlace rdf:type owl:Class ; rdfs:subClassOf :LocationOfCollectingSite.
:LongitudeOfCollectingSite rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	-
:NumberOfSamplesCollected rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	-

:Photograph rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	-
:PlantPopulationDensity rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	-
:ProvinceState rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	-
:TypeOfSample rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor.	-

The "*:LocalVernacularName*" class has two sub-classes named "*:LanguageOfLocalVernacularName*" and "*:MeaningOfLocalVernacularName*".

Hazelnut Trait Dictionary consists of forty-five object properties. They make links between individuals which are the objects of the relevant domain. In other words, individuals are the instances of the classes within the ontology. Hazelnut Trait Dictionary has 371 individuals which have been created to represent the relevant hazelnut traits. The object properties which provide to make relationships for these individuals are described as follows:

- *":chooseOfTreeCondition*" is used to create existential restrictions that means "*owl:someValuesFrom*" between "*:ConditionOfTree*" and the following individuals "*:Dying*", "*:MatureDiseased*", "*:MatureNonVigorous*", "*:OldDeclining*", "*:YoungBearing*", and "*:YoungNotBearing*".
- *":hasAnswer"* is used to provide appropriate answers for the traits which may be represented with the "*:No*" and "*:Yes*" individuals.
- "*:hasBiologicalStatusOfAccession*" is defined to specify the biological status of the accession. It is used to create relationships for ":AdvancedImprovedCultivar", ":BreedingResearchMaterialBreedersline", ":Weedy", ":Wild", ":BreedingResearchMaterialMutantGeneticStock", and ":TraditionalCultivarLandrace" individuals.
- "*:hasBranchingDensity*" object property creates relationships for "*:Dense*", "*:Intermediate*", and "*:Sparse*" individuals.
- "*:hasBudColour*" object property creates relationships for "*:BrownGreen*", "*:Green*", and "*:Reddish*" individuals.
- :hasBudShapeType object property is used to make relationships for ":ConicalPointed", ":Globular", and " :Ovoid" individuals.

- :hasCollectingSource object property is used to construct relationships for ":FarmOrCultivatedHabitatFallowLand", ":FarmOrCultivatedHabitatField", ":FarmOrCultivatedHabitatGarden", ":FarmOrCultivatedHabitatOrchard", ":FarmOrCultivatedHabitatPasture", ":FarmOrCultivatedHabitatStore", ":InstituteResearchOrganizationExperimentalStationGenebank", ":MarketOrShopOtherExchangeSystem", ":MarketOrShopTown", ":MarketOrShopUrbanArea", ":MarketOrShopVillage", ":SeedCompany", ":WeedyDisturbedOrRuderalHabitat", ":WildHabitatDesertTundra", ":WildHabitatForestWoodland", ":WildHabitatGrasslands", and ":WildHabitatShrubland".
- :hasCroppingSystem object property creates relationships for ":Intercropping", ":Monoculture", ":NaturalCropping", and ":Agropastoralism" individuals.
- :hasFrequencyOfTheSpeciesAtCollectingSite object property creates relationships for ":Abundant", ":Frequent ", ":Occasional", ":Rare", and ":VeryAbundant" individuals.
- :hasGeneticOrigin object property creates relationships for ":OpenPollination", ":ArtificialPollination", and ":ClonalSelection" individuals.
- :hasGroundwaterQuality object property is used to make relationships for ":Fresh", ":Oxygenated", ":Polluted", ":Saline", ":Stagnating", and ":Brackish" individuals.
- :hasHigherLevelLandform object property is used to create relationships for ":Hill", ":Mountain", ":Plain", ":Plateau", ":Upland", ":Valley", and ":Basin" individuals.
- :hasIrrigation object property is used to create relationships for ":Rainfed", ":RiverBanks", ":Run-off", and ":Irrigated" individuals.
- :hasLandElementAndPosition object property is used to create relationships for ":RoundedSummit", ":Summit", ":Terrace", ":UpperSlope", ":Valley", ":ValleyFloor", ":Beach", ":BeachRidge", ":Caldera", ":Channel", ":ClosedDepression", ":CoralAtoll", ":CoralReef", ":DrainageLine", ":Dune", ":Escarpment", ":InterdunalDepression", ":Interfluve", ":Lagoon", ":Levee", ":LongitudinalDune", ":LowerSlope", ":Mangrove", ":Mid-slope",

":OpenDepression", ":Pan", ":PlainLevel", ":Ridge", and ":Floodplain" individuals.

- :hasLeafBladeShape object property is used to create relationships for ":Ovate", ":Rounded", and ":Elliptic" individuals.
- :hasLeafColour object property is used to create relationships for ":DarkGreen", ":Green", ":LightGreen", ":Red", ":RedGreen", and ":Yellow" individuals.
- :hasLightLevel object property is used to create relationships for ":Shady" and ":Sunny" individuals.
- :hasMatrixColour object property is used to create relationships for only ":Black" individual.
- :hasMethodUsedToProduce object property is used to create relationships for :Grafted, :RootedCutting, :Seed, :SuckerOrLayer, and :TissueCulture individuals.
- :hasMonthlyMean is used to specify monthly mean of temperature and rainfall.
- :hasShootDensityOfLenticels object property is used to create relationships for ":Low", ":High", and ":VeryHigh" individuals.
- :hasShootHairiness object property is used to create relationships for ":Weak", ":Medium", ":Strong" individuals.
- :hasShootThickness object property is used to create relationships for ":Thin", ":Medium", and ":Thick" individuals.
- :hasSoilDrainage object property is used to create relationships for ":PoorlyDrained", ":ModeratelyDrained", and ":WellDrained" individuals.
- :hasSoilErosionLevel object property is used to create relationships for "":Low, ":Intermediate", and ":High" individuals.
- :hasSoilFertilityLevel object property is used to create relationships for "":Low, ":Moderate", and ":High" individuals.
- :hasSoilMoisture object property is used to create relationships for ":Yellow", ":YellowishBrown", ":YellowishRed", ":Wet", "":White, ":SlightlyMoist", ":Reddish", ":ReddishBrown", ":ReddishYellow", ":Red", ":Moist", ":GreenishGreen", ":Grey", ":Greyish", ":Dry", ":Brown", ":Brownish", ":BluishBlack", and ":Blue" individuals.
- :hasSoilOrganicMatterContent object property is used to create relationships for ":Peaty", ":Nil", ":Low", ":Medium", and ":High individuals".

- :hasSoilParticalSizeClass object property is used to create relationships for ":Clay", ":CoarseSand", ":CoarseSilt", ":FineSand", ":FineSilt", ":MediumSand", ":VeryCoarseSand", and ":VeryFineSand" individuals.
- :hasSoilTextureClasses object property is used to create relationships for ":Clay", ":ClayLoam", ":CoarseSand", ":CoarseSandyLoam", ":FineSand", ":FineSandyLoam", ":Loam", ":LoamyCoarseSand", ":LoamyFineSand", ":LoamySand, ":LoamyVeryFineSand", ":MediumSand", ":SandUnsorted", ":SandUnspecified", ":SandyClay", ":SandyClayLoam", ":SandyLoam", ":Silt", ":SiltClay", ":SiltClayLoam", ":SiltLoam", and ":VeryFineSand" individuals.
- :hasStoninessRockinessHardpanCementation object property is used to create relationships for ":TillageUnaffected", ":TillageImpossible", ":TillageDifficult", ":TillageAffected", and ":EssentiallyPaved" individuals.
- :hasSuckeringType object property is used to create relationships for ":Strong", ":Medium", ":Absent", ":Weak", and ":VeryStrong" individuals.
- :hasTopography object property is used to create relationships for ":AlmostFlat", ":Flat", ":GentlyUndulating", ":Hilly", ":Mountainous", ":Rolling", ":SteeplyDissected", and ":Undulating" individuals.
- :hasTrainingSystem object property is used to create relationships for ":Bush" and ":Tree" individuals.
- :hasTreeGrowthHabit object property is used to create relationships for ":VeryErect", ":Spreading", ":Semi-erect", ":Erect", ":Drooping", and ":Contorted" individuals.
- :hasTreeVigour object property is used to create relationships for ":Low", ":Intermediate", and ":High" individuals.
- :hasTypeOfGermplasmStorage object property is used to create relationships for ":InVitroCollection", ":FieldCollection", ":CryopreservedCollection", ":SeedCollectionShortTerm", ":SeedCollectionMediumTerm", ":SeedCollectionLongTerm", ":InVitroCollection", ":FieldCollection", and ":CryopreservedCollection" individuals.
- :hasTypeOfSample object property is used to create relationships for ":Pollen", ":Seeds", ":TissueCulture", and ":Vegetative" individuals.

- :hasWaterAvailability object property is used to create relationships for “:Flooded”, ”:Irrigated”, “:Rainfed”, “:RiverBanks”, and “:SeaCoast” individuals.
- :selectCropAgriculture object property is used to create relationships for “:AnnualFieldCropping”, “:PerennialFieldCropping”, and “:TreeAndShrubCropping” individuals.
- :surroundCollectingSite object property is used to create relationships for “:Woodland”, “:NormalAverageHeightShrub”, “:GrasslandHerbaceous”, “:ForblandHerbaceous”, “:DwarfShrub”, and “:ClosedForest” individuals.
- :useOfSampleCollected object property is used to create relationships for “:WoodTimber”, “:SeedlingRootstock”, “:Pollinator”, “:Ornamental”, “:NutProduction”, “:Medicinal”, and “:ClonalRootstock” individuals.

Data properties are used to make link between an individual and a scalar value which is represented as XML Datatypes in OWL ontologies. In Hazelnut Trait Ontology, there are many data properties such as “:hasDateValue”, “:hasDayLengthValue”, “:hasDissolvedSaltValue”, “:hasElectroConductivityValue”, “hasLeafLengthValue”, “hasLeafWidthValue”, “hasLengthValue”, “hasPercentageValue”, “hasRainfallValue”, “hasReferenceCultivars”, “hasTemperatureValue”, “hasTicknessValue”, “hasValue”, “hasValueOfSoilDepth”, “hasWeightValue”, “hasWidthValue”, and “hasWindVelocityValue”. The first data property “*:hasDateValue*” is used to make links any individual to “*xsd:dateTime*” XML Schema datatype value. The second data property “*:hasDayLengthValue*” is used to represent the length of a day as hours. It links the relevant individuals to “*xsd:double*” XML Schema datatype value within the Hazelnut Trait Ontology. The third data property “*:hasDissolvedSaltValue*” provides linking individuals of the “*:DissolvedSalts*” class to “*xsd:double*” XML Schema datatype value. The fourth data property “*:hasElectroConductivityValue*” is used to create restrictions on the individuals of the “*:Electro-conductivity*” class and it links them to “*xsd:int*” XML Schema datatype value. The “*:hasLeafLengthValue*” and “*:hasLeafWidthValue*” data properties provide to represent the leaf length and leaf width values of hazelnut trees as “*xsd:double*” datatype value within Hazelnut Trait Ontology. The “*:hasLengthValue*” and “*:hasWidthValue*” data properties are used to define kernel and nut length and kernel and nut width as “*xsd:double*” datatype value. The “*:hasPercentageValue*” data property provides to set each percentage values to relevant individuals. The "*:hasRainfallValue*" data property links rainfall which might be specified as annual or monthly mean to "*xsd:double*" datatype value. Some individuals in Hazelnut Trait Ontology have reference cultivars such as Ennis, Fertile de Coutard, and Negret etc. They are represented with the type of "*xsd:string*" within Hazelnut Trait Ontology. The "*:hasReferenceCultivars*" data property provides linking the relevant individuals to the relevant reference cultivars. The "*:hasTemperatureValue*" data property describe relationships between the relevant individuals and their values which are represented as "*xsd:double*" datatype value in Hazelnut Trait Ontology. The "*:hasThicknessValue*" data property provide to represent the kernel and nut thickness values in Hazelnut Trait Ontology as "*xsd:double*" datatype value. The "*:hasValue*" data property which is for general usage in the context of linking individuals to certain values is used to set some scalar values to the relevant individuals. Furthermore, it might be used to create restriction for individuals. The following example illustrates how to use the "*:hasValue*" data property in minimum-maximum cardinality restrictions:

SoilParticleSizeClasses and
(hasValue some (xsd:double[>= "1251.0"^^xsd:double] and
xsd:double[<= "2000.0"^^xsd:double]))

The "*:hasValueOfSoilDepth*" data property is used to provide setting the value of soil depth according to groundwater table. It is represented as "*xsd:double*" datatype value in Hazelnut Trait Ontology. The "*:hasWeightValue*" data property provides for setting the weight value of 100-Kernel in Hazelnut Trait Ontology as "*xsd:double*" datatype value. The last data property "*:hasWindVelocityValue*" is used to represent the annual maximum wind velocity value as "*xsd:double*" datatype value in Hazelnut Trait Ontology.

4.3.2 Evaluation of Hazelnut Trait Ontology

We begin by taking a closer look at frameworks, methodologies, and tools to evaluate the quality of Hazelnut Trait Ontology.

Andrew Burton-Jones et al. proposed a metrics suite to assess the quality of domain ontologies (Burton-Jones, 2005). Their metrics suite consists of four metrics to evaluate the ontology quality which is the overall metric of the suite. The first metric of the suite is *syntactic quality*. Syntactic quality has two attributes: lawfulness and richness. The second metric of the suite is *semantic quality* which has three attributes: interpretability, consistency, and clarity. The third metric is *pragmatic quality* which has three attributes: comprehensiveness, accuracy, and relevance. The last metric is social quality, and it has two attributes: authority and history.

The second quality model we focused on is based on the hierarchical model, which is one of the software quality model types. The model proposed by Hong Zhu et al. divided the quality attributes into three aspects: contents which have quality attributes with a focus on the content of the ontology; presentation that has quality attributes related to the way in which the ontology presents the domain knowledge; and usage that has quality attributes which manifest themselves when the ontology is used (Zhu et al., 2017).
Another quality model we examined is developed for assessing the quality of a biodiversity ontology. The model is an operationalization of the information quality (IQ) assessment framework which combines conceptual and empirical approaches to identify an IQ problem structure and the requirements for an information object and grounding IQ metrics (Stvilia, 2007). The relevant model has twelve dimensions (accuracy/validity, cohesiveness, complexity, semantic consistency, structural consistency, currency, redundancy, naturalness, precision/completeness, verifiability, volatility, and authority) and metrics related to these dimensions. Each of metrics has a type of cost as automatic or semiautomatic. OntoQA (Metric-based Ontology Quality Analysis), which is proposed by Samir Tartir et al., has metrics as schema and instance. The schema metrics (relationship richness, attribute richness, inheritance richness) are related to the designation of an ontology. Instance metrics, which is divided into two metrics; namely, knowledgebase metrics (class richness, average richness, cohesion) and class metrics (importance, fullness, inheritance richness, relationship richness, connectivity, readability), are related to placement of instance data and distribution of the data (Tartir et al., 2005). The method proposed by Gomez-Pérez prevents the inconsistency, incompleteness, and redundancy errors by using the criteria consistency, completeness,

conciseness, expandability, and sensitiveness (Gomez-Perez, 2001). Protégé which is a project at Stanford University is an important tool to create, visualize and query the ontologies (Musen, 2015). This tool is also used to evaluate the ontologies by summarizing the ontology metrics, which are categorized into metrics, class axioms, object property axioms, data property axioms and annotation axioms. OntoClean methodology validates the ontological adequacy of taxonomic relationships and uses the broad concepts regarding ontologies like essence, identity, and unity which characterize relevant aspects of the intended meaning of the properties, classes, and relations that build an ontology (Guarino and Welty, 2004). OntoMetric method based on a multilevel framework is also called a taxonomy of 160 characteristics. It provides a way to choose and compare existing ontologies. While investigating the appropriate ontology for relevant projects, it is necessary to emphasize that some viewpoints called dimensions should be considered. These dimensions include the content of the ontology and the organization of their contents; the language in which it is implemented; the methodology that has been followed to develop it; the software tools used to build and edit the ontology; and the costs when the ontology will be necessary in a certain project (Lozano-Tello and Gomez-Perez, 2004). AKTiveRank, which uses a variety of metrics such as class match, density, semantic similarity, and betweenness to evaluate the ontologies considering the strength of representation of the concepts, is an experimental system (Alani et al., 2006). Gangemi et al. present a method to evaluate and validate the ontologies using three dimensions (structural measures, functional measures, and usability-profiling measures), which are based on a metaontology called O2 complemented with an ontology of ontology validation called oQual (Gangemi et al., 2006). OQuaRE considers three concepts (evaluation support, evaluation process, and metrics), and five characteristics (reliability, operability, maintainability, compatibility, transferability and functional adequacy) while evaluating the ontologies (Duque-Ramos et al., 2011). OOPS! (OntOlogy Pitfall Scanner!) is a tool that identifies pitfalls within the ontologies (Poveda-Villalón et al., 2014). TOMM (Tool for Ontology Modularity Metrics) is a software tool that is used to apply structural criteria, logical criteria, relational criteria, information hiding criteria, and richness criteria to ontology modules (axiom abstraction, vocabulary abstraction, high-level abstraction, weighted abstraction, and feature expressiveness) created by NOMSA modularization tool (Khan, 2018).

As Hazelnut Trait Ontology is a typical instance of an agricultural ontology, it is a kind of trait dictionary because it reflects detailed descriptors with respect to hazelnut agricultural product in five different groups. The document “Descriptors for Hazelnut” that we used to create Hazelnut Trait Ontology is the only one, and the most comprehensive scientific study all over the world. Although this document provides much information with respect to the hazelnut agricultural product, the created ontology using this document should have been evaluated by using a variety of metrics. The quality of Hazelnut Trait Ontology has been evaluated in terms of quality with the scope of this research using three tools and one-question-based methodology. In accordance with McDaniel and Storey, who created a timeline of domain ontology evaluation initiatives, the first study for evaluating ontology quality started with Gomèz-Pèrez’s study named Initial Criteria; and the studies in this area continued with Bioinformatics created by Bodenreider (Mcdaniel and Storey, 2019). Protégé, OntoMetric, and TOMM tools, which cover many evaluation metrics, are used to evaluate Hazelnut Trait Ontology within the scope of this study. These tools are available and open access for ontology creators. Ontology creators could evaluate their ontologies in terms of completeness, adaptability, conciseness, consistency, and clarity by using a questioning method. This kind of evaluation could be performed using FOCA methodology, which provides questioning ontologies in terms of five different goals. Figure 4.9, representing the tools and methodologies, is used to evaluate Hazelnut Trait Ontology. The results calculated by using these tools and methodology are examined and discussed in this section of the thesis.

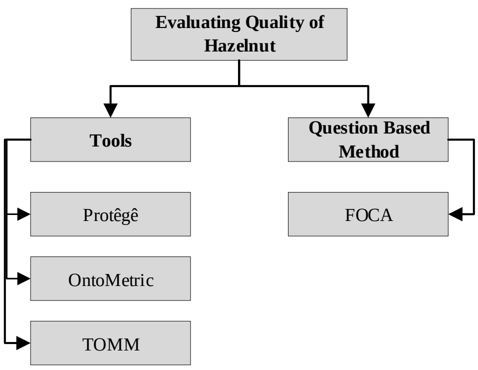


Figure 4.9 Tools and methods used to evaluate Hazelnut Trait Ontology

Protégé which is used to create Hazelnut Trait Ontology is the first tool we used to evaluate the quality of Hazelnut Trait Ontology as well. The method of Protégé for evaluating ontology is to indicate the count of the essential components of ontologies such as metrics, class axioms, object property axioms, individual axioms, and annotation axioms.

Table 4.10 Metrics and Axioms of Hazelnut Trait Ontology / Protégé

Metrics	Axiom	:	2255
	Logical axiom count	:	1406
	Declaration axioms count	:	740
	Class count	:	304
	Object property count	:	45
	Data property count	:	19
	Individual count	:	371
	Annotation Property count	:	3
Class axioms	Sub Class Of	:	378
	Disjoint Classes	:	6
Object property axioms	Sub object property of	:	2
	Functional object property	:	9
	Object property domain	:	1
	Object property range	:	1
Individual axioms	Class assertion	:	614
	Object property assertion	:	245
	Data property assertion	:	141
	Different Individuals	:	1
Annotation axioms	Annotation assertion	:	109

When one looks at Table 4.10, one can see that each metric (class axioms, object property axioms, individual axioms, and annotation axioms) have sub-metrics as well. The major components of OWL ontologies are axioms. Axioms as a metric in Protégé indicate the total count of logical and non-logical axioms. An OWL Ontology might have different kinds of axioms such as declarations, facts, keys, datatype definitions, and axioms concerning classes, objects, data properties, annotations. Class means a set of individuals known as instances in OWL Ontologies. Object properties specify the associations between two individuals. Data properties enable us to assign specific values to the individuals. Annotations are used to assign additional information regarding individuals, classes, object properties, and datatype properties. The class axioms consist of two different ones; sub class of and disjoint classes. There might be a hierarchical relationship between two classes in ontologies. For instance, Descriptor is the super-class in Hazelnut Trait Ontology. Evaluation is the sub-class of Descriptor. There are four different object property axioms in Protégé: sub object property of, functional object property, object property domain, and object property range. Object sub property axioms are similar to subclass axioms. The properties might be functional; and when a property is functional, that property is used to associate only one individual with another individual. Properties are used to associate objects from the domain to objects from the range. Individual axioms consist of class assertion, object property assertion, data property assertion, and different individuals. Class assertions provide a way to express which individual is an instance of which class. Object property assertions enable us to bind an individual with another individual by a specific circumstance. Data property assertions are expressions of connecting an individual to a literal.

The last metric is annotation axiom which has only one sub-metric such as annotation assertions. It is used to add meaningful, explanatory, and human-readable expressions and comments to individuals and IRIs. OntoMetric which is a web-based tool created by University of Rostock. It is the second tool we used to evaluate the Hazelnut Trait Ontology. The calculation results of OntoMetric tool in different categories for Hazelnut Trait Ontology are represented on Table 4.11, Table 4.12, Table 4.13, and Table 4.14. However, calculation results of class metrics have not been represented in this section of the thesis because Hazelnut Trait Ontology has 304 classes; and each class has nine calculation results. This means 2736 table of calculation results should take place in this section. So, these results were not included in this part of the thesis. OntoMetric tool evaluates an ontology by calculating diverse metrics for instance base, schema, class, knowledgebase and graph. Base metrics stand for the number of integral parts of ontology such as classes, axioms, object properties, individuals, etc. The calculation results of base metrics measured by OntoMetric tool are likewise with Protégé. That's why, it is appropriate to explain only schema, knowledge base, and graph metrics in this part of the thesis.

Table 4.11 Base metrics' calculation results / Onto Metric Tool

Base Metrics		
Axioms:	:	2255
Logical axioms count:	:	1406
Class count:	:	304
Total classes count:	:	304
Object property count:	:	45
Total object properties count:	:	45
Data property count:	:	19
Total data properties count:	:	19
Properties count:	:	64
Individual count:	:	371
Total individuals count:	:	371
DL expressivity:	:	ALCHOF(D)

Schema metrics should be taken into account while making decisions regarding how well ontology design models the domain knowledge. Schema metrics consist of a sort of metrics used to designate the richness, width, depth and inheritance of an ontology schema design. The most essential and significant metrics of schema category are the following: attribute richness, inheritance richness, relationship richness, attribute-class ratio, equivalence ratio, axiom class ratio, inverse relations ratio, and class relation ratio. Calculation results of schema metrics could be seen on Table 4.12. An ontology with many attributes depicts the relevant domain in an appropriate format. Attribute richness is calculated as the number attributes for all classes divided by the number of classes. It is a well-known fact that classes and their subclasses are essential components used for expressing the knowledge well within the ontologies. The metric inheritance richness is a way to indicate the grouping structure of classes within the ontology and identified as the average number of subclasses per class. Relationship richness examines the varieties of relations within the ontologies; and is calculated as the number of non-inheritance relationships, divided by the total number of relationships. Attribute-class ratio metric is used to specify the association between the classes that have attributes and all classes in the ontology. It is calculated as the number of classes containing attributes divided by the number of classes. Equivalence ratio enables us to calculate the rate between similar classes and all classes. It is computed as the number of the same classes divided by the number of all classes. Axiom class ratio specifies the ratio between axioms and classes; and is calculated as the number of axioms divided by the number of classes. Inverse relations ratio calculates the ratio between the inverse relations and all relations; and is computed as the summation of inverse object properties count and inverse functional data properties count divided by the summation of all object properties count and all functional data properties count. Class relation ratio calculates the ratio between classes and relations.

Table 4.12 Schema metrics' calculation results / Onto Metric Tool

Schema metrics		
Attribute richness:	:	0.0625
Inheritance richness:	:	1.243421
Relationship richness:	:	0.118881
Attribute class ratio:	:	0.0
Equivalence ratio:	:	0.0
Axiom/class ratio:	:	7.417763
Inverse relations ratio:	:	0.0
Class/relation ratio:	:	0.708625

Table 4.13 Knowledge base metrics' calculation results / Onto Metric Tool

Knowledgebase metrics		
Average population	:	1.220395
Class richness	:	0.3125

Ontology quality can be measured considering the data that took part within the ontology as it points to how well the ontology is designed, and how much the ontology represents the real-world. Average population and class richness are two different metrics calculated by OntoMetric tool. Average population is computed as the number of instances of the knowledge base divided by the number of classes defined in the ontology schema. Class richness enables comparisons between the counts of classes which have instances and total number of classes. It is computed as the percentage of the number of classes with instances divided by the total number of classes.

Table 4.14 Graph metrics' calculation results / Onto Metric Tool

Graph metrics		
Absolute root cardinality	:	1
Absolute leaf cardinality	:	236
Absolute sibling cardinality	:	304
Absolute depth	:	1480
Average depth	:	4.713376
Maximal depth	:	7
Absolute breadth	:	314
Average breadth	:	4.485714
Maximal breadth	:	37
Ratio of leaf fan-outness	:	0.776316
Ratio of sibling fan-outness	:	1.0
Tangledness	:	0.223684
Total number of paths	:	314
Average number of paths	:	44.857143

Graph metrics are used to calculate the structure of ontologies. As can be seen from Table 4.14, which represents the calculation results of the graph metrics, OntoMetric tool calculates seven diverse graph metrics such as cardinality, depth, breadth, fan-outness, tangledness, total number of paths, and average number of paths. Absolute root cardinality specifies the count of root nodes. The count of leaf nodes means absolute leaf cardinality and the number of sibling nodes is absolute sibling cardinality. The depth metric that consists of absolute, average, and maximal is associated with cardinality of the paths. The breadth metric is represented by three different metrics such as absolute, average, and maximal expresses the cardinality of levels. Ratio of leaf and ratio of siblings are two fan-outness (how graph nodes distribute) metrics calculated by OntoMetric tool. Tangledness is the measurement of the multi-hierarchical nodes in the graph. The total number of paths is the summation of distinct paths, which exist in the graph. They are placed between a root node and a leaf node. The metric average number of paths is computed as the total number of distinct paths divided by the number of graphs.

TOMM is the last tool we used to evaluate Hazelnut Trait Ontology. Table 4.15 represents the calculation results of TOMM. It computes similar metrics with other tools we used. However, it calculates three new metrics such as atomic size, appropriateness, and intra-module distance. Atomic size means average size of a group of interdependent axioms (Khan, 2018). Appropriateness is a way mapping module sizes to values which are between 0 and 1 by defining an appropriate function considering the defect density correlation (Schlicht and Stuckenschmidt, 2006). The intra-module distance means the number of relations in the shortest path between two entities (d'Aquin et al., 2009).

Table 4.15 TOMM's calculation results

Metrics for Hazelnut Trait Ontology		
No. of classes in module	:	304
No. of object property in module	:	45
No. of data property in module	:	19
No. of individual in module	:	371
Size of module	:	739
Atomic size of module	:	4.920162381596753
No. of axioms in module	:	2255
Appropriateness of module	:	-1.0
Intra-module distance	:	298562.0
Cohesion of module	:	0.062487717670379514
Attribute richness of module	:	0.34210526315789475
Inheritance richness of module	:	4.588235294117647

Attribute richness metric was computed by OntoMetric and TOMM tools. The following results were obtained from these tools: 0.0625, 0.34210526315789475, respectively. The distinction of these values is due to the formula used by OntoMetric and TOMM tools.

Although many tools use only the functional attributes to calculate attribute richness metric, OntoMetric uses all attributes declared in the ontology, and also handles the datatype as attributes. According to these calculation results, Hazelnut Trait Ontology has not got enough attributes (slots). From the results, it is apparent that Hazelnut Trait Ontology should be examined in terms of ontology design's quality and the quantities of information related to instance data. Another metric calculated by OntoMetric and TOMM tools is inheritance richness. This metric is calculated as 1.243421 by OntoMetric tool and also it is calculated as 4.588235294117647 by TOMM. This calculation result demonstrates that Hazelnut Trait Ontology is a typical deep, in other words, vertical ontology. A vertical ontology contains detailed information concerning a particular domain. Average population metric was calculated by only OntoMetric tool; and the result is 1.220395. This calculation shows us comparison between the count of individuals and the count of classes. The relevant calculation result could inform us regarding how well the instances represent the whole knowledge. The computed average population of Hazelnut Trait Ontology by both tools is low. This means that Hazelnut Trait Ontology does not have enough number of individuals per classes. On the one hand, it is important to emphasize some classes might have many instances. On the other hand, some of them might not have many instances within the ontology. As a consequence of that, this metric is not enough to express regarding the rank of quality of Hazelnut Trait Ontology. According to the value of class, richness of Hazelnut Trait Ontology has enough results in terms of class richness. The results of relationship richness show that classes in Hazelnut Trait Ontology are used by fewer numbers of instances. Attribute class ratio is measured as 0.0 by OntoMetric. This means that Hazelnut Trait Ontology has no class with attributes. Similarly, equivalence ratio has been measured as 0.0; and this expresses that Hazelnut Trait Ontology has no equivalent classes. The size of ontology has been measured by TOMM as 739. This value is the summation of the number of classes (304), object properties (45), data properties (19), and individuals (371) in Hazelnut Trait Ontology. Atomic size of Hazelnut Trait Ontology has been computed as 4.920162381596753 using TOMM. This calculation result specifies that 4.920162381596753 axioms of the ontology are grouped together in an atom. Optimal appropriateness metric's value is 1.0. The value of appropriateness metric computed by TOMM is 1.0 for Hazelnut Trait Ontology. When this calculation result is taken into consideration, it would be acceptable to indicate Hazelnut Trait Ontology as appropriate. The calculation result of the intra-module distance metric helps us specify the distance between entities within the ontology. This value is computed by TOMM as 298562.0 for Hazelnut Trait Ontology; and it indicates that the entities of Hazelnut Trait Ontology are close to each other. The cohesion metric is related to the relationship between entities and measured by TOMM as 0.062487717670379514 for Hazelnut Trait Ontology. As this value is relatively low, it can be expressed that the entities of Hazelnut Trait Ontology have less relationship with each other.

FOCA methodology based on Goal, Question, and Metric (GQM) consists of thirteen different questions which aim to compute significant metrics and belong to five essential goals (Bandeira et al., 2016). It enables evaluators to grade six metrics as follows: adaptability, clarity, completeness, computational efficiency, conciseness, and consistency. Afterwards, total ontology quality is computed using the beta regression models (Equation 1) considering the mean values of each goal. Some of these questions have sub questions, so to calculate the grade of the relevant question, the average of grades of sub questions should be calculated first. Table 4.16 demonstrates the grades that were given by the ontology evaluator for each question. It is not convenient to answer some of these questions due to the type of ontology. According to FOCA methodology, if evaluating ontology is a kind of domain or task ontology, Question 4 (Q4) should not be answered by the evaluators. So, Q4 was not verified for Hazelnut Trait Ontology because it is a domain ontology. FOCA methodology recommends grading these questions using appropriate points like 25, 50, 75, and 100. At the end of the grading process, the evaluator should calculate the final grades for each question considering the sub questions. It should be noted that if a question has sub questions, the final score of it is the mean of sub questions' grades.

Table 4.16 FOCA Methodology's Quality Evaluation Results

Goal	Question	Metric	Sub Questions	Grade
1. Check if the ontology complies with Substitute.	Were the competency questions defined?	Completeness	Does the document define the ontology objective?	100
			Does the document define the ontology stakeholders?	100
			Does the document define the use of scenarios?	100
	Were the competency questions answered?	Completeness	-	100
	Did the ontology reuse other ontologies?	Adaptability	-	0
2. Check if the ontology complies with Ontological Commitments.	Did the ontology impose a minimal ontological commitment?	Conciseness	-	-
	Did the ontology impose a maximum ontological commitment?	Conciseness	-	100
	Are the ontology properties coherent with the domain?	Consistency	-	100
3. Check if the ontology complies with Intelligent Reasoning	Are there contradictory axioms?	Consistency	-	100
	Are there redundant axioms?	Conciseness	-	50
4. Check if the ontology complies with Efficient Computation	Does the reasoner bring modelling errors?	Computational efficiency	-	100
	Does the reasoner perform quickly?	Computational efficiency	-	50

5. Check if the ontology complies with Human Expression.	Is the documentation consistent with the modelling?	Clarity	Are the written terms in the documentation the same as the modelling?	100
			Does the documentation explain what each term is and does it justify each detail of modelling?	100
	Were the concepts well written?	Clarity	-	100
	Are there annotations in the ontology bringing the concepts definitions?	Clarity	-	25

Table 4.17 represents final grades of the questions. According to these results, one can make the following inferences:

- Hazelnut Trait Ontology is adequate in terms of completeness because the scores of entire completeness' metric questions are 100,
- Hazelnut Trait Ontology is poor in terms of adaptability because it does not reuse any ontology,
- The conciseness and computational efficiency scores of Hazelnut Trait Ontology is over average
- It could be expressed that Hazelnut Trait Ontology is consistent, considering the grades of consistency questions
- The score of clarity metric for Hazelnut Trait Ontology is 81.25. This score is sufficient. However, it could be enhanced by adding annotations in the ontology bringing the concepts definitions.

Table 4.17 Summary Table of Question's Grades

Question	Grade	Question	Grade
Q1	100	**Q8**	50
Q2	100	**Q9**	100
Q3	0	**Q10**	50
Q4	-	**Q11**	100
Q5	100	**Q12**	100
Q6	100	**Q13**	25
Q7	100	**-**	-

From Table 4.18, it is apparent that averaged point of second goal is the highest as each question belonged to this goal has 100 points. On the other hand, third and fourth goals have the same and the lowest averaged points of question because one of the questions regarding each of these goals have only 50 points. First and the last goals have almost the same average value, too. The results represented on this table demonstrate that Hazelnut Trait Ontology has attained each goal proposed by FOCA methodology.

At the end of the quality evaluation process for Hazelnut Trait Ontology, Equation 1 might thus be used to calculate the overall quality of it using average values of the relevant goals represented on Table 4.18.

Table 4.18 The Mean Value for Goals

Goal	Mean
1	80
2	100
3	75
4	75
5	81.25

Equation 1- The Formula for Calculating Overall Quality

$$\mu_i = \frac{exp\left\{-0.44+0.03(Cov_S\ x\ Sb)_i+0.02(Cov_C\ x\ Co)_i+0.01(Cov_R\ xRe)_i+0.02(\ Cov_{Cp}xCp)_i-0.66LExp_i-25(0.1xNl)_i\right\}}{1+exp\left\{-0.44+0.03(Cov_S\ x\ Sb)_i+0.02(Cov_C\ x\ Co)_i+0.01(Cov_R\ xRe)_i+0.02(\ Cov_{Cp}xCp)_i-0.66LExp-25(0.1xNl)_i\right\}}$$

(Cov_S average grades of goal 1, Cov_c average grades of goal 2, Cov_R average grades of goal 3, Cov_{Cp} average grades of goal 4,LExp experience variable of evaluator ,Nl is 0 if the evaluator is good at ontology assessment)
(If the evaluator would like to calculate the overall quality of the relevant ontology Sb, Co, Re, and Cp should be 1.)

Equation 2 shows the formula with real values of variables regarding Hazelnut Trait Ontology. After this calculation is carried out, the overall quality of Hazelnut Trait Ontology is obtained as 0.997994791.

Equation 2- Overall Quality Score of Hazelnut Trait Ontology

$$\mu = \frac{exp\{-0.44+0.03(80x1)+0.02(100x1)+0.01(75x1)+0.02(75x1)-0.66(0)-25(0.1x0)\}}{1+exp\{-0.44+0.03(80x1)+0.02(100x1)+0.01(75x1)+0.02(75x1)-0.66(0)-25(0.1x0)\}}$$

$$\mu = 0.997994791$$

4.4 Limitations

In this chapter of the thesis, a crop-specific trait ontology called "Hazelnut Trait Ontology" has been developed, introduced, and evaluated using existing ontology evaluation tools and methods. This ontology constructs the main pillar of the models and approaches proposed within this thesis. However, several important limitations to this study need to be considered. First, it may be a limitation that entire domain stakeholders may not come to a consensus on using this ontology for solving any problems of hazelnut production. So, it is essential to disseminate usage of Hazelnut Trait Ontology. Second, it has not been possible to include entire domain stakeholders into development process of the ontology. So, it has been a limitation in terms of making contributions by whole domain stakeholders to the ontology development process. Third, Hazelnut Trait Ontology has been evaluated only using existing evaluation tools and methods. However, it is quite important including many stakeholders into the evaluation process of the ontology in terms of enhancing its consistency, viability, and reliability.

5. DATA ACQUISITON USING ONTOLOGIES

5.1 Data Acquisition Using Ontologies

Ontologies are frequently used to specify the traits of an agricultural product. Crop Ontology (CO) project, which is a well-known agricultural trait dictionaries publisher, has thirty-four trait dictionaries for a variety of agricultural products such as banana, soybean, potato, cotton, rice, etc. We focused on hazelnut in this thesis, so Hazelnut Trait Ontology; in other words, hazelnut trait dictionary we created is used to generate data acquisition forms. We consider that while creating data acquisition forms to gather data related to an agricultural product from stakeholders of the relevant domain, these traits can be modeled as attributes of the data models. Trait dictionaries provide entire concepts of a domain and relationships between these concepts. It might be determined what kind of information to be collected using trait dictionaries in the context of agriculture. In doing so, any domain stakeholder might decide what type of form elements to be created for data acquisition forms by considering trait dictionaries in the agriculture domain. For instance, a stakeholder whose purpose is to collect data through wireless sensor networks (WSNs) might identify with ease which trait corresponds to sensor measurement value by examining the "*Environment and Site*" descriptors of hazelnut agricultural product. It enhances collecting certain domain-specific data and provides producing meaningful information concerning hazelnut. Since it shall prevent collecting irrelevant data in the context of hazelnut, it makes significant contributions for data processing. As data collected through data acquisition forms is generated using agricultural trait dictionaries and is represented by a class in trait dictionaries, data annotation might be performed with ease using such data. For instance, topography means to the profile in elevation of the land surface on a broad scale and is a kind of "*Environment and Site*" descriptor of hazelnut. It might be collected through data acquisition form. Considering that the data acquisition form is generated using hazelnut trait dictionary, it shall be easy to annotate the topography trait data using the following data restrictions within the hazelnut trait dictionary: range of 0-0.5% is "Flat", range of 0.6-2.9 is "Almost Flat" etc.

Thus, this part of the thesis seeks to address the following research questions:

(i). what is the appropriate generic model to collect domain-specific data regarding a particular agricultural product using the capabilities of semantic web technologies, contemporary software design patterns, and data storage options to meet data requirements of miscellaneous software applications;

(ii). how could web-based data acquisition forms be published automatically through open data platforms to collect more sophisticated and structured data, concerning an agricultural product, using trait dictionaries and supplementary ontologies, which were utilized as conceptual models to design user interfaces;

(iii). what is the usability of developed tools based on a proposed model from domain stakeholders' point of view?

There are several important points where this study makes an original contribution to use semantic web technologies in the agriculture domain.

First, we propose a generic ontology-based data acquisition model to create data acquisition forms based on MVC design pattern, to publish and make use of, on the agricultural open data platforms.

Second, we propose a web control ontology to match ontology classes to web controls.

Third, we describe a tool called OWL2MVC that integrates the Hazelnut Trait Ontology, which illustrates the effectiveness of the proposed model for generating data acquisition forms. OWL2MVC is a completely ontology driven tool to semantically enrich processes generating data acquisition forms. Therefore, new agricultural ontologies can be uploaded to generate responsive data acquisition web forms on an open data platform. Meanwhile, it is necessary here to specify that usage of OWL2MVC Tool does not require strong technical skills. Entire domain stakeholders could use this tool easily, effectively, and rapidly, in compliance with their objectives.

Lastly, the tool OWL2MVC was evaluated in terms of usability by fifty-three respondents implementing the case-study scenario. Among others, the findings show that the tool has satisfactory usability score overall and is promising to provide stakeholders with the required support for agricultural open data platforms.

The tool OWL2MVC, which is developed according to the proposed ontology-based data acquisition model within this study, aims to enable agriculture domain stakeholders collect domain-specific structured data by generating and publishing data acquisition forms via web-based data platforms. As previously mentioned, the most important feature of the tool OWL2MVC is to be developed for web-based open data platforms in the context of agriculture. As it is developed based on a contemporary design pattern named MVC, it builds robust technical infrastructure for agriculture stakeholders to achieve their requirements of domain-specific data acquisition by exploiting the capabilities of semantic web technologies. Hazelnut Trait Ontology which represents the metadata regarding hazelnut carries out the interoperability between heterogeneous agricultural data sources. It is used as a conceptual model of developed tool named OWL2MVC. The major goal of the tool OWL2MVC is to create ontology-based data acquisition forms published through agricultural open data platforms to eliminate data heterogeneity in the context of a particular agricultural product.

As previously mentioned, OWL2MVC is based on MVC, which is a design pattern widely used in different application development platforms such as mobile applications development, web applications development, or rich client applications development for desktops, laptops, and tablet PCs. MVC design pattern is not a new paradigm in object-oriented programming. It was conceived to produce design solutions to particular problems in terms of handling a large and complex data set by Trygve Reenskaug (Reenskaug, 1979) (Reenskaug, 2003). According to MVC design pattern, the model is responsible for taking over the operations with respect to application domain from objects of different classes; the view is responsible for displaying the application's state and visualizing the model; the controller manages the interaction between the model and the view (Krasner and Pope, 1988). Today, in particular, it plays critical roles much more in practices of web and mobile application development. As well as OWL2MVC is developed based on such a significant, contemporary, and widely used design pattern, it might use three data storage options such as MS SQL Server as relational database, RDF as file-based option, and Neo4j as graph database to meet different requirements for collected data through created forms. Graph databases have been developed since 1998. Today, they have an important role and are widely used instead of relational databases in the semantic web in terms of storing data as graphs. A graph might be defined as a group of vertices and edges. In other words, a graph means a set of nodes and relationships which interconnect them (Robinson, 2016). In relational databases, an entity is represented as a table. However, in graph databases, entities are defined as nodes. Graph database is a kind of database system that stores data as graphs and formalizes it as a graph-theoretic data model (Graves et al., 1995). A graph database is directly labeled multigraph, which means a pair of nodes might be connected by more than one edge (Amann and Scholl, 1996). According to Guting, a database consists of three kinds of object classes: simple classes, defined as the nodes of the database graph; link classes, defined as edges of the database graph; and path classes, which contain references to nodes and edges (Guting, 1994). Identically, the representation of data is performed using directed labeled graph whose nodes are the objects of the database in (Gyssens and Paredaens, 1990). On the other hand, database schemas and instances are represented by using labeled graphs (Hidders and Paredaens, 1993). Database options have advantages and disadvantages compared to each other. Considering the use of Object Relational Mapping (ORM) techniques, it is preferable to use a relational database in this study. On the other hand, graph database is also used due to compatibility with semantic web technologies and simplicity to store RDF datasets. Since RDF file-based option enables us to export form data in open formats, it is adopted as a storage option of the proposed model.

Thus far, ontology-based data acquisition form generation and model proposing have been addressed by some studies for various domains. Eriksson et al. introduced a development environment named Mecano, which is developed as a component of Protégé, and provides designing knowledge-acquisition software tools automatically (Eriksson et al., 1994). They illustrate a component of Mecano called DASH, which consists of two design steps of user interfaces: dialog design and layout design. They use

Table 5.1 Evaluating proposed models and developed tools

	Eriksson et al. (Eriksson et al., 1994)	Rubin et al. (Rubin et al., 2002)	Harth (Harth, 2004)	Tran et al. (Tran et al., 2011)	Zheng et al. (Zheng et al., 2011)	Girardi et al. (Girardi et al., 2012)	Chen et al. (Chen et al., 2013)	Gonçalves et al. (Gonçalves et al., 2015, 2017)	Vcelak et al. (Vcelak et al., 2017)	Proposed work
Does it propose a model?	√	√	√	X	X	√	√	√	√	√
Does it propose a tool (based on the proposed model)?	√	√	√	√	√	√	√	√	√	√
On which domain does the study focus?	–	Pharmacogenetics	–	Health	Biology	Health	–	Biomedicine	Medicine	Agriculture
Does it cover the data requirements of stakeholders in a particular domain?	–	X	X	√	√	X	–	X	√	√
Which data storage options are supported by the developed tool?	–	Relational	Relational	Relational	–	Relational	–	Graph	–	Relational, Graph, and File-based
Does it use any software design pattern(s)?	–	X	X	X	X	X	X	X	MVC	MVC
Does it create data acquisition forms for open data platform?	X	X	X	X	X	X	X	X	X	√
Is it appropriate for non-technical users	X	X	X	√	X	√	X	X	√	√

to create forms dynamically?										
Is usability of the developed tool measured using an appropriate method?	X	√ (Tested by implementing in a production system)	X	√ (Validation and evaluation are provided.)	X	X	X	X	X	√

a layout algorithm that derives from the domain ontology the interface components such as text fields, check boxes, etc. Their work has not aimed to generate ontology-based data acquisition forms to collect domain-specific data and to publish through open data platforms considering the data requirements of stakeholders within a particular domain. Rubin et al. propose a method that interfaces ontology models with data acquisition from an external data source; uses an interface between ontology and the data source; and the interface is modeled in the ontology and implemented using XML schema (Rubin et al., 2002). Their method focuses on pharmacogenetics. It has not covered the data requirements of stakeholders within the relevant domain. Similarly, its purpose of development is not to create data acquisition forms published by non-technical users or domain stakeholders via web-based open data platforms. Harth presents a system named SECO that enables human and software agents browsing and querying the data repositories by providing interfaces, which are a kind of HTML output, generated from RDF, using XML (Harth, 2004). In the study, it could not be identified whether the proposed system focuses on a particular domain or not. The proposed system in this study uses relational database. However, it has not been developed based on MVC or any other contemporary software design patterns. On the other hand, the aim of the system is not to generate data acquisition forms for web-based open data platforms by taking into consideration the non-technical system users. Tran et al. propose an ontology-driven and web-based framework named OnWARD, which is focused on health domain and used to generate user-centered dynamic forms for any clinical study (Tran et al., 2011). Nevertheless, their web-based framework does not aim to generate data acquisition forms for web-based open data platforms. In addition, a proposed generic ontology-based data acquisition model could not be encountered in their study. Zheng et al. illustrate how to design submission forms to submit standardized data, using ontology-based models by reducing the responsibility of end users in the biology domain (Zheng et al., 2011). Similarly, their work does not include a generic ontology-based data acquisition model. As their study does not utilize a software design pattern, it doesn't aim to generate dynamic data acquisition forms published through web-based open data platforms considering the non-technical domain stakeholders. Girardi et al. created an ontology-based data acquisition infrastructure and developed a generic web-based data acquisition system by using a particular meta-model in the health domain (Girardi et al., 2012). Their infrastructure does not use a software design pattern; and also its purpose is not to generate data acquisition forms for collecting domain-specific data through open data platforms. On the other hand, their work considers non-technical users. Chen et al. propose a modelling language to map web forms of back-end databases to ontologies (Chen et al., 2013). The focused domain could not be determined in this work. In addition, it is not highlighted whether this work considers the data requirement of stakeholders within a particular domain or not. Furthermore, it does not aim to create data acquisition forms through web-based open data platforms considering the non-technical domain stakeholders. Gonçalves et al. illustrate a software system for the biomedicine domain, which works with forms modelled in OWL, and is based on an information model, domain ontologies for formal descriptions of entities, and a data model (Gonçalves et al., 2015)(Gonçalves et al., 2017). Their system uses a graph database. However, it is not developed based on MVC or another different contemporary software design pattern. Furthermore, their system does not consider non-technical domain stakeholders and does not aim to run on open data platforms as well. Ungkawa et al. propose a system regarding the reuse of design with a case-based reasoning approach to generate HTML form design automatically (Ungkawa et al., 2017). The last study concerning ontology-based web forms generation has been proposed in medicine domain by Vcelak et al. They propose an approach to generate ontology-based web forms automatically; and this approach allows editing generated forms when the relevant ontology changed (Vcelak et al., 2017). However, this work does not aim to generate data acquisition forms published through open data platforms.

Table 5.1 summarizes and evaluates nine representative studies from literature in terms of eight criteria. The first criterion assesses each work in terms of whether it proposes a model or not. Seven studies propose a model, but two studies do not. The second criterion examines each work according to tool development. While deciding to create this criterion, we did not only take into account whether the tool was developed based on the proposed model. Besides this, we examined whether a tool was developed or not within the relevant study. The third criterion is related to the domain focusing on the proposed models and developed tools. Most of them are proposed and developed for a similar or the same domain. However, the proposed work focused on agriculture, which is a vital sector due to the contribution to employment, exportation, and domestic income, providing raw materials sources for industry. The fourth criterion focuses on the requirements of stakeholders within the relevant domain. The proposed model and tool consider the requirements of hazelnut domain's stakeholders such as researchers, analysts, farmers, and experts. The fifth criterion evaluates the studies in terms of data storage options. The proposed work uses both relational and graph databases. Besides these, it enables us to store data in RDF files. Only one of the studies uses a design pattern while developing their tool. However, the proposed tool was developed, completely based on the MVC design pattern. None of the studies aims to publish data acquisition forms on an open data platform. The purpose of the proposed work is to dynamically create data acquisition forms for agricultural open data platform. The last criterion is about the usability of developed tools. Only three of these tools are appropriate for non-technical users. OWL2MVC Tool provides very easy UI so that any stakeholder of a relevant domain might create responsive and dynamic web forms, which are adaptable to open data platform based on modern solutions.

As can be seen from the table, almost none of these studies includes usability of the developed tool or approach. Furthermore, none of these works aims to generate ontology-based data acquisition forms published through web-based open data platforms considering the non-technical agriculture domain stakeholders. On the other hand, it is worth noticing that they provide a limited support for three different data storage options together considering the data requirements of domain stakeholders.

The tool OWL2MVC comes into prominence with the following capabilities and its robust characteristics:

(i). it is developed based on a generic ontology-based data acquisition model, which is proposed and introduced within this study;

(ii). it utilizes a contemporary software design pattern named MVC;

(iii). it supports three data storage options such as relational databases, graph databases, and different files based on open standards;

(iv). it considers non-technical agriculture domain stakeholders in terms of generating data acquisition forms published on open data platforms;

(v). it meets data requirements of agriculture domain stakeholders;

(vi). it is evaluated and tested with satisfactory usability score by fifty-three respondents.

In the following section, we shall introduce an ontology-driven acquisition data model and the tool OWL2MVC that exhibits all these characteristics.

5.2 Ontology-based Data Acquisition Model

This section of the thesis introduces the proposed ontology-based data acquisition model for agricultural ontologies. Figure 5.1 represents this model. There are not any restrictions in terms of usage of programming languages, IDEs, and frameworks to implement this model. It has been proposed as a generic model to create web-based data acquisition forms using MVC design pattern. So, any stakeholders of any agricultural domain could use this model to create web-based data acquisition forms using this model. In this part of the thesis, only the proposed model shall be introduced in general. Nevertheless, deep technical details will not be mentioned, and they shall be discussed within the scope of further parts.

The proposed model consists of four different parts: uploading ontology to open data platform; selecting-listing and setting ontology classes; creating model-view-controller;

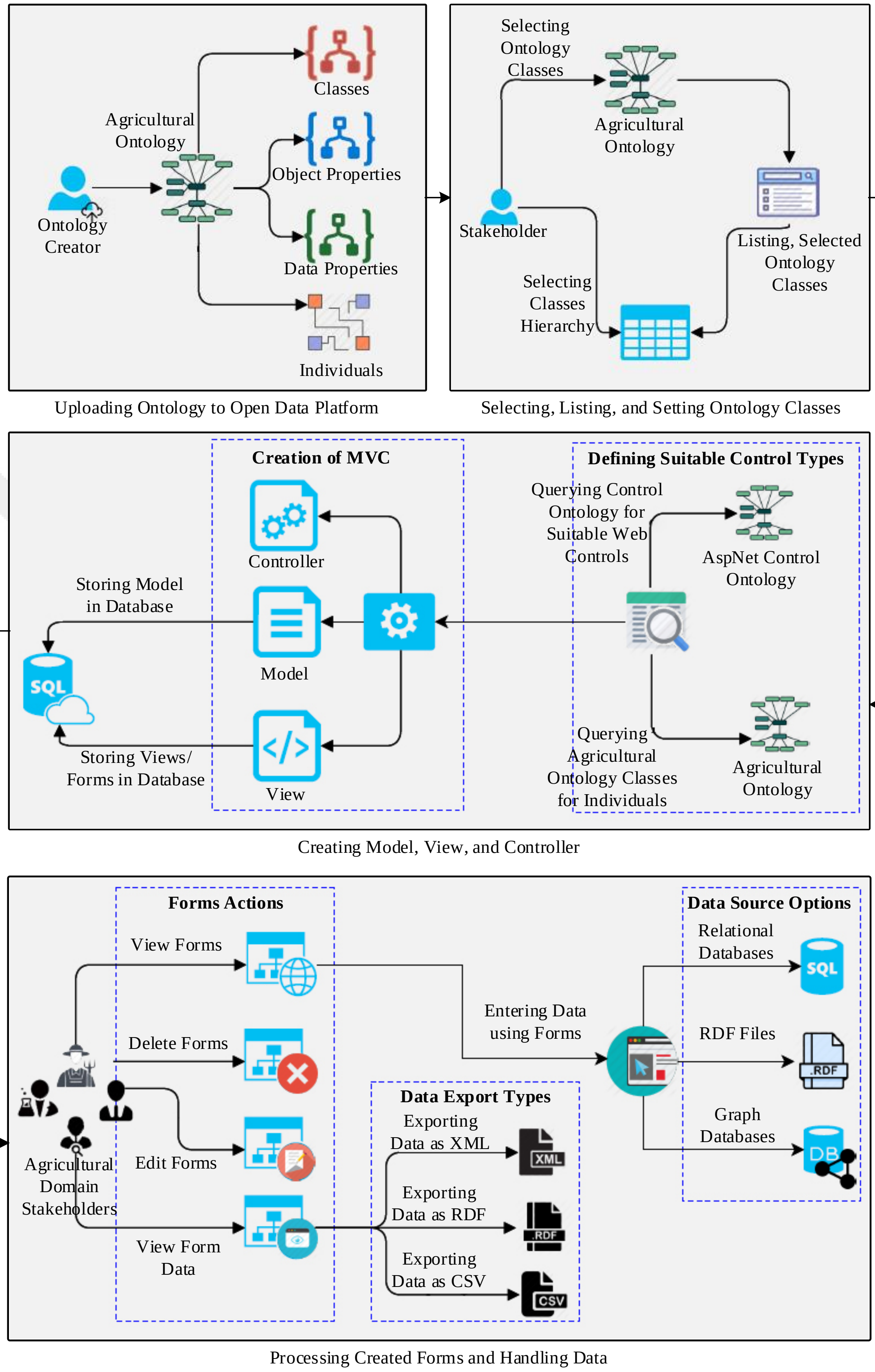


Figure 5.1 Ontology-based Data Acquisition Model

and processing created forms and handling data. Each of these parts includes a variety of actions and processes. The first part of the model is “Uploading Ontology to Open Data Platform.” The ontology creator should have uploaded the relevant agricultural ontology on the open data platform to extract the ontology’s essential elements, which are classes, object properties, data properties and individuals. If we needed to explain exactly what is meant by the agricultural ontology within the scope of this model, it could be defined as a kind of “phenotype and trait dictionary (or crop ontology).” An agricultural trait dictionary includes a variety of descriptors with respect to phenotype, breeding, germplasm, pedigree and traits in one or more categories regarding any agricultural product. These types of ontologies have been generally created and published in the Open Biomedical Ontologies (OBO) format. Contrary to general usage of OBO format for agricultural trait dictionaries, the proposed model uses an OWL ontology to create data acquisition forms.

The first part of the proposed model is “Uploading Ontology to Open Data Platform.” According to this part of the proposed model, any agricultural ontology creator uploads the ontologies on the platform to view it in a tree view structure with its all essential elements such as classes, object properties, data properties and individuals. The platform that uses the proposed model has to list the general ontology metrics such as class count, object property count, data property count, and individual count of the relevant ontology. These tree-view structured lists are the most appropriate way to introduce the relevant ontology as open data platform users could see all the elements of ontologies explicitly by using them. In addition, these lists enable us to view the general structure of the ontologies and also give an idea regarding which classes are favorable for creating data acquisition forms.

The second part of the proposed model is “Selecting, Listing and Setting Ontology Classes”. According to this part of the model, any stakeholders such as domain experts, researchers, and analysts should view the ontology tree structure in expanded form. This tree view structure should show which classes are suitable to use as a form element and allow us to select and list them within a summary table of selected ontology classes. Form elements are created using the names of ontology classes. However, the form creators might want to change the name of the form element. So, this table should enable the form creator (or stakeholder) to change the display name of the form elements by using the super class – sub class relationship of the ontology. With the aforementioned reasons, the summary table of selected ontology classes should consist of two columns. The first column should list the selected classes, which are used to create form elements; and the second column should show all their direct super classes in a selectable format to set the class hierarchy.

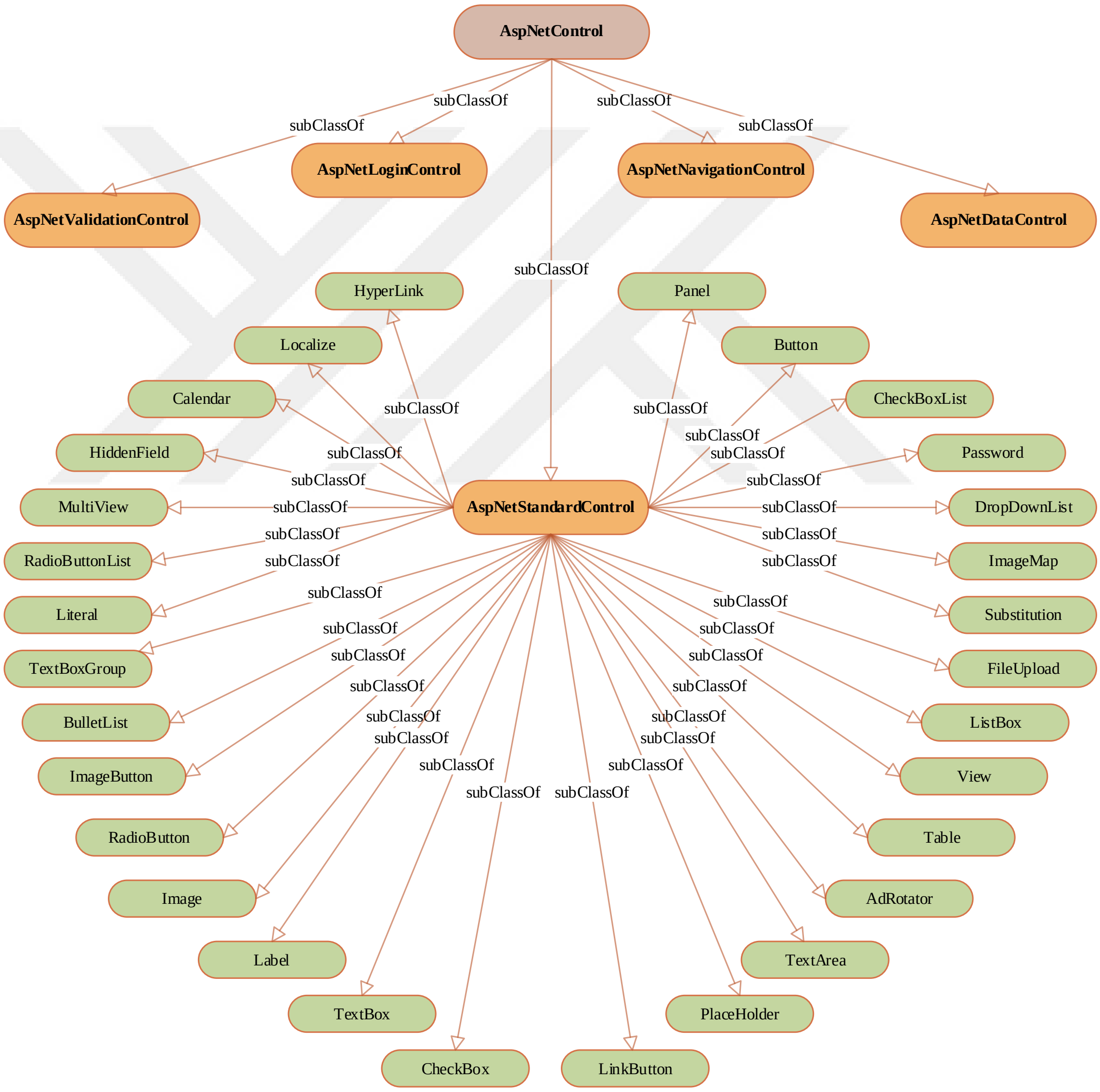


Figure 5.2 ASPNET Control Ontology

The third part of the proposed ontology-based data acquisition model is “Creating MVC”. This part consists of two layers, which are “Defining Suitable Control Types” and “Creation of MVC,” respectively. The first layer enables us to make queries on the web controls ontology, which is represented as “ASPNET Control Ontology” within the scope of our model to match ontology classes to web controls. “ASPNET Control Ontology,” depicted in Figure 5.2, is created to represent the general vocabulary regarding web-based form controls; in other words, web form elements; and it consists of sixty-seven classes, sixty-six sub classes and two object properties.

Table 5.2 OASP with its Classes, Properties, Individuals, Data Range Identifiers, and Axioms

According to F. Zhang et al. an OWL 2 ontology can be formulated as a tuple $\boldsymbol{O} = \{ \boldsymbol{I}, \boldsymbol{P}, \boldsymbol{X}, \boldsymbol{D}, \boldsymbol{A} \}$ [41]. The letters *I*, *P*, *X*, *D*, and *A* stand for a set of individuals, a set of properties, a set of classes, a set of data range identifiers, and a set of axioms, respectively. Let’s assume that O_{ASP} represents ASPNET Control Ontology. It can be expressed as the following tuple:

$$O_{ASP} = \{ I, P, X, D, A \}.$$

I = { };

P = { *hasDataSourceItems*, *inInputOrSelectionControl* };

X = { *AspNetControl*, *AspNetDataControl*, *AspNetLoginControl*, *AspNetNavigationControl*, *AspNetStandardControl*, *AspNetValidationControl*, *Calendar*, *Checkbox*, *DropDownList*, *ListBox*, *Password*, *RadioButton*, *RadioButtonList*, *TextArea*, *TextBox*, *TextBoxGroup*,…….. };

D = { *xsd:Decimal* };

A = { SubClassOf(*DropDownList*, *AspNetStandardControl*),
SubClassOf(*ListBox*, *AspNetStandardControl*),
SubClassOf(*RadioButtonList*, *AspNetStandardControl*),
SubClassOf(*TextBoxGroup*, *AspNetStandardControl*),
SubClassOf(*Calendar*, *AspNetStandardControl*),
SubClassOf(*CheckBox*, *AspNetStandardControl*),
SubClassOf(*Password*, *AspNetStandardControl*),
SubClassOf(*RadioButton*, *AspNetStandardControl*),
SubClassOf(*TextArea*, *AspNetStandardControl*),
SubClassOf(*TextBox*, *AspNetStandardControl*),
ObjectProperty(*hasDataSourceItems* range*(DropDownList)*)
ObjectProperty(*hasDataSourceItems* range*(TextBoxGroup)*)
ObjectProperty(*hasDataSourceItems* range*(RadioButtonList)*)
ObjectProperty(*inInputOrSelectionControl* range*(Calendar)*),
ObjectProperty(*inInputOrSelectionControl* range*(TextBox)*),
ObjectProperty(*inInputOrSelectionControl* range*(RadioButton)*),
ObjectProperty(*inInputOrSelectionControl* range*(Password)*),
ObjectProperty(*inInputOrSelectionControl* range*(TextArea)*),
ObjectProperty(*inInputOrSelectionControl* range*(CheckBox)*)};

Because “Asp Net Data Control”, “Asp Net Login Control”, “Asp Net Navigation Control”, and “Asp Net Validation Control” are complex, and not generally usable controls on the web, the queries to match ontology classes to web form elements are generally made on "Asp Net Standard Control Classes." "Asp Net Standard Control" class has thirty sub classes. However, entire classes cannot be created as web form elements due to the same reason, which is mentioned above. Some of these web form controls have data source items; and some of them are input or selection controls. Because it is necessary to make a distinction between the two types of standard controls, two object properties named "hasDataSourceItems" and "*isInputOrSelectionControl*" are created within the "ASPNET Control Ontology." Table 5.2 illustrates its components.

Table 5.3 illustrates the SPARQL query, which aims to get classes in range of "*hasDataSourceItems*" object property.

Table 5.3 SPARQL Query for getting web controls that have data source items

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
CONSTRUCT  { ?s ?p ?o . }
WHERE { VALUES ?p { rdfs:range } ?s ?p ?o
FILTER
(?s = <http://opendatainagriculture.com/AspControls#hasDataSourceItems>)}
```

Table 5.4 shows the result of this query as well.

Table 5.4 Query result of the SPARQL query

Subject	Predicate	Object
hasDataSourceItems	rdfs:range	TextBoxGroup
hasDataSourceItems	rdfs:range	RadioButtonList
hasDataSourceItems	rdfs:range	DropDownList
hasDataSourceItems	rdfs:range	ListBox

Likewise, the classes in range of the object property "*isInputOrSelectionControl*" are shown in Table 5.5 as a result of SPARQL query represented in Table 5.6.

Table 5.5 SPARQL Query for getting web controls which are input or selection controls

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
CONSTRUCT  { ?s ?p ?o . }
WHERE { VALUES ?p { rdfs:range } ?s ?p ?o
FILTER
(?s = <http://opendatainagriculture.com/AspControls# isInputOrSelectionControl >)}
```

Table 5.6 Query result of the SPARQL query

Subject	Predicate	Object
isInputOrSelectionControl	rdfs:range	TextArea
isInputOrSelectionControl	rdfs:range	TextBox
isInputOrSelectionControl	rdfs:range	Password
isInputOrSelectionControl	rdfs:range	CheckBox
isInputOrSelectionControl	rdfs:range	Calendar
isInputOrSelectionControl	rdfs:range	RadioButton

The agricultural ontology which takes place within "Defining Suitable Control Types" layer of the third part of the proposed model is a kind of trait dictionary, as mentioned above. The classes of this type of ontologies are generally used to describe data with respect to the relevant agricultural product in a variety of categories. In other words, these classes represent in which categories, and what kinds of data should be collected with respect to the agricultural product. For instance, "BudColour" is a class in "Hazelnut Trait Ontology;" and it has three individuals: "Brown Green", "Green", and "Reddish". This means that if any stakeholders in hazelnut domain want to collect "BudColour" data, he/she has three options. So, these options have to be represented in appropriate web form elements. The control type suggestions are made for data acquisition forms as follows:

- If a class has individuals, then the model suggests the control types which match the web control classes in the range of "hasDataSourceItems" object property,
- If a class does not have any individuals, then the model suggests the control types which match the web controls classes in the range of "isInputOrSelectionControl" object property.

Table 5.7 shows which class of "ASPNET Control Ontology" correspond to which HTML element.

Table 5.7 Matching Web Control Ontology Classes to Forms Elements

ASPNET Ontology Class	HTML Element
TextArea	<textarea />
TextBox	____
Password	
CheckBox	☐
Calendar	____ (Date picker format)
RadioButton	☐
TextBoxGroup	____ (number of individuals)
RadioButtonList	☐ (number of individuals)
DropDownList	<select/>
ListBox	<select multiple="multiple"/>

According to the second layer "Creation of MVC" that belongs to the third part of the proposed model, data acquisition form should be created after the selection of classes and decisions regarding which control types are made. Data acquisition forms are generated based upon MVC design pattern using the HTML form elements which match the ontology classes in compliance with Table 5.7, within the scope of proposed model. The controller that bridges between views and controller and includes several actions regarding data acquisition forms is not dynamically generated because it enables us to create, delete, edit, and view operations for data acquisition forms. Furthermore, it is responsible to generate web pages to view and delete collected data using forms. However, the models and the views (in other words, generated data acquisition forms) are generated dynamically and stored with meaningful names in the databases. Data acquisition forms are stored as HTML strings in the databases and rendered again when one needs to view and use them for collecting data. The models which indicate the objects of programming language used to develop data acquisition forms creator are stored in the database with the name given to the form. Each ontology class that matches the relevant form elements is converted to the properties of the model. These models can be dynamically created and instantiated using the reflection libraries in the relevant programming languages in order to manipulate the data collected using data acquisition forms.

The last part of the proposed ontology-based data acquisition model is "Processing Created Forms and Handling Data." This part shows how to handle forms, how to export and store data collected using forms. It consists of three different layers such as "Form Actions", "Data Export Types", and "Data Source Options." "Form Actions" layer enables us to view, edit, and delete data acquisition forms, and also view and delete collected data using the relevant form. Furthermore, it identifies the boundaries regarding data acquisition forms. For instance, any form that is used to collect data cannot be deleted. Also, any form that is not activated by users cannot be viewed. “Data Source Options” layer recommends three options to store collected data: relational databases, RDF files, and graph databases. These options might be extended with any favorable databases or file types in compliance with the aim of the proposed model. However, because it might be necessary to use different methods and techniques to store data, additional data source options should be selected thoughtfully. The data collected and stored using data acquisition forms might be exported in three formats such as XML, RDF, and CSV in compliance with the layer “Data Export Types”. These data file types are recommended because they are frequently used by developers. Also, lots of programming languages have their own parsers to handle these types of data. However, any reasonable formats might be added to these formats. When extending data export options, it is important to consider whether the file format is generally used by developers and can have parser libraries for programming languages.

5.3 OWL2MVC Tool

OWL2MVC Tool is developed as a web-based software platform to create ontology-based data acquisition forms for the agriculture domain in compliance with the model introduced within the scope of previous part of this section. It is a kind of software application that runs integrated with open data platform which is currently developed for agriculture domain. OWL2MVC is a .NET software project. It is developed with C# programming language using ASP .NET Web Forms and ASP .NET MVC in Visual Studio 2017, which is a powerful, popular, and feature-rich integrated development environment. MS SQL Server 2016 is used to store models, views, and uploaded ontologies in the database. OWL2MVC Tool is a nonprofit free tool and can be used by any agriculture domain stakeholders such as analysts, researchers, and farmers if they register for the open data platform. The form elements are generated by OWL2MVC Tool and styled with Bootstrap, which is an open source and popular front-end component library to build responsive web-based applications.

In Appendix B screenshots of the relevant OWL2MVC Tool modules have been provided. Figure B.1 presents the module of uploading and viewing ontologies for data acquisition form generation. Figure B.2 illustrates the module of viewing and selecting ontology classes for data acquisition form generation. Figure B.3 represents the module of providing to map asp.net controls to ontology classes. Figure B.4 provides the module of viewing and editing the generated data acquisition form. Figure B.5 presents the module of listing generated data acquisition forms. Figure B.6 demonstrates the module of listing and exporting data collected using generated data acquisition form.

There are no restrictions or limits in terms of uploading ontologies on the open data platform. Any stakeholder can upload and view any number of agricultural ontologies in OWL format on the open data platform using OWL2MVC Tool. OWL2MVC is not a tool for developing and maintaining ontologies. It can only be used to generate ontology-based data acquisition web forms with respect to the relevant agricultural domain. It matches uploaded ontology classes to HTML form elements and rendered these elements as HTML strings. If any stakeholders who have experience with developing web forms using HTML, CSS, and JS can edit these rendered strings before storing them into the database. OWL2MVC Tool saves the uploaded ontology files into directories. On the other hand, it stores ontology name, ontology file name, and ontology uploaded date into the database as well. The tool creates menu items using the stored ontology information for accessing and viewing each of its data acquisition forms. So, the ontologies' file names should be a suitable name representing their domain. According to the proposed model, collected data using data acquisition forms can be stored in the graph databases. OWL2MVC Tool uses Neo4j Graph Database, which is one of the most popular, open sources and ACID-compliant transactional databases to store collected data.
In this section of the thesis, OWL2MVC Tool is generally introduced. However, a real-life instance of creating data acquisition forms using an agricultural ontology shall be elaborated in the "*Case Study*" part. Thus, it would be provided clearly and deeply understanding with respect to OWL2MVC Tool and the proposed model.

5.4 Case Study

Hazelnut Trait Ontology, which is introduced in the fourth section of the thesis, is used

Table 5.8 Selected Ontology Classes, Super Classes, and Individuals for Generating Data Acquisition Form

Super Classes Axioms	**Selected Ontology Classes Axioms**	**Individual(s) Axioms (*if exists*)**
SubClassOf (*CollectingDescriptor*, *Passport*), SubClassOf (*Passport*, *Descriptor*)	SubClassOf (*TypeofSample*, *CollectingDescriptor*)	Individual(*Pollen* type(*TypeOfSample*)) Individual(*Seeds* type(*TypeOfSample*)) Individual(*Tissue Culture* type(*TypeOfSample*)) Individual(*Vegetative* type(*TypeOfSample*))
SubClassOf (*InstitueCode*, *AccessionDescriptor*), SubClassOf (*AccessionDescriptor*, *Passport*), SubClassOf (*Passport*, *Descriptor*)	SubClassOf (*CuratorName*, *InstitueCode)*	-
	SubClassOf (*SiteWhereMaintained*, *InstitueCode)*	-
SubClassOf (*NutAndKernal*, *PlantDescriptor*), SubClassOf (*PlantDescriptor*, *Characterization*), SubClassOf (*Characterization*, *Descriptor*)	SubClassOf (*NutLength*, *NutAndKernal)*	-
	SubClassOf (*NutWidth*, *NutAndKernal)*	-
	SubClassOf (*ShapeOfNutApex*, *NutAndKernal)*	Individual(*Broad Acute* type(*ShapeOfNutApex*)) Individual(*Flat* type(*ShapeOfNutApex*)) Individual(*Narrow Acute* type(*ShapeOfNutApex*)) Individual(*Obtuse* type(*ShapeOfNutApex*))
SubClassOf (*PhenologyDescriptor*, *PlantDescriptor*), SubClassOf (*PlantDescriptor*, *Characterization*), SubClassOf (*Characterization*, *Descriptor*)	SubClassOf (*FirstFemaleBloomDate*, *PhenologyDescriptor*)	-
	SubClassOf (*FirstMaleBloomDate*, *PhenologyDescriptor*)	-
SubClassOf (*BudDescriptor*, *PlantDescriptor*), SubClassOf (*PlantDescriptor*, *Characterization*), SubClassOf (*Characterization*, *Descriptor*)	SubClassOf (*BudColour*, *BudDescriptor*)	Individual(*Brown Green* type(*BudColour*)) Individual(*Green* type(*BudColour*)) Individual(*Reddish* type(*BudColour*))
	SubClassOf (*BudShape*, *BudDescriptor*)	Individual(*Conical Pointed* type(*BudShape*)) Individual(*Globular* type(*BudShape*)) Individual(*Ovoid* type(*BudShape*))

as agricultural trait dictionary to generate data acquisition forms. TypeOfSample, CuratorName, SiteWhereMaintained, NutLength, NutWidth, ShapeOfNutApex, FirstFemaleBloomDate, FirstMaleBloomDate, BudColour, and BudShape classes have been selected from “Hazelnut Trait Ontology” by considering generating entire web controls. Table 5.8 illustrates the axioms of selected classes.

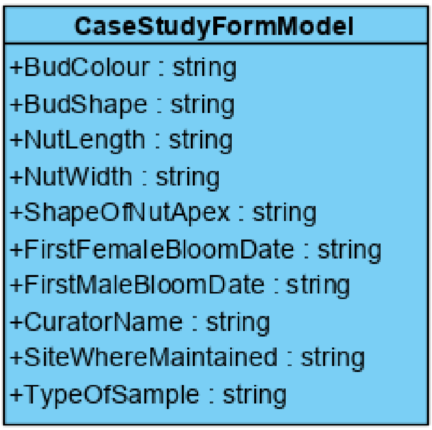
Users can change the display names of generated controls with super classes’ names which are listed into dropdown list controls on class definition and selection web page in OWL2MVC Tool. This allows users to define the best suitable display names for generated controls. If an ontology class has individuals, OWL2MVC Tool suggests only the following controls; DropDownList, ListBox, RadioButtonList, and TextBoxGroup. If an ontology class does not have any individuals, then it means that ontology class can match a control which is used to get scalar values from users. Table 5.9 represents selected classes, super classes of selected classes, selected control types, and whether selected classes have any individuals or not.

Table 5.9 Selected Classes, Superclasses, Control Types, and Presence of Individuals

Selected Super Class	Class Name	Has Individual(s)	Control Type
CollectingDescriptor	TypeofSample	TRUE	RadioButtonList
InstitueCode	CuratorName	FALSE	TextBox
InstitueCode	SiteWhereMaintained	FALSE	TextBox
NutAndKernal	NutLength	FALSE	TextBox
NutAndKernal	NutWidth	FALSE	TextBox
NutAndKernal	ShapeofNutApex	TRUE	DropDownList
PhenologyDescriptor	FirstFemaleBloomDate	FALSE	Calendar
PhenologyDescriptor	FirstMaleBloomDate	FALSE	Calendar
BudDescriptor	BudColour	TRUE	DropDownList
BudDescriptor	BudShape	TRUE	ListBox

As mentioned before, OWL2MVC Tool is developed by using ASP.NET Web Forms and MVC with C# programming language. ASP .NET MVC framework, which is released and introduced by Microsoft in 2009 with its first version, provides some benefits to the developers such as scale—which means the complexity of a web application--unit testing, and flexibility--which means responding easily to changes within the application development lifecycle (Freeman, 2014). The MVC framework is developed as an alternative to ASP .NET Web Forms, which have the following seven pillars: (i) ASP .NET is integrated with the .NET Framework; (ii) ASP .NET is compiled, not interpreted; (iii) ASP .NET is multi-language; (iv) ASP.NET is hosted by the common language runtime (CLR); (v) ASP .NET is object-oriented; (vi) ASP .NET supports all browsers; and (vii) ASP .NET is easy to deploy and configure (MacDonald and Freeman, 2010).

Due to the integration of ASP .NET with .NET Framework, the model created by OWL2MVC is a .NET object. Furthermore, it is a plain-old CLR object (POCO) created with C# programming language. This object is created in compliance with selected ontology classes. Rather than defining the classes of trait dictionaries as C# objects, it is more appropriate to be properties of the objects because the classes of agricultural trait dictionaries are generally used to characterize a variety of descriptors, which belong to different categories with respect to the relevant agricultural products. In object-oriented programming, properties are broad concepts to describe the specific characteristics of an object. They encapsulate the private fields inside the objects and provide a secure way to manipulate them. Therefore, the ontology classes are defined as properties of the POCO classes, which are used to pass data between views and controllers. However, it is important to emphasize that the approach we proposed is only convenient with trait dictionaries. Generating data acquisition forms process might not be accrued like with trait dictionaries when using other types of ontologies with this approach.




```
public class CaseStudyFormModel
{
    [Required(ErrorMessage ="*")]
    [Display(Name ="Bud Colour")]
    public string BudColour { get; set; }
    [Required(ErrorMessage = "*")]
    [Display(Name = "Bud Shape")]
    public string BudShape { get; set; }
    [Required(ErrorMessage = "*")]
    [Display(Name = "Nut Length")]
    public string NutLength { get; set; }
    [Required(ErrorMessage = "*")]
    [Display(Name = "Nut With")]
    public string NutWidth { get; set; }
    [Required(ErrorMessage = "*")]
    [Display(Name = "Shape of Nut Apex")]
    public string ShapeOfNutApex { get; set; }
    [Required(ErrorMessage = "*")]
    [Display(Name = "First Female Bloom Date")]
    public string FirstFemaleBloomDate { get; set; }
    [Required(ErrorMessage = "*")]
    [Display(Name = "First Male Bloom Date")]
    public string FirstMaleBloomDate { get; set; }
    [Required(ErrorMessage = "*")]
    [Display(Name = "Curator Name")]
    public string CuratorName { get; set; }
    [Required(ErrorMessage = "*")]
    [Display(Name = "Site Where Maintained")]
    public string SiteWhereMaintained { get; set; }
    [Required(ErrorMessage = "*")]
    [Display(Name = "Type of Sample")]
    public string TypeOfSample { get; set; }
}
```

Figure 5.3 Class Diagram Representing Form Model and Generated C# Class using Reflection Library

The generated data acquisition form named “CaseStudyForm” needs a model to transfer data to the controller. So, OWL2MVC Tool creates the .NET object named “CaseStudyFormModel” by using .NET reflection library to represent domain specific data and the state of the application for “CaseStudyForm”. Figure 5.3 shows the class diagram and generated C# class of “CaseStudyFormModel”. From Figure 5.3, it is apparent that each selected ontology class is defined as the properties marked with the public access modifier of the model and their data types are “strings.” Due to storing data gathered from generated forms as key-value pairs into the databases, it seems to be reasonable to define data types of each property as a “string”.

Table 5.10 Instances of Some HTML Strings for Case Study Form

HTML Control Type	Generated HTML String by OWL2MVC
DropDownList	<select name="BudColour" class="form-control"> <option value="Brown Green">Brown Green</option> <option value="Green">Green</option> <option value="Reddish">Reddish</option> </select>
ListBox	<select id="Bud Shape" name="BudShape" class="form-control" multiple="multiple"> <option value="Conical Pointed" selected="selected">Conical Pointed</option> <option value="Globular">Globular</option> <option value="Ovoid">Ovoid</option> </select>
TextBox	<input class="form-control" name="NutLength" size="30" type="text" value="">
Calendar	<input type="text" class="ff-behaviour form-control hasDatepicker" data-ff-behaviour="datetimepicker" data-ff-format="g" name="FirstMaleBloomDate" value="" id="dp1574617241801">
RadioButtonList	<div class="radio"> <label> <input type="radio" name="TypeOfSample" value="Pollen" checked=""> Pollen </label> </div>

Table 5.10 illustrates some HTML example strings of generated controls by OWL2MVC Tool, which belong to “Case Study View,” which are shown in Figure 5.4. Drop-down list and list box controls are created using <select> element. The only difference between these two controls is “multiple” attribute, which is used to convert a drop-down list to a list box control. The calendar control is a kind of <input type=”text”> control that is styled with jQuery and CSS to convert it to a time picker HTML control. Radio button list control consists of a series of div elements marked with a CSS class name as “radio”.

Each div is created in compliance with the number of individuals of the relevant ontology class including one <input type=”radio”> and <label> elements.

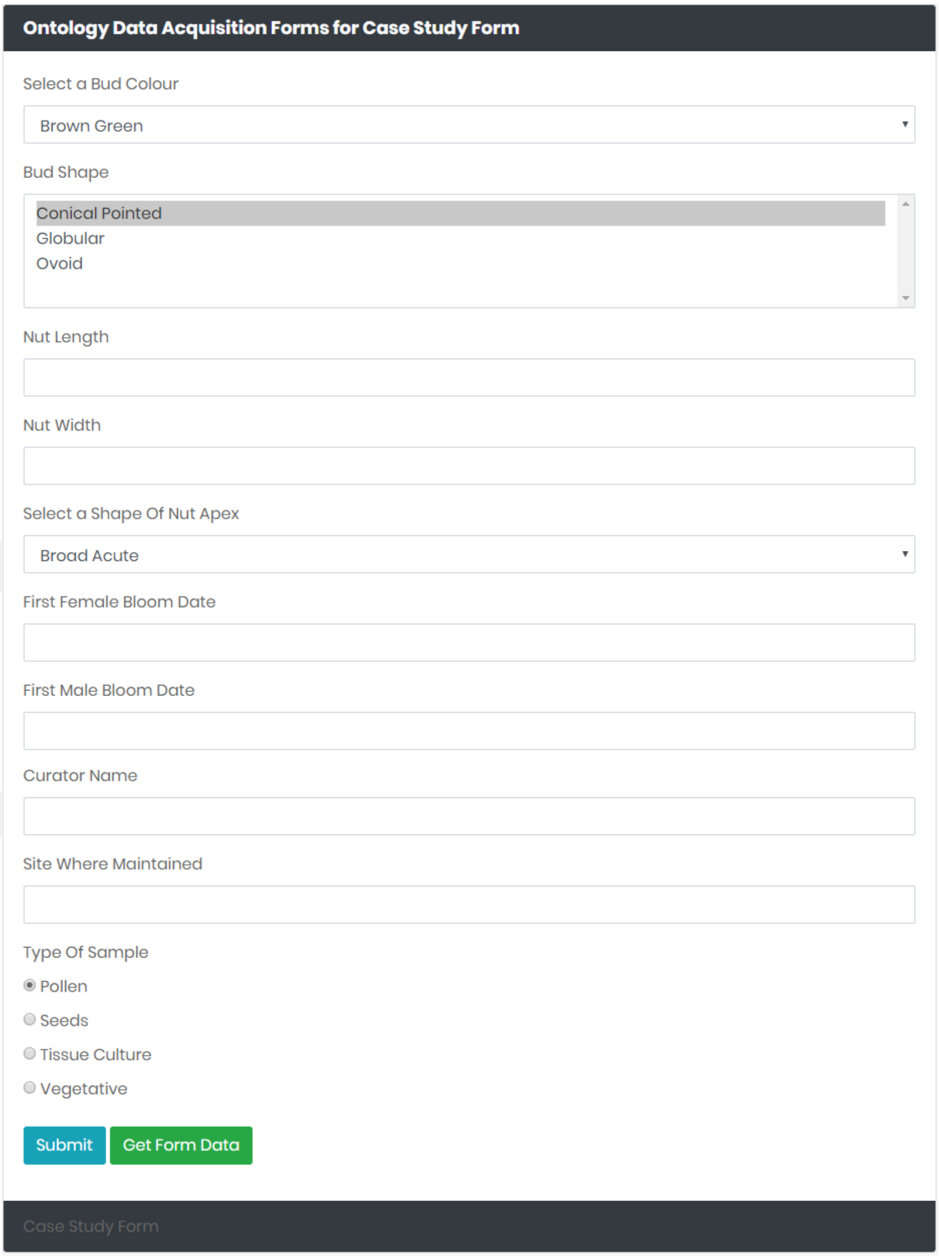


Figure 5.4 Generated Form

It is necessary to explain the other controls which do not exist within the “Case Study Form” as well. Check box HTML control is created using <input type=”checkbox”>. It also includes “onchange” event to detect whether control is checked or not. Creation and usage of password and text area HTML controls are the same with native HTML. They are created like <input type=”password”>, <textarea rows=””>, respectively. Although

TextBoxGroup is not a control, it is defined within the ASPNET Control Ontology to create text control groups for multiple data entrance. For instance, Hazelnut Trait Ontology has two classes named Rainfall and Temperature.

Any stakeholder might want to enter monthly mean data for these classes. Therefore, OWL2MVC Tool creates a series of HTML text controls named “TextBoxGroup” to meet this requirement. Each text control gets its name from the relevant ontology class’ individual’s name like the list items of RadioButtonList, ListBox, and DropDownList controls. As mentioned before, OWL2MVC Tool is developed in accordance with MVC design pattern. According to MVC design pattern, there should be a controller that acts as a bridge between models and views. This controller is “Form Creator” in OWL2MVC Tool. It is responsible to generate the HTML controls within the views for selected ontology classes. The final version of “Case Study Form” created by the controller is represented in Figure 5.4.

According to the proposed model, data gathered from generated forms can be stored in different kinds of database and file systems. In our case study, MS SQL Server and Neo4j database systems are used as relational and graph databases, respectively to store data. Each of form elements is stored in tables with their names, which comes from the relevant ontology classes. Entered data with respect to these form elements are stored under another column into the same table that correspond to form elements’ names. They might be considered as key-value pairs. Let’s assume that any user enters “Brown Green” value for the “BudColour” form element. In this case, the key is “BudColour,” and the value is “Brown Green.” Three tables are designed and created to store form data in MS SQL Server. The first table is created to save the following data with respect to forms; form name, form body, and form activation state. The second one is created to save form elements’ information and the third one stores the values of these form elements.

Figure 5.5 shows the graph generated by Neo4j of stored data gathered from the created data acquisition forms by OWL2MVC Tool. The data is stored as three nodes, which are “Form”, “FormProperty”, and “FormPropertyValue” in Neo4j Graph Database. “PROPERTY_OF” and “PROPERTY_VALUE_OF” are the relationship types which link these nodes. “PROPERTY_OF” connects the “Form” and “FormProperty” nodes to each other; and it specifies which form has what form elements. “PROPERTY_VALUE_OF” connects the “FormProperty” and “FormPropertyValue;” and it specifies which elements of form have what data value. In Figure 5.5, we can see that there exist four different forms such as Form 1, Form 2, Form 3, and Case Study Form, which are represented as green circle nodes within the graph stored in the database. The elements of data acquisition forms are represented as yellow circle nodes, and interconnected with green circle nodes (form nodes) with “PROPERTY_OF” relationship type. Lastly, it is apparent from the figure that purple circle nodes which represent the values of the form elements are interconnected with the yellow circle nodes by using “PROPERTY_VALUE_OF” relationship type.

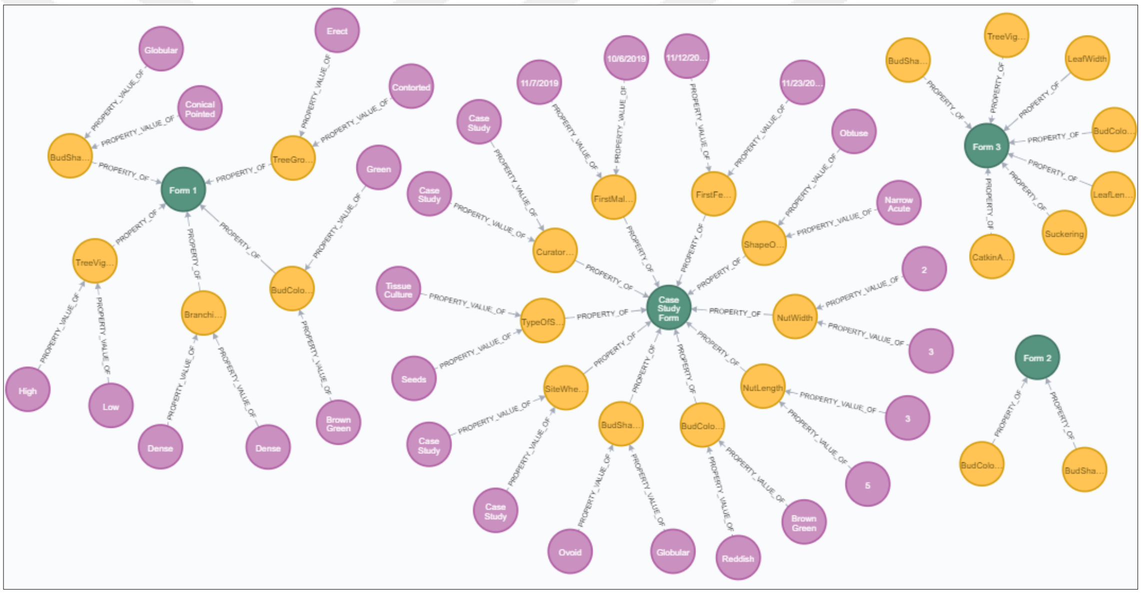


Figure 5.5 Storing Forms’ Data into Neo4j Graph Database

The Neo4j’s graph query language Cypher codes represented in Table 5.11 provides forms, elements of forms, and values of elements into Neo4j graph database. The Cypher codes could be run into Visual Studio using Neo4jClient and Ne4jDriver libraries which allow .NET developers to perform CRUD operations with C# programming language.

Table 5.11 Creating forms, elements of forms, and values of elements into Neo4j

```
CREATE(f: Form {
        FormId: { formId },
        FormName: { formName },
        OntologyId: { ontologyId },
        IsActive:  { isActive } })

CREATE(p: FormProperty {
        FormClassObjectPropertyId: { propertyId },
        FormId: { formId },
        FormClassObjectName: { objectName },
        FormClassObjectPropertyName: { objectPropertyName },
        FormClassObjectCreatedDate: {createdDate} })

CREATE(v: FormPropertyValue {
        FormClassObjectPropertyValueId: { propertyValueId },
        FormClassObjectPropertyId: { objectPropertyId },
        FormClassObjectPropertyInsertedValue: { insertedValue },
        FormClassObjectPropertyValueInsertedDate: { insertedDate } })
```

Table 5.12 shows the Cypher code how to create indexes for nodes in Neo4j. It can be seen from the table that FormId, FormClassObjectPropertyId, and FormClassObjectPropertyValueId have been created as the indexes for Form, FormProperty, and FormPropertyValue nodes, respectively.

Table 5.12 Creating Index

```
CREATE INDEX ON :Form(FormId)

CREATE INDEX ON :FormProperty(FormClassObjectPropertyId)

CREATE INDEX ON :FormPropertyValue(FormClassObjectPropertyValueId)
```

Table 5.13 Creating Data Relationships

```
MATCH (p:FormProperty), (f:Form)
WHERE p.FormClassObjectPropertyId = {propertyId} AND f.FormId = {formId}
CREATE (p)-[r:PROPERTY_OF] → (f)
RETURN type(r)

MATCH (v:FormPropertyValue), (p:FormProperty)
WHERE v.FormClassObjectPropertyValueId = {propertyValueId} AND p.FormClassObjectPropertyId =
{propertyId}
CREATE (v)-[r:PROPERTY_VALUE_OF] → (p)
RETURN type(r)
```

The relationships are used to connect entities in graph databases. As mentioned above, our graph has two relationship types named “PROPERTY_OF” and

“PROPERTY_VALUE_OF”. Table 5.13 illustrates Cypher code used to create relationships between Form, FormProperty, and FormPropertyValue nodes.

The data of acquisition forms stored in Neo4j can be listed using the code represented within Table 5.14. This code creates a table view that consists of four columns named FormId, PropertyName, PropertyValue, and Form, respectively for the relevant form.

Table 5.14 Select Query for Listing Case Study Form Data

```
MATCH (v:FormPropertyValue)→(p:FormProperty)→(f:Form)
WHERE f.FormId={ formId }
RETURN f.FormId as FormId,
p.FormClassObjectPropertyName as PropertyName,v.FormClassObjectPropertyInsertedValue as
PropertyValue,
COLLECT(f.FormName) as Form
```

5.5 Usability of OWL2MVC

OWL2MVC Tool is a kind of web-based software tool which generates ontology-based data acquisition forms and adapts them to the agricultural open data web platforms considering the stakeholders’ needs. It is based upon semantic web technologies, ontologies, and generally accepted design pattern MVC. Measuring the usability of such a new software application is important to determine whether the software meets the requirements or not. The usability term has been defined by ISO as “The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions” (ISO IEC 9126-1:2001, 2001). Usability of a software application is measured with a pre-specified set of tasks which might be performed by test or real users (Nielsen, 1993). Measuring the usability of OWL2MVC Tool, first, a case study approach was used to seek fifty-three respondents. They generated the data acquisition form which is proposed within the scope of “Case Study” part of this section. The task sheets which include organized listing of the data acquisition form generation processes to be followed have been delivered to each of the respondents. Furthermore, elapsed time of data acquisition form generation for each of the respondents has been recorded as well. After performing this case study, it is identified that all the participants completed all the tasks successfully.

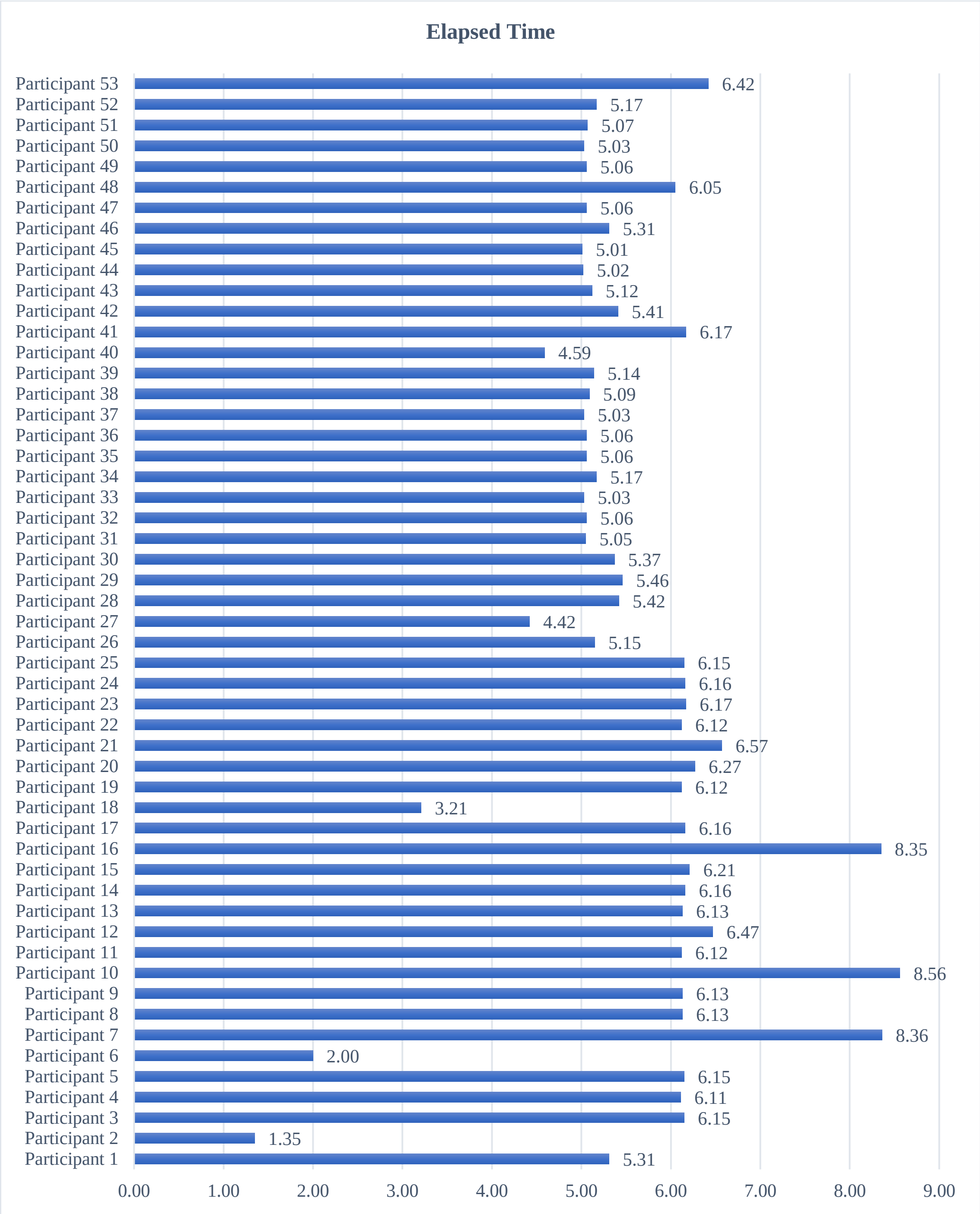


Figure 5.6 Elapsed time of generating data acquisition forms of each participant

The elapsed time for data acquisition form generation by each participant has been recorded using stop-watchers and they have been illustrated on Figure 5.6. According to the results, one might perform ontology-based data acquisition form generation in 5.56 minutes on an average. A web-based data acquisition form generation is generally a complex process which includes collecting requirements from users, designing user interface with CSS and HTML, creating data models, coding, and publishing. However, OWL2MVC tool facilitates this process using the strong characteristics of semantic web technologies.

After completing the case study for data acquisition form generation by fifty-three respondents, SUMI (Software Usability Measurement Inventory) questionnaire has been carried out to measure the usability of OWL2MVC Tool. SUMI is a tested, proven, and widely accepted method for measuring software quality in terms of user's perspective; and consists of a 50-item questionnaire (Kirakowski and Corbett, 1993). SUMI, which is a method of measuring the user experience, and was developed as part of MUSiC (Metrics for Usability Standards in Computing) project by University College Cork uses a meticulous scientific method of analysis; and has been supported by industrial applications for over twenty-five years (Kirakowski and Corbett, 1993) (Kirakowski, 1995) (Bevan, 1995) (Veenendaal, 2002). For the purpose of usability measurement, SUMI calculates five different scales: affect, controllability, efficiency, helpfulness, learnability, and global usability. It defines usability as "weighted composite of statements from each of these scales." The affect scale is psychological term for emotional feeling and refers to respondent feelings while interacting with the software. The controllability scale is what the feelings of respondents are when the software application responds in an expected and consistent way to inputs and commands. The efficiency scale is the degree of quickly locating and doing users' interests effectively and economically. The helpfulness scale means what the perceptions of respondents are when the software application runs in a helpful way and is of assistance for solving operational problems. The learnability means that one can use the software application with minimum instructions and understand it easily.

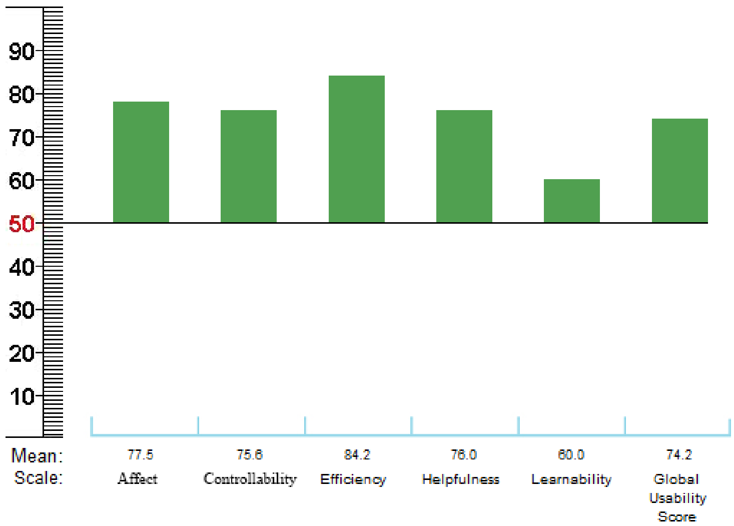


Figure 5.7 Graphical profile of OWL2MVC

The global usability score is based on the respondents' feeling of satisfaction concerning the software application being evaluated. Figure 5.7 presents the average values of five SUMI scales and usability score. Table C.1 represents the SUMI user records for OWL2MVC Tool.

Table 5.15 shows the numeric summary of SUMI results including mean and standard deviation of each scale. The mean value is numerical average of the individual scores of the respondents. The reasonable value of standard deviation is 20 for this kind of data. If standard deviation is smaller, it means that all respondents agreed on their evaluations of OWL2MVC Tool. If standard deviation is much greater, it means that the respondents have divergent opinions with respect to the usability of the tool. From the data in Table 5.15, it is apparent that standard deviation scores of each scale are smaller. In accordance with the results shown in Table 5.15 and Figure 5.7, all the respondents agreed on their evaluations of OWL2MVC Tool.

Table 5.15 Numeric summary of SUMI results

Scale	Mean	Standard Deviation
Affect	77.5	22.46
Controllability	75.6	24.16
Efficiency	84.2	22.64
Helpfulness	76.0	20.84
Learnability	60.0	22.97
Global Usability Score	74.2	19.23

The score for each scale is satisfactory in terms of usability since they are above the average (50). Furthermore, if the score is above 70, this means that the evaluated web application is exceptional on that scale. Considering this explanation, it is apparent from Figure 5.7 and Table 5.15 that OWL2MVC is exceptional on affect, controllability, efficiency, helpfulness, and global usability score scales. Even though learnability scale is below 70, it is satisfactory because its score is 60 and above average.

Table 5.16 and Table 5.17 (expands Table 5.16) show the results of measurements, which specify the importance of OWL2MVC Tool in terms of respondents' point of view. Seventy percent of respondents agree that OWL2MVC Tool is extremely important. Besides, twenty-two percent of the respondents accept OWL2MVC Tool as important. Only three respondents state that it is not very important.

Table 5.16 Responses to fixed questions: "How important for you is the tool OWL2MVC you have just been rating?"

Choice	Number of respondents	Global Usability Score (GUS)
Extremely important	38 (71%)	82.24
Important	12 (22%)	61.50
Not very important	3 (5%)	23.67
Not important at all	0 (0%)	0.00

Table 5.17 Mean rating for each of the SUMI scales

Choice	Affect	Controllability	Efficiency	Helpfulness	Learnability
Extremely important	84.26	85.21	92.55	82.82	68.55
Important	69.75	60.17	70.67	67.25	41.25
Not very important	22.00	16.00	32.33	25.00	26.00
Not important at all	0.00	0.00	0.00	0.00	0.00

In addition to these numeric results, three open-ended questions were answered by each user. The first question is "What part of OWL2MVC Tool do you find most interesting or useful?" Forty-nine respondents agreed that generating data acquisition forms using ontology dynamically and easily is the most interesting and useful feature of OWL2MVC Tool. The second question is "What do you think is the best aspect of OWL2MVC, and why?" Seven respondents did not answer this question. Forty-six respondents agreed that OWL2MVC Tool facilitates data acquisition form generation processes. Furthermore, they agreed on the following statement, generating such a form takes many hours. However, OWL2MVC provides that only in a few minutes. The last open-ended question is “Is there anything you think is missing from OWL2MVC Tool?” Forty-four respondents agreed that there is not anything missing from OWL2MVC Tool. Nine respondents did not answer this question.

Table 5.18 provides the summary statistics for reliability of OWL2MVC. Cronbach’s Alpha which is created by Lee Cronbach, means the measure of internal consistency of interrelated items in a test or scale (Cronbach, 1951) (Tavakol and Dennick, 2011). It was used to calculate the reliability of OWL2MVC considering six SUMI scales results of each respondents. As can be seen from Table 5.18, reliability calculation result is 0.936. When one looks the reliability calculation result, one can see that OWL2MVC has acceptable reliability value (Range of 0.70 – 0.95 is acceptable reliability value).

Table 5.18 Reliability of OWL2MVC

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	Number of Items
.936	.938	6

5.6 Limitations

In this chapter, a generic ontology-based agricultural data acquisition model which is constructed on MVC design pattern has been proposed. In addition, a tool named OWL2MVC has been developed based on the proposed model with the aim of creating data acquisition forms dynamically for open data platforms. There is no limitation in terms of using the proposed generic model. However, there are a few limitations need to be acknowledged in terms of using OWL2MVC tool. First, it has not been possible to collect real-domain data via a generated data acquisition form by OWL2MVC tool. Collecting data from different domain stakeholders such as farmers, researchers, and analysts depends on disseminating the usage of OWL2MVC tool by whole domain stakeholders. Second, only crop-specific trait ontologies which have been created using OWL can be used by OWL2MVC. However, including OBO format may boost usage of OWL2MVC tool.

6. ONTOLOGY-BASED DATA INTEGRATION

Enhancing crop production by applying the concepts, methods, and tools of information and communication technologies (ICTs) might be described as precision agriculture (PA) in a nutshell. One way of implementing PA might be to provide data with respect to meteorological factors such as weather humidity, temperature, rainfall, wind, and sunshine (Karim et al., 2017). Also, gathering and analyzing environmental and site-specific parameters such as topography, slope, soil moisture, soil pH, water availability, and soil fertility improve PA. Furthermore, PA facilitates boosting effectiveness and productiveness of an agricultural production system which consists of varied ICTs and organizes whole production cycle of plants and animals using a scientific approach (Mazon-Olivo et al., 2018). IoT technologies, particularly RFID and wireless sensor network (WSN) which are widely used in the agriculture domain, contributes improving efficient and safety agricultural production (Minbo et al., 2013). Advancement in sensor technologies provides increasing the usage of WSN in precision agriculture. WSN-based agricultural systems which are supported with mobile-based or web-based data management tools might improve productivity of farmers by providing mixed crop farming and ubiquitous computing (Muangprathub et al., 2019).

Most of meteorological factors, environmental and site-specific data might be obtained by utilizing the sensor technologies. Farmers, who have the key role in agricultural production cycle, might benefit for themselves by collecting, processing, providing, and using data efficiently (Paraforos et al., 2017). In other words, it is necessary to provide farmers with relevant data related to whole agricultural production activities to make effective decisions. Considering the high effect of environmental and site-specific data throughout agricultural production cycle, processing and managing such data obtained from sensors is crucial in terms of achieving PA.

In the agricultural domain, there might be varied data sources such as farmers, sensors, market, and government stats related to agricultural products. The data gathered such heterogeneous sources should be processed, integrated, and exchanged through open data platforms which are the major approach for providing openness in the agriculture domain.

Considering the lack of research endeavors on the application or platform level for ensuring integrated web-accessible data in a unified way, it is clear that more focus is needed on open distributed platforms which facilitate integrating sensor data and understanding the exact meaning of integrated data (Nagib and Hamze, 2016). Thus, this section of the thesis is aimed to develop an Open Data Platform which fulfills the requirements mentioned above and is built upon the very idea of an ontology-based data management (OBDM) system. OBDM, which is identified as a system constructing the representation of ontology explicitly and makes links between this ontology and data sources using a formal way, thus consists of such layers as ontology, mapping, and data sources (Daraio et al., 2016). These layers have vital role while performing the data integration processes for agricultural open data platforms.

An ontology-based data integration approach has been proposed considering the significance of ontology, mapping, and data storage layers in this section of the thesis. In addition, data integration components and tools of Open Data Platform are developed based on the proposed ontology-based data integration approach. The proposed approach examines data integration processes from IoT devices to open data platforms considering syntactical and semantic interoperability levels. With this approach it shall be elaborated how to use agricultural trait dictionaries for achieving semantic interoperability. There has been a great deal of purposes behind motivation of proposing this approach. First, this approach illustrates how the agricultural data is gathered from IoT devices using varied sensors in WSNs, how the gathered data is integrated to the open data platform and how to publish this data through web services for consuming by domain stakeholders. Second, it depicts how to annotate agricultural data using the strength of semantic web technologies. In addition, it constructs a framework to create data models for object relational mapping for data storage options.

As can be seen from this approach, ontology, mapping, and data sources layers play important role of integrating sensor data into agricultural open data platforms. Domain ontologies might be used to annotate data gathered from devices with the contextual information thus, providing well understood data by the users and data interoperability might be achieved (Kaed et al., 2018). In this study, hazelnut trait dictionary which plays the domain ontology role of open data platform and is introduced in the fourth section of the thesis, is used to make connections between sensor measurement values and site-specific parameters of agricultural products. Furthermore, it provides transforming mapped sensor data to linked-open sensor data as well. Mapping procedures are approved by domain experts using the rule engine of Open Data Platform. Thus, transformed data is became exportable and downloadable in open formats such as RDF, JSON, Turtle, N-Triples etc.

The platform draws on the strength of semantic web technologies while generating operations of mapping sensor devices' data with trait dictionary, creating datasets stored in open formats, and performing manipulation on stored data.

The objectives of the developed open data platform are as follows:

(i). to collect domain-specific data concerning particular agricultural products through ontology-based data acquisition forms generated by domain stakeholders using agricultural trait dictionaries;

(ii). to gather and visualize stream data concerning site-specific parameters of particular agricultural products through WSNs;

(iii). to produce domain-specific linked open data using mapping rules constructed by any domain stakeholder using agricultural trait dictionaries;

(iv). to store semantically annotated agricultural data within the diverse databases and files such as relational databases, graph databases, XML files, RDF files etc.;

(v). to provide syntactical interoperability using Web Services and APIs which allow stakeholders share data for a particular agricultural product between different kinds of software applications;

(vi). to publish well-defined, well-structured, and semantically annotated data concerning a particular agricultural product using open standard in appropriate formats such as RDF/XML, RDF/JSON, N-Triples, Notation 3, Turtle, XML, JSON, HTML, CSV, and Excel.

Considering the entire objectives of the developed open data platform are constructed based on agricultural trait dictionaries, it would be appropriate to describe them as the main pillars of the platform. The domain stakeholders who want to use the capabilities of the open data platform might use an existing agricultural trait dictionary or upload a new one to the platform using the user interface represented on Figure A.1 to achieve their purposes.

This section of the thesis is organized as follows. Part 6.1, Part 6.2, Part 6.3, and Part 6.4 describe the general terms focused within the scope of this section and addresses some challenges regarding semantic interoperability. The architecture of the proposed approach is detailed in Part 6.5. Part 6.6 and Part 6.7 introduce the implementation of open data platform which integrates IoT data based on the proposed approach. Lastly, Part 6.8 illustrates the evaluations results of the relevant tools of developed Open Data Platform in terms of usability.

6.1 Internet of Things (IoT)

The presentation of Kevin Ashton in 1999, which is in the context of supply chain management system, pioneered appearing the term Internet of Things (IoT) (Ashton, 1999). Today, IoT applications are being used in a wide range of sectors such as discrete manufacturing, transportation, logistics, utilizes, B2C, healthcare, energy and natural resources, retail, insurance etc. Gubbi et al., distinguishes IoT into three components which enable ubiquitous computing such as hardware, middleware, and presentation (Gubbi et al., 2013). Atzori et al., define IoT as a paradigm and it might be characterized in three visions such as internet-oriented, things-oriented, and semantic-oriented (Atzori et al., 2010). According to the forecast of International Data Corporation (IDC), there will be 41.6 billion connected IoT devices, or “things”, and the amount of data generated by these devices will reach to 79.4 zettabytes in 2025 (Shirer and MacGillivray, 2019). According to Jabbar et al., there exist varied issues to handle with respect to IoT, such as standards, scalability, device diversity, constructing generally accepted service language, finding service for a particular domain, providing interoperability (Jabbar et al., 2017) (Jabbar et al., 2017). One of the most important issue regarding IoT is interoperability.

6.2 Interoperability in IoT

Considering the vast amount of data gathered from IoT devices, semantic web technologies might play a significant role to represent, store, interconnect, discover, and organize this data. Furthermore, semantic web technologies are an important way of revealing the common knowledge of a particular domain by performing reasoning procedures (Rhayem et al., 2017). IoT data, in machine-readable format might be ensured by using semantic web technologies thus, meaning of data might be apprehended clearly (Su et al., 2014). Understanding the true meaning of data by utilizing the semantic web technologies is a key factor which simplifies providing interoperability among IoT components. Semantic web technologies are also used for creating data models and integrating data gathered from heterogeneous IoT devices. Annotating IoT data using the strength of semantic web technologies allow to ensure the interoperability between IoT applications and to make these applications smarter (Al-Osta et al., 2017). Utilizing IoT data which are obtained from heterogeneous data sources requires more complex operations and technical skills due to interoperability issues. Considering the heterogeneity of IoT data, semantic web technologies might be used for sharing data gathered from heterogeneous IoT devices by generating common models (Elsaleh et al., 2019). The diversity of IoT devices from several vendors might uncover semantic and syntactic errors (Ullah et al., 2017). However, IoT data generally bound up with heterogeneous data models, to overcome this heterogeneity it is needed to make data available homogeneously to allow integration from wide variety of sources (Nagib and Hamza, 2016). With the aforementioned reasons, it is absolutely necessary to take a closer look at interoperability in IoT. Interoperability has been acknowledged as an important issue in IoT due to its heterogeneous structure.

Interoperability between two systems is addressed into different levels by researchers. Wang et al., describes interoperability between two systems as seven levels of conceptual interoperability model (Wang et al., 2009). Level 0 means that there is no interoperability between systems. Level 1 which is called as technical interoperability as well, identifies the networks and protocols to communicate between systems. Level 2 named syntactic interoperability, indicates what kind of software infrastructure is needed while sharing or exchanging data between systems. This level deals with structuring data formats shared between systems, also it might cover developing Web Services or APIs to exchange data in relevant formats (Wang et al., 2009) (Dobrev et al., 2007). Level 3 known as semantic interoperability, addresses the extracting the meaning of data. Level 4 defined as pragmatic interoperability in the model, is related to use of the information. Level 5 called as dynamic interoperability, defines effect of exchanging information between systems. Level 6 which is the last and highest level of the model and named conceptual interoperability, specifies the need of documenting conceptual models using engineering methods (Wang et al., 2009). Pantsar-Syväniemi et al., proposed a model for interoperability of smart environments that consist of several physical things such as devices, actuators, sensors, they also separated this model into six different interoperability levels which are connection, communication, semantic, dynamic, behavioral, and conceptual (Pantsar-Syväniemi et al., 2012). Noura et al. addressed IoT interoperability into five perspectives considering the models which are proposed by different researchers (Noura et al., 2019) (Noura et al., 2018). They categorized IoT interoperability into five different interoperability levels such as device, network, syntactic, semantic, and platform.

Semantic interoperability perspective means that providing common understanding for the things and sharing the data obtained from these things by reducing ambiguous data descriptions for varied stakeholders (Jayaraman et al., 2015). Semantic interoperability has some advantages in comparison with the other interoperability perspectives. However, it is worth bearing mind that implementing semantic interoperability requires more affords such as developing an ontology and utilizing it to meet the interoperability requirements (Ganzha et al., 2017). Ontologies which allows to apply inference-based techniques in the integrated data play a key role while implementing semantic interoperability in IoT. There exist many ontologies with respect to IoT devices, particularly sensors. These ontologies generally aim to model resources, services, and location information, however new ontologies are developed to meet interoperability requirements as well (Tayur and Suchithra, 2019). The overall purpose of these ontologies is to provide robust solutions to heterogeneity issues regarding hardware, software, and the data management for IoT devices and they might be classified in four categories divided into generic and domain ones such as sensor, context-aware, location, and time-based (Bajaj et al. 2017).

6.3 Ontologies in IoT

Avancha et al., proposed a sensor node ontology which identifies sensor node's essential elements (Avancha et al., 2004). Matheus et al., created an ontology for one-level sensor fusion regarding naval operations, as well as the general concepts of this ontology are suitable to apply to any domain which includes sensor fusion (Matheus et al., 2005). Russomanno et al., created OntoSensor which is a prototype sensor knowledge repository and involves the definitions of concepts and properties adopted in part from SensorML, extensions to IEEE SUMO and references to ISO 19115 (Russomanno et al., 2005). Eid et al., built an ontology called sensor-data ontology for managing information of sensors in an efficient way, and to ease data retrieving processes for users or search engines (Eid et al., 2006). Neuhaus and Compton developed an ontology which identifies the facilities and activities of the sensors (Neuhaus and Compton, 2009). SWAMO Ontology which is compatible with existing ontologies such as Sensor Web Enablement (SWE) and Sensor Model Language (SML) developed by Open Geospatial Consortium (OGC), provides making decision automatically and handling the requests from dynamic Sensor Web environment (Witt et al., 2008) (Underbrink et al., 2008). SWE Common Data Model, describes models which are used to transmit data obtained from sensors in low-level (Robin, 2011). SML is one of the standards which are generated under SWE, and it ensures both semantic and syntactic interoperability (SensorML, 2011). Another ontology which identifies sensors, their accuracy and capabilities, and their procedures utilized while sensing, is Semantic Sensor Network (SSN) (Compton et al., 2012). Sensor, Observation, Sample, and Actuator (SOSA) ontology includes core concepts which are used by SSN as well (Haller et al., 2018). The MMI Device Ontology, which is developed by Marine Metadata Interoperability (MMI), identifies varied types of instruments using an extendible conceptual model and controlled vocabularies (Rueda et al., 2010). The Extensible Observation Ontology (OBOE), which is a formal ontology and used for modelling scientific observation and measurement semantically, provides better interpretation of data, eases reusing, and enables developing particular and adequate systems for searching and discovering the data (W3C XML, 2015).

6.4 Challenges in IoT in terms of Interoperability

Several challenges might be encountered while performing semantic interoperability between systems. Liyanage et al., categorized challenges concerning semantic interoperability into four levels such as meaning, granularity, temporal, and structural (Liyanage et al., 2015). On the other hand, Gyrard et al. mentioned six different challenges which should be handled to achieve semantic interoperability in IoT applications (Gyrard et al., 2018). These challenges are as follows; a unified model to semantically annotate IoT data, reasoning mechanisms, linked data approach, horizontal integration with existing applications, design lightweight versions for constrained environments, and alignment between different vocabularies (Gyrard et al., 2018).

As well as considering the aforementioned challenges, the following issues might be addressed in agriculture domain while integrating IoT data into open data platforms:

(1). decomposition of raw sensor data stream in compliance with the measurement data types,

(2). cleaning decomposed data,

(3). creation of object models to provide general infrastructure of storing sensor data,

(4). designing databases to store data,

(5). mapping heterogeneous sensor measurement data types with the relevant traits of agricultural product,

(6). annotating IoT data using agricultural trait dictionaries,

(7). determining the sensor measurement data is related to the particular agricultural product,

(8). providing data exchange between applications using common data formats, and

(9). sensor data exportation using open data formats and/or open standards.

This section of the thesis set out with the aim of designing and creating solutions to overcome the aforementioned challenges. Therefore, a comprehensive architecture is proposed in the further part of this section to achieve applying ontology-based data integration approach for sensor data obtained from IoT devices, stored and export in open data formats and shared through web services/APIs. In addition, this architecture focuses on semantic and syntactical interoperability in IoT.

6.5 Ontology-based Data Integration Approach for Agricultural Open Data Platforms

As shown in Figure 6.1, the proposed approach consists of four layers which provide acceptable solutions for data integration issues in terms of publishing data in appropriate formats from agricultural open data platforms to domain stakeholders.

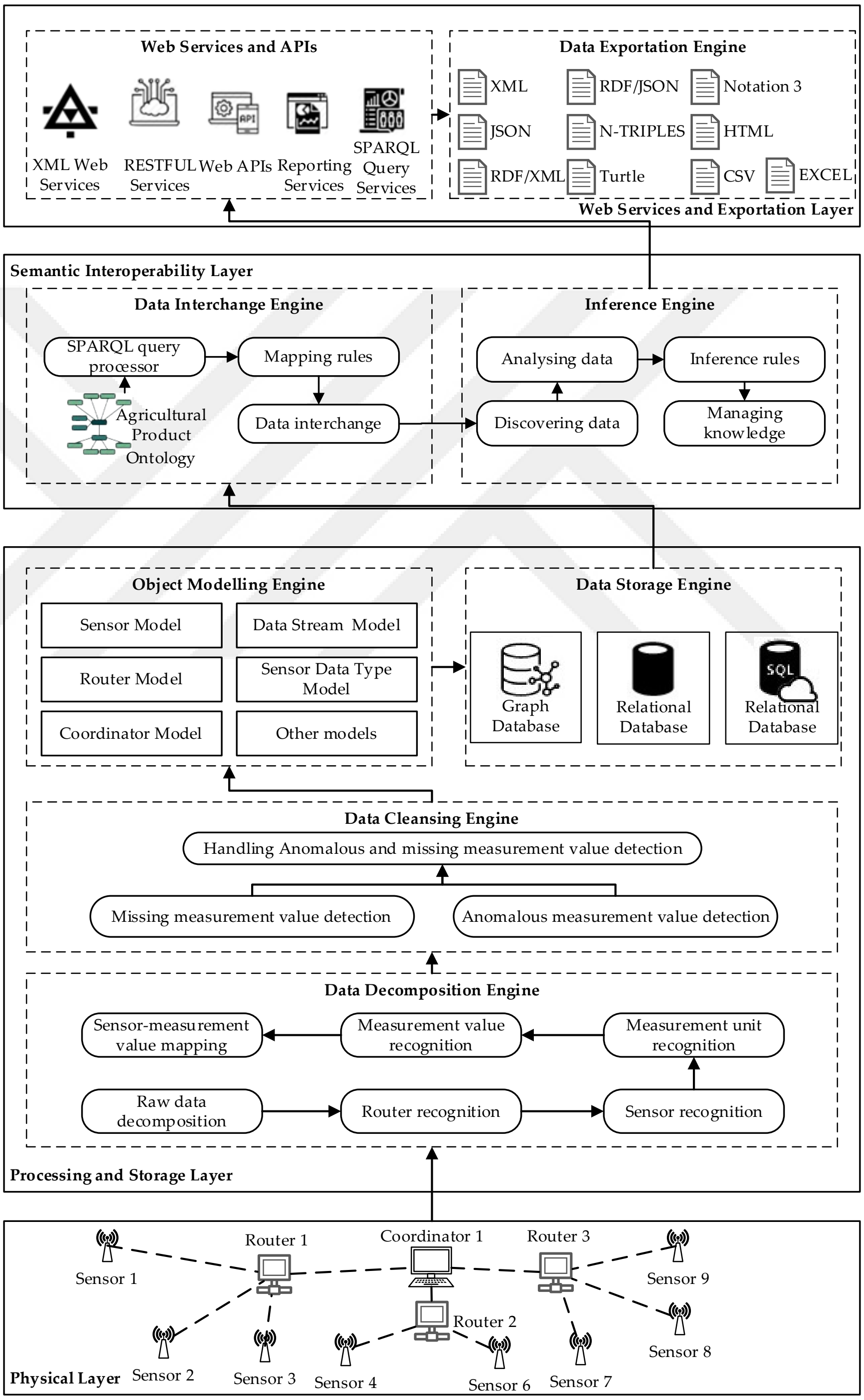


Figure 6.1 Proposed Ontology-based Data Integration Approach

These interconnected layers are

(1) Physical Layer,
(2) Processing and Storage Layer,
(3) Semantic Interoperability Layer, and
(4) Web Services and Exportation Layer respectively.

Physical Layer (PL). In agriculture domain, IoT devices have been frequently used for gathering environmental and site-specific data such as soil moisture, soil salinity, soil pH, and climate of the site (temperature, rainfall, wind, light, relative humidity). PL, which is the primary level of proposed approach, comprises IoT devices formed as wireless sensor networks (WSN) which are structured as several routers controlled by one coordinator. Two variants of WSNs are generally used in agricultural applications: terrestrial WSNs and underground WSNs (Ojha et al., 2015). These agricultural WSNs applications might use different WSN standardizations such as ZigBee, WirelessHART, and 6LoWPAN. There is no restriction in terms of choosing WSN types and standards according to the proposed approach. Thus, any options might be used while establishing IoT devices on any agriculture field.

Processing and Storage Layer (PSL). This layer corresponds to decomposing raw sensor data streams, cleaning raw sensor data, modelling objects, and storing data. PSL enables transforming raw sensor data into appropriate format mapped to object models and storing this data into varied database options. PSL is composed of four sub-layers; Data Decomposition Engine (DDE), Data Cleansing Engine (DCE), Object Modelling Engine (OME), and Data Storage Engine (DSE). Next, these four sub-layers have been described in detail.

-Data Decomposition Engine (DDE). DDE handles with decomposing the raw sensor data that are generally in varied forms. Sensors are low-cost sensing solutions for measuring environmental and site-specific data in agricultural applications. As well as sensors measure at least one value, they might measure more as well. Considering there are varied sensors plugged on router devices, proposed approach recommends using key-value pairs to define sensors on WSNs. Suppose that one router (id=1) has a sensor (id=1) that measures weather humidity (key=WH), weather temperature (key=WT), and barometric pressure (key=PR) at the same time. There might be different ways for composing such a kind of data stream. However, the proposed approach solves this issue using the following pattern.

[Router Identifier]:[Value];[Sensor Identifier]:[Value];[Sensor Measurement Identifier]:[Measurement Value];

According to this pattern, sensor measurement value might be formulated as RID:1;SID:1;WH:45;. This means that the value %45 of weather humidity is measured by the sensor numbered 1 which is plugged to router 1. Figure 6.2 illustrates how to decompose sensor raw data stream. DDE decomposes the raw sensor data and extracts router definitions, sensor definitions, and sensors' measurement values from this data.

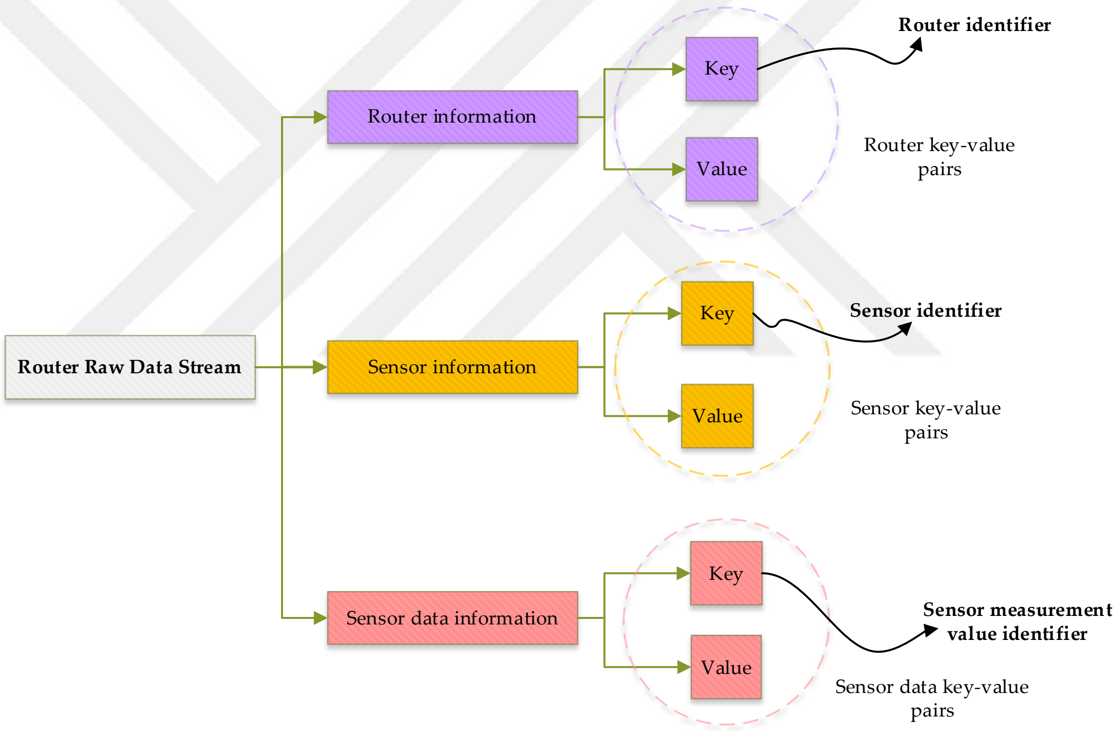


Figure 6.2 Decomposition raw sensor data stream

DDE is also responsible to match the sensor measurement value to its unit using sensors information stored in the databases or any files and to map sensors to their measurement value for creating appropriate object models of routers and sensors to utilize by visualization application tools. To meet mapping requirements, key-value pairs of routers and their sensors might be passed as parameters through the queries. When sensor raw data stream is received by WSN coordinators, recognized router and sensor id are retrieved from databases or files. Then, mapping procedures are accomplished by data decomposition engine.

-Data Cleansing Engine (DCE). Missing value problems frequently occur within the sensors, and it is accepted quite widespread in WSNs. There exist several various problems which cause missing value in WSNs, such as synchronization issues, sensor errors, or transmission issues (Li and Parker, 2008). DCE is in charge of handling missing sensor measurements and anomalous values. It might be very crucial to estimate missing sensor measurement values and detect anomaly in terms of critical systems. In case of anomaly detection in WSNs statistical, support vector machine (SVM), and cluster analysis techniques play a significant role (Xie et al., 2015). These techniques might be applied to sensor data stream since providing solutions for estimating missing values and detecting anomaly.

-Object Modelling Engine (OME). OME, which is a sub-layer of PSL, enables creating coordinator object model, router object model, sensor object model, data stream object model, and sensor data type object model. In other words, it provides mapping database objects to high-level abstract models created within any programming language using object-oriented programming techniques.

-Data Storage Engine (DSE). DSE is responsible storing data obtained from sensors in three different storage options such as graph database, relational database, and cloud database.

Semantic Interoperability Layer (SIL). SIL plays an important role in the data integration process of the proposed approach. Whole considerable procedures for data integration such as executing SPARQL queries, applying mapping rules, making data interchange, and storing mapped data within varied file formats are carried out within this layer. Agricultural ontologies, in other words agricultural trait dictionaries have also very critical role of the sub-component named data interchange engine of SIL. It utilizes agricultural trait dictionaries to map sensor measurement values to relevant traits of agricultural product for publishing more annotated data through open data platforms for usage of any domain stakeholders. Domain experts may create these mapping rules by selecting relevant ontology classes and mapping them to corresponding sensor measurement values. Mapping rules are also used to link data obtained from IoT devices with agricultural ontology classes. They might be stored in RDF format and implemented by retrieving and manipulating with SPARQL to perform interchange operations when needed. Discovering information in RDF datasets is performed using SPARQL and the result of the relevant SPARQL query is formed as RDF triples (Charalampidis and Keramopoulos, 2018). The inference rules might be defined by domain stakeholders via the relevant module of the platform which is represented on Figure A.3 to help managing knowledge in the context of agriculture. These rules assist in domain stakeholders' understanding of the true meaning of sensor measurement values to perform more precise agricultural activities.

Web Services and Exportation (WSE). The open data platforms might publish data in different formats. The proposed approach allows users choose directly downloadable and web services options considering heterogeneity of applications which utilizes the published data. The platform users might export linked open data which are created by mapping agricultural ontology classes with data gathered from IoT devices in varied file formats such as XML, JSON, HTML, CSV, Excel spreadsheets, RDF/XML, RDF/JSON, N-Triples, Turtle, and Notation 3. Table 6.1 gives a brief overview with respect to these data formats.

On the other hand, data also might be published though web services in XML and JSON formats. Consuming web services to process data requires relatively less afford rather than semantic web technologies. However, in the open data world it is indispensable to publish data in open data formats. Software applications which are designed and developed for stakeholders belonging to agriculture domain might utilize published data through open data platforms. Software applications handling published data might be a desktop application, web application, or mobile application. A major problem with this software applications heterogeneity in terms of utilizing published data is that there need to be a mechanism provides suitable data formats for each types of software applications.

Table 6.1 Data formats used for the proposed approach

File Format	Abbreviation	Mime Type	Description
eXtensible Markup Language	XML	text/xml	Software and hardware independent for sharing data between different applications (W3C XML, 2015).
JavaScript Object Notation	JSON	application/json	Frequently used lightweight format for sharing and storing data (W3C JSON, 2020).
Hypertext Markup Language	HTML	text/html	Standard markup language for Web pages (W3C HTML, 2020).
Comma-separated values	CSV	text/csv	A type of data file which separate values using comma
Excel spreadsheet	Excel	application/vnd. ms-excel	Stores data into rows and columns, provides powerful analyzing capabilities and calculating operations.
Resource Description Framework	RDF/XML	application/rdf+xml	Standard model for data interchange on the Web (W3C RDF, 2014).
	RDF/JSON	application/json	An RDF graph to be written in a form compatible with JSON (Davis et al., 2013).
	N-Triples	application/n-triples	A line-based, plain text format for encoding RDF graph (Beckett, 2014)
	Notation 3	text/n3	An assertion and logic language, superset of RDF (Berners-Lee and Connolly, 2011).
Terse RDF Triple Language	Turtle	text/turtle	An RDF graph to be written in a compact and natural text form (Beckett et al., 2014).

The proposed approach seeks to come up with developing XML Web Services, REST Web Services, and Web APIs which help to address and provide robust and effective solutions for data type compatibility issues among software applications. In addition, web services, particularly RESTful web services, might help solving interoperability problems by allowing software applications which run on different platforms request, access, and manipulate data which can be easily processed. Interoperability between software applications is not only a problem for applications run on different platforms. It might be a major problem for applications run on the same platform as well. Therefore, the proposed approach strongly recommends developing REST Web Services, Web APIs, or both. Another service of proposed approach is Reporting Service which provides creating and managing well-designed data reports. The last service is SPARQL Query Service which is developed for executing the SPARQL query statements to retrieve data from RDF datasets.

6.6 Development of Ontology-based Open Data Platform in accordance with the Proposed Approach

Within the scope of this section of this study, an ontology-based open data platform which is developed by following the layers of proposed approach is introduced. The platform has several modules such as ontology visualization tool and data acquisition form generation tool, however the main focuses of this section are on how to integrate sensor data using agricultural trait dictionaries into open data platform and how to make use of integrated data for domain stakeholders in open data file formats.

The initial component of the open data platform is the establishment of a typical wireless sensor networks to meet the requirements of gathering environmental data using sensors. Hence, a WSN, which consists of one coordinator and seven sensors plugged on three routers was built. The first sensor plugged on router 1 is BME280. It can measure relative humidity from 0 to 100% with ±3% accuracy, barometric pressure from 300Pa to 1100hPa with ±1hPa absolute accuracy, and temperature from -40°C to 85°C with ±1.0°C accuracy. The second sensor plugged on router 1 is GUVA-S12SD UV. It detects the UV wavelength from 240nm to 370nm in sunlight. The third and the last one plugged on router 1 is MICS-4514. It can detect the following gases: carbon monoxide from 1ppm to 1000 ppm, nitrogen dioxide from 0.05 ppm to 10 ppm, ethanol from 10 ppm to 500 ppm, hydrogen from 1 ppm to 1000 ppm, ammonia from 1 ppm to 500 ppm, and methane greater than 1000ppm. Router 2 has three sensors such as soil moisture, rain, and BH-1750 light level. The soil moisture sensor categorizes its measurements as dry soil, humid soil, and in water. The rain sensor, actually it is a raindrop sensor, does not measure rainfall. It categorizes its measurements as raining, rain warning, and not raining. BH-1750 sensor can measure light intensity from 1lx to 65535lx. Router 3 has only one sensor named MPU6050 which is 6 axis acceleration and gyro sensor. These sensors were defined for representation using abbreviations by a set of named constants which are illustrated on Table 6.2.

Table 6.2 Enumerating measurements types of sensors

Sensor	Measurement	Abbreviation
BH1750 - Light Level Sensor	Light Intensity	LI
BME 280 Temperature Humidity Barometric Pressure Sensor	Weather Temperature	WT
	Weather Humidity	WH
	Pressure	PR
	Altitude	AL
MPU6050 6 Axis Acceleration and Gyro Sensor	Gyro measurement range	GR
Rain Sensor	Rain Rate	RR
Soil Moisture Sensor	Soil Moisture	SM
UV Sensor Module Arduino Ultraviolet Ray I2C	Ultraviolet	UV
MICS-4514 Carbon Monoxide Nitrogen Oxygen Sensor	Carbon monoxide	CO
	Nitrogen dioxide	NO2

Coordinators and their routers are illustrated on maps on open data platform according to their latitude and longitude information. Platform users might access the detailed information of WSNs by following the link on coordinators tooltips. Figure 6.3 shows how coordinators and routers are viewed on map by using their location information. Each of sensors, routers, and coordinators might be visualized as hierarchical nodes on the platform and platform users might access sensors' data which are measured in last five minutes and visualized into appropriate graphs by using clickable feature of the routers' icons. Figure 6.4 shows how each component of WSN is viewed on the open data platform.

It is a well-known fact that reading sensor inputs and transforming them into meaningful format are required using microcontrollers. Therefore, Arduino, which is an open-source member of family of electronic microprocessor boards based on easy-to-use hardware and software was used as router and coordinator devices within the WSN. Other reasons to use Arduino as router and coordinator devices are following; it is inexpensive, its IDE runs on cross-platforms and easy-to-use, its hardware and software are open source (Arduino Web Page, 2020). Furthermore, it ensures key features to collect both economic and practical benefits for researchers (González-Buesa and Salvador, 2019).

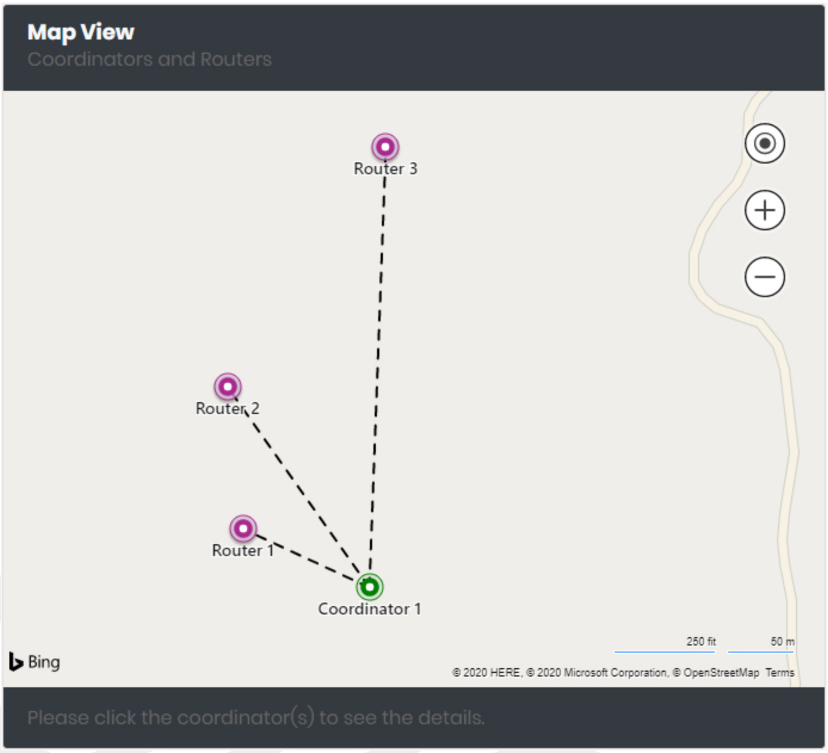


Figure 6.3 Viewing coordinators and routers on the map

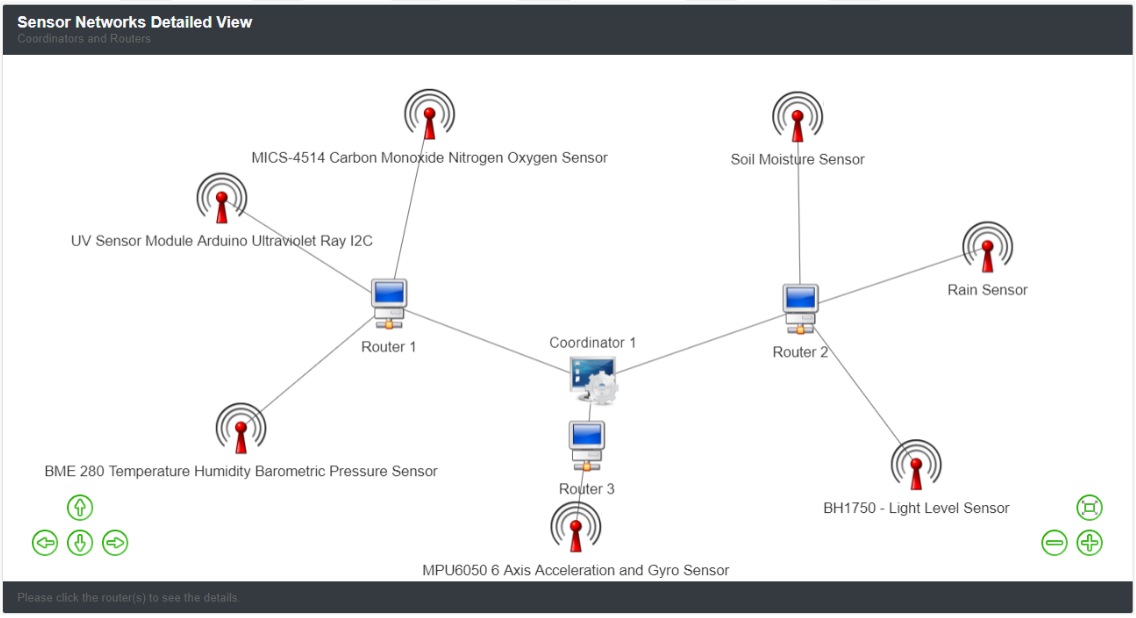


Figure 6.4 Detailed view of Wireless Sensor Networks

Raspberry PI, which is low-cost and single-board computer is used for interfacing the coordinator device through serial ports to receive data assembled from routers. Raspbian which is a kind of operating system based on Debian developed for Raspberry Pi might be freely accessed and is downloadable by any users. In addition, Raspbian is commonly preferred operating system option for Raspberry Pi computers and freely available community project under active development. Therefore, it has been considered as the suitable operating system option that will run on the Raspberry Pi computer which plays the manager role of established WSN. Furthermore, a Python program was developed to read raw sensor data gathered from coordinator device through serial port. This program is also responsible to store raw data into relational database using RESTful web service operations.

XBee module which supports multiple protocols, is highly configurable, and is small radio frequency (RF) devices that transmit and receive data over air using radio signals (DIGI, 2018), was used to transmit data wirelessly between routers and coordinator. Each XBee devices are configured as API mode within the built WSN. There exist several reasons behind preferring to use API mode. API operating mode eases managing transmission of data towards multiple destinations. Each received data contains the address of the sender devices. This mode provides advanced Zigbee addressing, advanced networking diagnostics, and remote configuration (DIGI, 2020). XBee devices which are configured as API operating mode, uses API frame illustrated in Figure 6.5 to transmit data from routers towards coordinator. After router devices measured environmental data through their sensors, they transform the sensor measurement values to API data frame using relevant Arduino libraries.

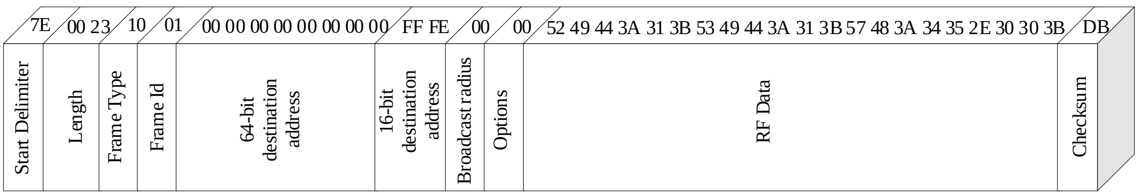


Figure 6.5 API data frame

The received data through coordinator are decomposed into several pieces to distinguish the router, sensor, and sensor measurement value. In other words, the purpose of this operation is to access RF data which are part of API frame to extract key-value pairs. After key-value pairs are extracted from RF data, it is necessary to match sensor to its measurement unit using sensor information stored in database. This matching process is performed by retrieving relevant data from database using REST web services. The reason for applying this method is to keep the total number of bytes included in the API frame's data filed small. Considering the sensors of WSN, the following measurement units are matched to relevant sensor measurement data types; lux for light intensity, °C for temperature, % for humidity, Pa for pressure, m for altitude, °/s for slope, nm for ultraviolet, ppm for carbon monoxide concentration, and ppb for nitrogen dioxide concentration.

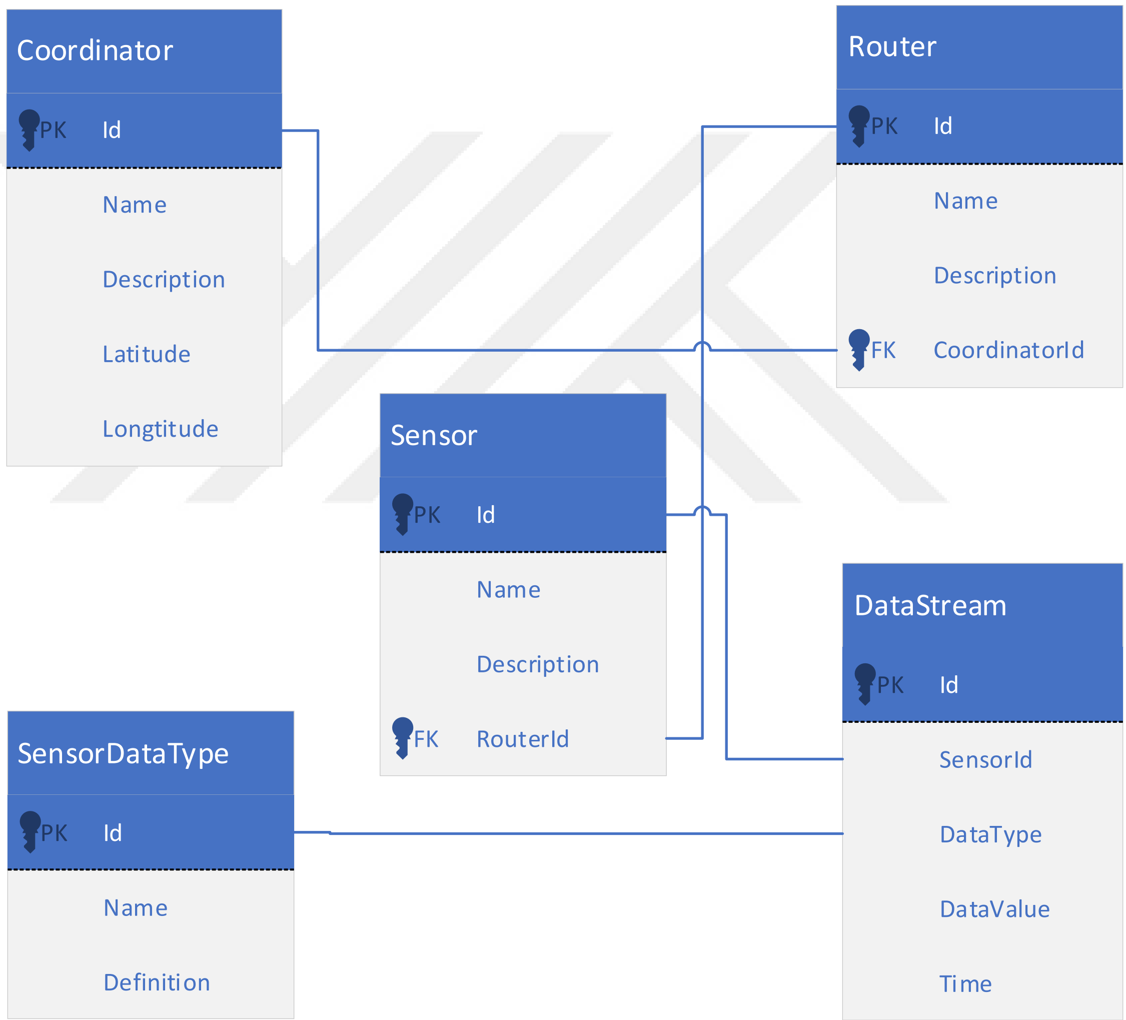


Figure 6.6 Data models created by OME

The model creation needs to be undertaken for decomposed and cleaned data using a programming language according to proposed approach. C# programming language is used to create objects which are representing in Figure 6.6. It is also major programming language which is used to develop web services and open data platform. There exist three different storage options recommended by DSE. MS SQL Server 2016, which was a relational database management system, is used to store data.

Windows Communication Foundation (WCF), which is a framework for building service-oriented applications, was used to develop XML and RESTful web services. dotNetRDF, which is a common .NET API for working with RDF triple stores was used to parse, manage, query, and write RDF datasets (dotNetRDF, 2020). ASP.NET Web API framework which provides an easy and secure way for building HTTP based services consumed by a wide range of software applications was used to develop the Web APIs.

6.7 Providing semantic interoperability between IoT devices and open data platforms using agricultural trait dictionaries

This part of the section will examine how to integrate sensor data using an agricultural trait dictionary. This section will also show how to create mapping rules for linking sensor data types with relevant ontology classes by storing in RDF file.

Hazelnut Trait Ontology is developed to share a common vocabulary and to provide an international format for standardizing general understanding with respect to hazelnut. Another purpose of creating an ontology regarding hazelnut is to provide a generally accepted common language for hazelnut. It is a generally complicated process to gather data concerning a specific domain. This ontology is created to help stakeholders while deciding which attributes should be defined within the gathered data as well. Hazelnut Trait Ontology is created to contribute to facilitating data storage, data retrieval, and data exchange in a rapid, reliable, and proper way by publishing accurate metadata with different types of services. As mentioned before the most general concept of Hazelnut Trait Ontology is Descriptor; and it has five types of general top-level concepts: Passport, Characterization, Environment and Site, Evaluation and Management.

The classes which might be mapped the relevant sensor measurement data types are sub-classes of :EnvironmentAndSite class which describes the environmental and site-specific parameters with respect to hazelnut within the Hazelnut Trait Ontology. After sensor data stream are decomposed as key-value pairs and stored into database in compliance with the object models which are created by OME, there need to create mapping rules using the relevant UI though open data platform. The open data platform uses these rules to map the sensor measurement data types with the relevant ontology classes to integrate and annotate data semantically. Varied data files mentioned in previous part of the section are generated by linking the stored datasets to relevant classes of agricultural ontology though using these mapping rules, thereby domain-specific datasets are built. Considering the presence of irrelevant sensor measurement data types within any WSN, it is reasonable to use these mapping rules for combining data from different sources into a single and unified view. The established WSN which was introduced within the previous section of this study uses seven sensor measurement data types corresponding the relevant Hazelnut Trait Ontology classes. Table 6.3 presents sensor measurement data types, abbreviations, data units and class axioms, which are utilized while creating mapping rules.

Table 6.3 Mapping ontology classes to relevant sensors' measurement types

Measurement	Abbreviation	Data Unit	Class Axiom
Light Intensity	LI	lux	:Light rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite.
Weather Temperature	WT	°C	:Temperature rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite.
Weather Humidity	WH	%	:RelativeHumidity rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite.
Pressure	PR	Pa	:AtmosphericPressure rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite.
Altitude	AL	m	:ElevationOfCollectingSite rdf:type owl:Class ; rdfs:subClassOf :CollectingDescriptor .
Gyro measurement range	GR	°/s	:Slope rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment .
Rain Rate	RR	-	:Rainfall rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite.
Soil Moisture	SM	-	:SoilMoisture rdf:type owl:Class ; rdfs:subClassOf :SiteEnvironment .
Ultraviolet	UV	nm	:Ultraviolet rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite.
Carbon monoxide	CO	ppm	:CarbonmonoxideConcentration rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite.
Nitrogen dioxide	NO_2	ppb	: NitrogendioxideConcentration rdf:type owl:Class ; rdfs:subClassOf :ClimateOfTheSite.

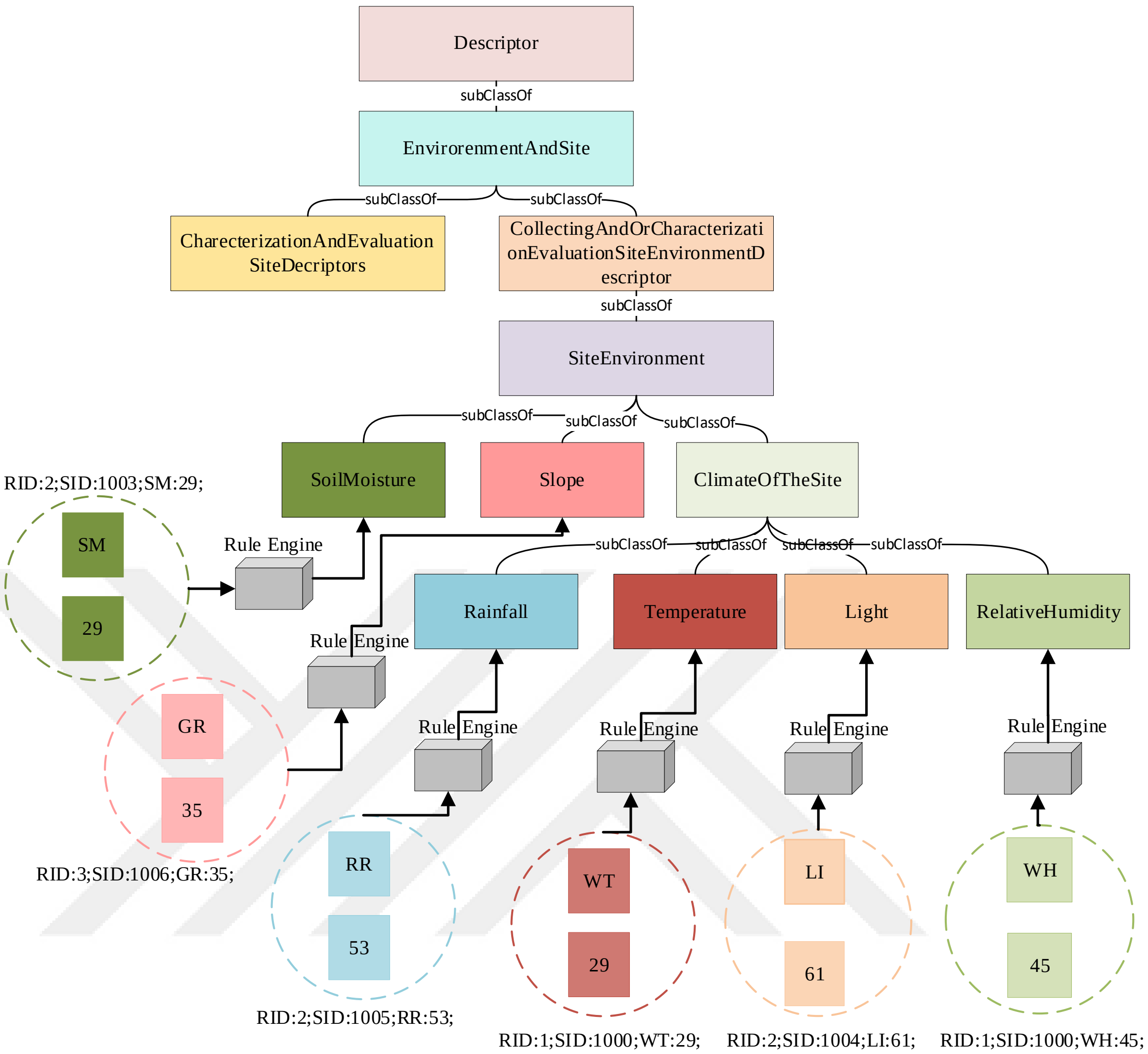


Figure 6.7 Mapping sensor measurement data type to ontology classes

Measurement values of mapped data types can be viewed using open data platform and they will be integrated with hazelnut datasets. As shown in Figure 6.7, decomposed sensor data streams are mapped with the relevant Hazelnut Trait Ontology classes by rule engine using mapping rules. One of the rule examples, which represents mapping the temperature measurement data type with :Temperature class of Hazelnut Trait Ontology, is illustrated in Table 6.4. It is apparent from this example that rules use URI reference to identify resources in datasets stored in RDF files. The base URI of Hazelnut Trait Ontology is *http://www.opendatainagriculture.com/ontologies/hazelnutontology*. The URI reference should be the same with the URI reference of the relevant ontology classes to make link between the sensor information and class definition within the mapping rule file.

Table 6.4 Mapping rule example

```
  <rdf:Description
rdf:about="http://www.opendatainagriculture.com/ontologies/hazelnutontology#Temperature">
    <mapping:ClassName>Temperature</mapping:ClassName>
    <mapping:DataUnit>°C</mapping:DataUnit>
    <mapping:SensorId>1000</mapping:SensorId>
    <mapping:SensorTypeName>BME 280 Temperature Humidity Barometric Pressure
Sensor</mapping:SensorTypeName>
    <mapping:TypeDef>WT</mapping:TypeDef>1
    <mapping:TypeId>2</mapping:TypeId>
    <mapping:TypeName>Weather Temperature</mapping:TypeName>
  </rdf:Description>
```

While linking sensor measurement data types with Hazelnut Trait Ontology classes, SPARQL query statements are utilized to retrieve each attribute of sensor data from mapping rule file. The SPARQL query statement representing in Table 6.5 retrieves base ontology URI with :Temperature class, sensor data type URI, and the value of Id attribute of the sensor data type from mapping rule RDF file. This query statement is executed by rule engine of open data platform and viewed as RDF triple which contains three components named subject, predicate, and object, respectively. In the example, subject, predicate, and object are :Temperature class, sensor data type, and the value of Id attribute of the sensor data type, respectively.

Table 6.5 SPARQL statement to retrieve sensor measurement data type from mapping rule file according to its “TypeId” attribute

```
SELECT   ?subject ?predicate  ?object
WHERE {
            ?subject ?predicate  ?object.
            FILTER(?object = "2")
           }
```

Subject	http://www.opendatainagriculture.com/ontologies/hazelnutontology#Temperature
Predicate	http://www.opendatainagriculture.com/sensors#TypeId
Object	2

When the mapping procedure is completed by data interchange engine of proposed approach, any stakeholder could create, export, and publish datasets with respect to agricultural product which is hazelnut in this case, in varied formats such as XML, JSON, RDF/XML, RDF/JSON, N-Triples, Turtle, Notation 3, HTML, Excel spreadsheet, and CSV. Figure A.7 illustrates how to export these datasets from the Open Data Platform. They might fulfil the procedures for discovering and analyzing using datasets stored in these files.

The open data platform improves the quality of data integration by applying inference rules on the traits which are mapped with the relevant sensor measurements. Furthermore, the inferred knowledge of a trait helps making better decisions concerning agricultural production for stakeholders who have lack of much information concerning to perform PA activities. For instance, the “Topography” class of Hazelnut Trait Dictionary might have the rules which are generated by domain expert represented in Table 6.6. According to the inferences provided by these rules any stakeholder might easily identify whether the topography of an agricultural land surface is flat, hilly, or mountainous.

Table 6.6 Rules for Topography Trait

Rule 1	Topography(?x) ∧ hasPercentegeValue(?x, ?y) ∧ swrlb:greaterThanOrEqual(?y, "0") ∧ swrlb:lessThanOrEqual(?y, "0.5") → hasTopography(?x, Flat)
Rule 2	Topography(?x) ∧ hasPercentegeValue(?x, ?y) ∧ swrlb:greaterThanOrEqual(?y, 0.6) ∧ swrlb:lessThanOrEqual(?y, 2.9) → hasTopography(?x, AlmostFlat)
Rule 3	Topography(?x) ∧ hasPercentegeValue(?x, ?y) ∧ swrlb:greaterThanOrEqual(?y, 3.0) ∧ swrlb:lessThanOrEqual(?y, 5.9) → hasTopography(?x, GentlyUndulating)
Rule 4	Topography(?x) ∧ hasPercentegeValue(?x, ?y) ∧ swrlb:greaterThanOrEqual(?y, 6.0) ∧ swrlb:lessThanOrEqual(?y, 10.9) → hasTopography(?x, Undulating)
Rule 5	Topography(?x) ∧ hasPercentegeValue(?x, ?y) ∧ swrlb:greaterThanOrEqual(?y, 11.0) ∧ swrlb:lessThanOrEqual(?y, 15.9) → hasTopography(?x, Rolling)
Rule 6	Topography(?x) ∧ hasPercentegeValue(?x, ?y) ∧ swrlb:greaterThanOrEqual(?y, 16.0) ∧ swrlb:lessThanOrEqual(?y, 30.0) → hasTopography(?x, Hilly)
Rule 7	Topography(?x) ∧ hasPercentegeValue(?x, ?y) ∧ swrlb:greaterThanOrEqual(?y, 31.00) ∧ Elevation(?x) ∧ hasValue(?x, ?z) ∧ swrlb:lessThanOrEqual(?z, 300)→ hasTopography(?x, SteeplyDissected)
Rule 8	Topography(?x) ∧ hasPercentegeValue(?x, ?y) ∧ swrlb:greaterThanOrEqual(?y, 31.00) ∧ Elevation(?x) ∧ hasValue(?x, ?z) ∧ swrlb:greaterThanOrEqual(?z, 301) → hasTopography(?x, Mountainous)

6.8 Evaluation of the Tools of Open Data Platform Used for Implementing the Proposed Approach

The usability of the tools of open data platform used for implementing the proposed data integration approach have been evaluated in terms of efficiency, affect, helpfulness, control, learnability, and usability by twenty-seven respondents-15 males and 12 females-with different levels of software skills and technical knowledge. The respondents are representative of the software developers who are experienced with developing web-based software platform and non-technical users who can use any web-based software platforms easily. They are also representatives of the IT specialists and researchers from agricultural domain, with a high-level education. Respondents average age is 29.44. Table 6.7 provides an overview of the respondents' profile.

Table 6.7. Respondents' Profiles

Age Range	**20-25**	**26-30**	**31-35**	**36-40**	**41-45**
Number of male respondents	7	3	2	0	3
Number of female respondents	4	3	1	2	2
Total	11	6	3	2	5

In this evaluation process, the following modules have been evaluated:

(i). "Ontology Viewing and Loading" represented on Figure A.1;

(ii). "Selecting Individuals of Traits for Inference Rules Creation" represented on Figure A.2;

(iii). "Creating Rules Definitions for Individuals" represented on Figure A.3;

(iv). "Creating Mapping Rules" represented on Figure A.4;

(v). "Wireless Sensor Networks Management" represented on Figure A.5;

(vi). "Charting and Managing Mapped Sensor Measurements" represented on Figure A.6.

The following tasks were assigned to each respondent to complete:

- register to the platform;
- login to the platform;
- access the UI of ontology viewing;
- access the UI of creation of mapping rules;
- select the individuals of traits for creating inference rules;
- access the UI of creating rules definitions for selected individuals;
- save defined rules for each selected individual;
- access the UI of WSN lists, select the relevant WSN and access the list of its routers and coordinators ;
- select any sensor of router and access the UI of visualizing collected data on the charts;
- access the UI of listing mapped sensor data.

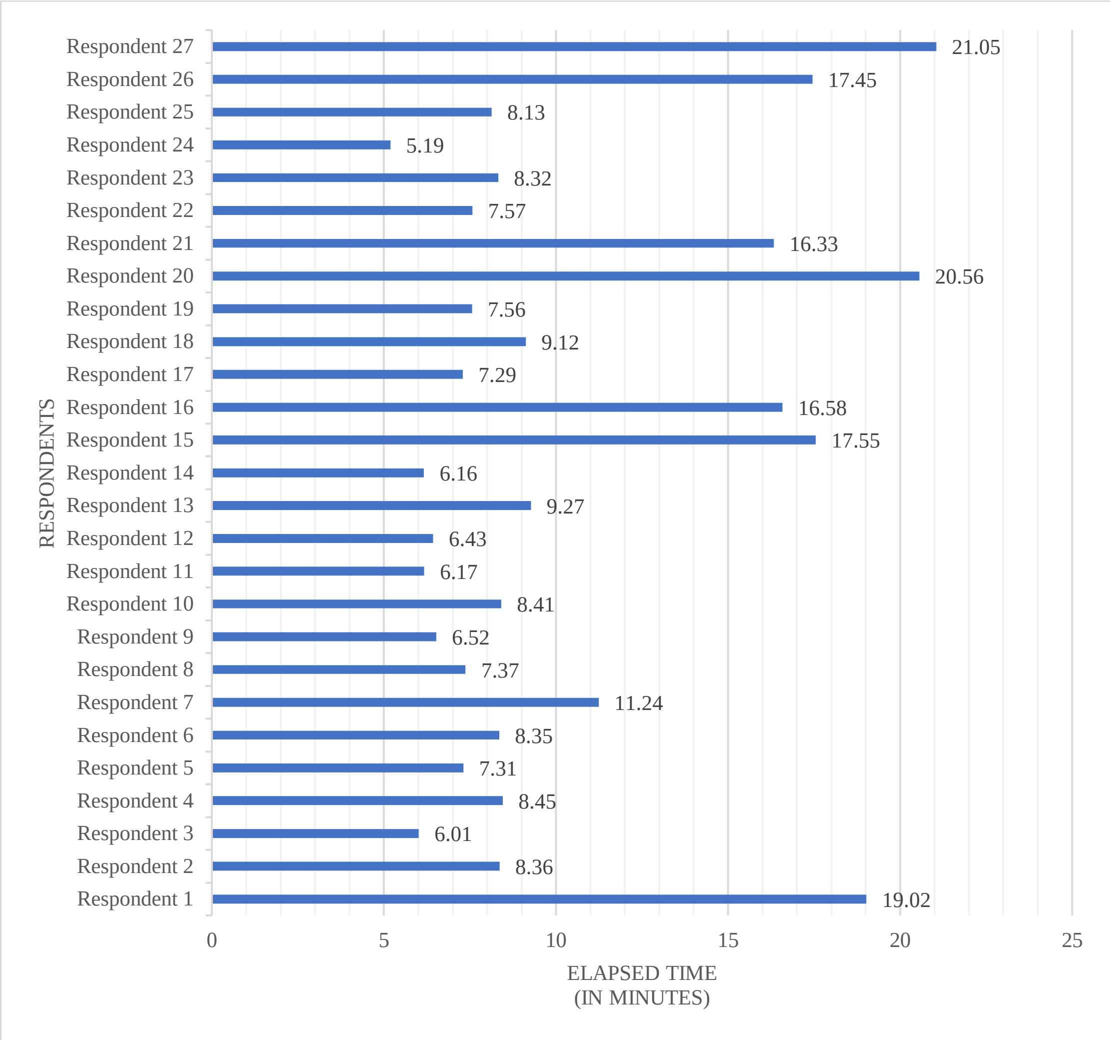


Figure 6.8 Elapsed time of each respondent for completing the tasks

Each respondent completed all assigned tasks to him/her successfully. While each respondent was performing the tasks, he/she recorded the elapsed time. Figure 6.8 shows the time spent by each respondent in completing her/his tasks. From Figure 6.8 one concludes that the average elapsed time is about 10.44 minutes. Considering the results, the most spent time for completing entire tasks is 21.05 minutes and the least spent time is 5.19 minutes.

The Software Usability Measurement Inventory (SUMI), which is a rigorously tested and proven method of measuring software quality from the end user's point of view, has been used to measure the usability of the open data platform. Figure 6.9 reveals that there have been satisfactory results of the entire scales. It also shows the range of the 95% confidence interval of the means of the scales. Table C.2 represents the SUMI user records for the relevant modules of the Open Data Platform.

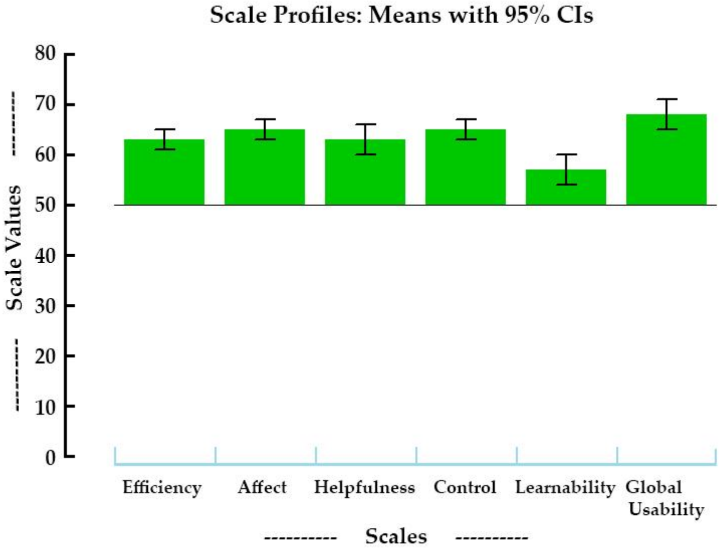


Figure 6.9 Graphical Summary of Scales

The SUMI scales are statistically adjusted so that the population mean in its database is 50. The range of SUMI scores goes from 73 to 10. SUMI scores exhibit a slight positive skew. Each scale is computed by a process of weighting and averaging of SUMI items. The mean, standard deviation, median, interquartile range (IQR), minimum, and maximum calculations results of each SUMI scale are presented in Table 6.8. As can be very clearly illustrated in Table 6.8, the developed open data platform has satisfactory result for each SUMI scale.

Table 6.8. Technical Information about SUMI Scales

	Mean	**Standard deviation**	**Median**	**Interquartile range (IQR)**	**Minimum**	**Maximum**
Global Usability	67.67	7.00	68.0	5.0	45	75
Efficiency	63.33	4.39	64.0	7.0	48	69
Affect	64.52	4.60	65.0	8.0	54	70
Helpfulness	63.07	8.71	65.0	8.0	38	72
Controlability	64.81	5.97	66.0	1.0	48	74
Learnability	56.78	7.30	57.0	9.0	40	69

Figure 6.10 shows the means and standard deviations for each of the SUMI scales and for the Global Usability scale of the sample being analyzed. The cross in the center of the ring shows the location of the mean, the bars are extended by one standard deviation on each side of the mean.

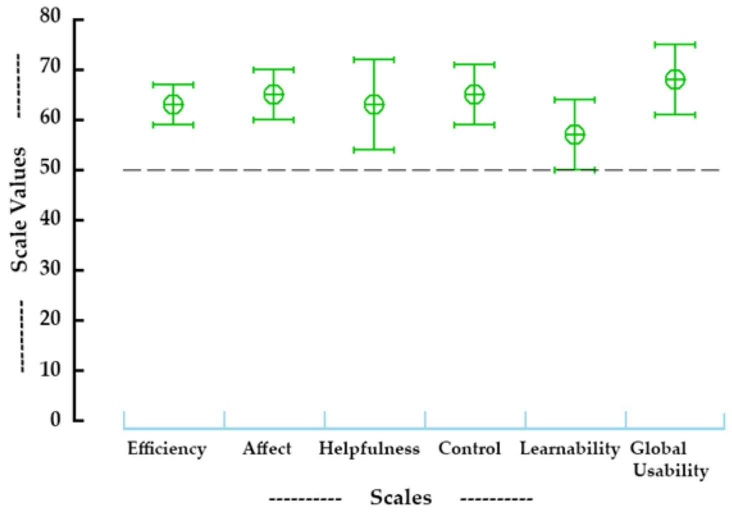


Figure 6.10 SUMI Scales Profiles: Means with Standard Deviations

Global usability of the open data platform, which means general feeling of satisfaction concerning the experiences of the users with the software platform has been calculated as 67.67. This result is quite reasonable in terms of usability score. The calculation results of efficiency, affect, helpfulness, controllability, and learnability are 63.33, 64.52, 63.07, 64.81, 56.78, respectively. Considering the scores of the SUMI scales are calculated in the range between 10 and 73, the calculated scores for open data platform are seen acceptable.

There are four different respondent profiles recommended by SUMI in terms of software skills and knowledge. Table 6.9 shows the profiles of respondents who were attended to evaluate the open data platform. Furthermore, it presents evaluation results of each scales obtained from respondents in different profiles.

Table 6.9 Respondents' Profiles in terms of Software Skills and Technical Knowledge

	n	Usability	Efficiency	Affect	Helpfulness	Control	Learnability
Very experienced and technical	19	69	64.4	63.9	64.9	65.6	59.1
I am experienced but not technical	2	60.5	52.5	63.5	54	57	51
I can cope with most software	6	65.8	63.5	66.8	60.3	64.8	51.5
I find most software difficult to use	0	-	-	-	-	-	-

The respondents have been asked to determine the importance of the evaluated open data platform with the following question: "How important for you is the kind of software you have just been rating?". Six respondents have noted that the open data platform is "important". On the other hand, the open data platform has been determined as extremely important by twenty-one respondents. Table 6.10 represents the calculated scores for each scale belonged to the two groups of respondents who described the open data platform as important and extremely important.

Table 6.10 Importance of the Platform for Respondents

	n	Usability	Efficiency	Affect	Helpfulness	Control	Learnability
Extremely important	21	67.9	63.7	64	63.2	65.1	57.9
Important	6	66.8	62	66.2	62.5	63.8	53
Not very important	0	-	-	-	-	-	-
Not important at all	0	-	-	-	-	-	-

As mentioned above the results of entire scales which have been used to measure the quality of the open data platform are quite reasonable. On the other hand, some enhancements should be carried out to increase the capabilities of the open data platform. For instance, some respondents have noted that the class selection screen should be more flexible. Furthermore, the design of some user interfaces should be edited with the aim of enabling non-technical users to do their tasks in a quick and effective manner.

6.9 Limitations

In this chapter, an approach has been proposed with the aim of integrating data collected from IoT devices to agricultural open data platforms from the perspective of semantic and syntactic interoperability. Furthermore, an open data platform for agriculture has been developed based on the proposed approach. There is no limitation in terms of using the proposed approach for any open data platforms. Although the study has successfully demonstrated how to provide semantic and syntactic interoperability by developing an agricultural open data platform based on the proposed approach, it has not been possible to develop any software applications which use web services and APIs to provide meaningful information concerning agricultural products. On the other hand, only semantic and syntactic interoperability have been carried out within the scope of this study. However, it needs to be considered that other interoperability aspects should be examined in terms of improving the productivity of the agricultural open data platforms.

7. INCLUDED PAPERS AND CONTRIBUTIONS

7.1 Paper I

Ş. Aydın, U. Ünal and M. N. Aydın, "Open Data in Agriculture: Sustainable Model Development for Hazelnut farms using semantics," 2018 6th International Conference on Control Engineering & Information Technology (CEIT), Istanbul, Turkey, 2018, pp. 1-6.

7.1.1 Summary

Turkey accounts for 75% of the global hazelnut production and 70-75% of the exportation. Taking into account the socio-economic importance of hazelnut, the stakeholders of hazelnut domain still have problems such as availability, meaningful, accuracy of the hazelnut related data. Providing data to stakeholders is crucial for sustainable agricultural activities. This data should be freely available to everyone to use and republish. With the aforementioned reasons “Open Data” is an efficient way in Turkish Agriculture.

In this paper, we shall investigate the open data term and semantics in the context of hazelnut data management. In addition, a data processing model with regard to agricultural open data is proposed.

7.1.2 Contributions

- This is the initial work of the research.
- This paper reviewed the research conducted on semantic web technologies.
- This paper attempts to show that how to use semantic web technologies in the context of agriculture domain for particularly hazelnut production.
- The aim of this paper is to provide a conceptual, applicable, and extendable open data processing model in agriculture.
- The proposed agricultural open data processing model contains following components; types of agricultural data sources (four types of data sources; farmers, sensor, government, and market data), cloud hub (two sub-layers; data

management and web service layers), and users (five different users; farmers, researchers, businesses, analysts, and experts)

7.2 Paper II

AYDIN ŞAHİN, AYDIN MEHMET NAFİZ (2018). Multi-layered Open Data Processing Model for Hazelnut farms. 5th International Conference on Management Information Systems (IMISC 2018)

7.2.1 Summary

In recent years several projects that are supported by information and communications technologies (ICT) have been developed in the agricultural domain to promote more precise agricultural activities. These projects account for different kinds of key ICT terms such as internet of things (IoT), wireless sensors networks (WSN), cloud computing (CC) etc. These projects are used for different agricultural products and it is a well-known fact that they can be essential to perform precise agricultural activities for the relevant agricultural products. It is important to emphasize that the success of implementing these projects depends on the extent to which support various stakeholders by leveraging relevant data, which are gathered from different kinds of data sources. Agriculture domain has a great deal of stakeholders. These stakeholders need sophisticated data and appropriate intelligence for getting benefits to performing precise agricultural activities. We agreed with scholars that “Open Data” idea, which means accessing data which is published on web and available in a machine readable format is an appropriate way to get benefit for precise agriculture by relevant stakeholders.

In this paper, we shall investigate the open data term in an agriculture context and create an open data processing model. We also show viability of the proposed model by developing an ICT-based solution. Taking into account the socioeconomic importance of hazelnut for Turkey, the stakeholders of hazelnut domain still have problems such as availability, meaningful, accuracy of the hazelnut related data so we shall focus on hazelnut within the scope of this paper.

7.2.2 Contributions

- This paper is the extended version of Paper I.
- This paper reviewed the research conducted on open data term.
- This paper attempts to show that how to use IoT technologies in the context of agriculture domain for particularly hazelnut production.
- The aim of this paper is to provide a multi-layered open data processing model which has been developed in the context of agriculture.
- The proposed multi-layered agricultural open data processing model contains following components; types of agricultural data sources (five types of data sources; farmers, sensors, statistical data from government, market data, and other data sources); data processing layer (raw data storage, data pre-processing and processing); semantic annotation layer (agricultural product ontology and data interchange); data storage layer (graph databases, relational databases and database services); services layer (REST web services, SOAP web services, mobile services, analysis services, reporting services and SPARQL query services); applications layer (open data web platform, open data mobile platform and open data desktop platform); and lastly end users layer (six different kinds of users; farmers, researchers, businesses, analysts, experts and other users).

7.3 Paper III

S. AYDIN, M.N. AYDIN "A Sustainable Multi-layered Open Data Processing Model for Agriculture: IoT Based Case Study Using Semantic Web for Hazelnut Fields", Advances in Science, Technology and Engineering Systems Journal, vol. 5, no. 2, pp. 309-319 (2020).

7.3.1 Summary

In recent years, several projects which are supported by information and communications technologies (ICT) have been developed in the agricultural domain to promote more precise agricultural activities. These projects account for different kinds of key ICT terms such as internet of things (IoT), wireless sensors networks (WSN), cloud computing (CC). These projects are used for different agricultural products; and it is a well-known fact that they can be essential to perform precise agricultural activities for the relevant agricultural products. The implementation of these projects successfully depends on the extent to which various stakeholders provide support by leveraging relevant data, gathered from heterogenous data sources. Agriculture domain has a great number of stakeholders. These stakeholders need sophisticated data and appropriate intelligence to get benefits in order to perform precise agricultural activities. Authors agreed with scholars that "Open Data" idea, which means accessing data published on the web and available in a machine-readable format is an appropriate way to get benefits for precise agriculture by relevant stakeholders. In this paper, authors shall investigate the open data term in an agricultural context, create an open data processing model, and develop an IoT-based solution to gather environmental data from agricultural fields. Authors also show viability of the proposed model by developing an ICT-based solution. Considering the socioeconomic importance of hazelnut for Turkey, the stakeholders of hazelnut domain still have problems such as availability, meaningful, accuracy of the hazelnut related data. Therefore, authors shall focus on hazelnut within the scope of this paper.

7.3.2 Contributions

- This paper is the combined and extended versions of Paper I and Paper II.
- A multilayered open data processing model has been created, and focused on the hazelnut agricultural product.
- Even though this model focuses on a sustainable agricultural production lifecycle of hazelnut in this paper, it is convenient for all agricultural products as well.
- This model is created by considering data sources, data processing, semantic annotations, data storage, services, applications, and users.
- The proposed open data model which demonstrates how to process data obtained from heterogeneous agricultural data sources and designed as multi-layered includes varied layers such as types of data sources (IoT devices, statistical data from government, market data, farmers, and other data sources); storing and processing raw data; semantic annotation layer (agricultural product ontology,

other ontologies and data interchange engine); data storage layer (graph databases, relational databases, cloud databases, RDF and XML file-based storage options and database service); services layer (XML web services, REST web services, web APIs, mobile services, analysis services, reporting services and SPARQL query services); applications layer (unified web based data platform based on open formats, mobile applications and desktop applications); and lastly, end users layer.

- This paper also demonstrates how to store environmental data collected from IoT devices into the graph databases and RDF files.
- Lastly, this paper shows how to query IoT data that is stored RDF files by using SPARQL.

7.4 Paper IV

Sahin Aydin, Mehmet N. Aydin, Ontology-based data acquisition model development for agricultural open data platforms and implementation of OWL2MVC tool, Computers and Electronics in Agriculture, Volume 175, 2020, 105589, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2020.105589.

7.4.1 Summary

In the open data world, it is difficult to collect data in compliance with a certain data model that is of interest to different types of stakeholders within a domain like agriculture. Ontologies that provide broad vocabularies and metadata with respect to a given domain might be used to create various data models. We consider that while creating data acquisition forms to gather data related to an agricultural product, which is hazelnut in this study, from stakeholders of the relevant domain, the traits can be modeled as attributes of the data models. We propose a generic ontology-based data acquisition model to create data acquisition forms based on MVC design pattern, to publish and make use of on the agricultural open data platforms. We develop a tool called OWL2MVC that integrates the Hazelnut Trait Ontology, which illustrates the effectiveness of the proposed model for generating data acquisition forms. Because model creation is implemented in compliance with the selection of ontology classes, stakeholders; in other words, the users of OWL2MVC Tool could generate data acquisition forms quickly and independently.

OWLMVC Tool was evaluated in terms of usability by fifty-three respondents implementing the case-study scenario. Among others the findings show that the tool has satisfactory usability score overall and is promising to provide stakeholders with required support for agricultural open data platforms.

7.4.2 Contributions

- This paper reviews the literature conducted on generating data acquisition forms and infrastructure using ontologies and semantic web technologies.
- In this paper, the studies carried out in the context of creating data acquisition forms using ontologies and semantic web technologies have been compared with each other. By this means, it has been aimed to highlight the needs of requirements for the proposed model and developed tool.
- In this paper, a generic ontology-based data acquisition model has been proposed to create data acquisition forms based on model-view-controller (MVC) design pattern, to publish and make use of on the agricultural open data platforms.
- A tool called OWL2MVC that integrates the Hazelnut Trait Ontology has been developed to illustrate the effectiveness of the proposed model for generating data acquisition forms.
- In this paper, ASPNET Control Ontology has been created to represent the general vocabulary regarding web-based form controls; in other words, web form elements; and it consists of sixty-seven classes, sixty-six sub classes and two object properties.
- This paper also demonstrates how to store data which is collected using generated data acquisition forms into graph databases.
- Lastly, OWLMVC Tool has been evaluated in terms of usability by fifty-three respondents implementing the case-study scenario in this paper.

7.5 Paper V

Aydin, S.; Aydin, M.N. Semantic and Syntactic Interoperability for Agricultural Open-Data Platforms in the Context of IoT Using Crop-Specific Trait Ontologies. Appl. Sci. 2020, 10, 4460.

7.5.1 Summary

In recent years, Internet-of-Things (IoT) -based applications have been used in various domains such as health, industry, and agriculture. Considerable amount of data in diverse formats is collected from Wireless Sensor Networks (WSNs) integrated into IoT devices. Semantic interoperability of data gathered from IoT devices are generally being carried out using existing sensor ontologies. However, crop-specific trait ontologies, which include site-specific parameters concerning hazelnut as a particular agricultural product, can be used to make links between domain-specific variables and sensor measurement values as well. This research seeks to address how to use crop-specific trait ontology for linking site-specific parameters to sensor measurement values. A data integration approach for semantic and syntactic interoperability is proposed to achieve this objective; and an open data platform is developed, and its usability evaluated to justify the viability of the proposed approach. Furthermore, this research shows how to use the web services and APIs to carry out the syntactic interoperability of sensor data in agriculture domain.

7.5.2 Contribution

- This paper reviews the literature conducted on interoperability in IoT.
- This study set out to focus on semantic and syntactic interoperability in the context of IoT.
- This paper has discussed the reasons for using crop-specific trait ontologies with the aim of mapping its classes to sensor measurement values.
- The present paper has been designed to demonstrate how to export IoT data as linked open data using open standards and formats.
- This paper has argued that how to use a crop-specific trait ontology for linking site-specific parameters to sensor measurement values.
- An ontology-based data integration approach has been proposed to achieve this objective and an open data platform has been developed to justify the viability of the proposed approach in this paper.
- This research has shown how to utilize the web services and APIs to carry out the syntactic interoperability of sensor data in agriculture domain.

- This paper introduces the Open Data Platform which has been developed to store, process, visualize, and publish agricultural data.
- This paper also illustrates how to create rules for providing inferred knowledge of a trait helps making better decisions concerning agricultural production for stakeholders who have lack of much information concerning to perform PA activities.
- Lastly, usability of the tools of open data platform used for implementing the proposed ontology-based data integration approach have been evaluated in terms of efficiency, affect, helpfulness, control, learnability, and usability by twenty-seven respondents who have different levels of software skills and knowledge.

8. CONCLUSIONS

In the chapter three, we proposed a multi-layered open data processing system that consists of different layers and components. All layers and its components were explained; and an implementation regarding sensors was developed and summarized in the implementation section. We examined only gathering data from IoT devices in this implementation. However, there are four other aspects such as farmers, government and statistical data sources, market data and other data sources. These other aspects are worth discussing and examining in the future. Also, we tried to justify the suitability of the data life cycle in terms of the proposed multi-layered model by implementing the wireless sensor network system. We believe that this study enhances academic understanding of creating a multi-layered open data processing model. However, the life cycle of other data from other data sources should be examined and justified for viability. A number of limitations and promising research areas can be listed as : difficulty of gathering data from farmers (lack of knowledge of technology usage), defining the boundaries of gathering market data, determining the potential data sources, which is defined as “Other Data Sources” in heterogeneous data sources layer, and spreading the usage of applications mentioned in “Applications Layer”.

Chapter four contributes to better understanding and fulfilment of requirements for Hazelnut Trait Ontology. The arguments given within the parts of this chapter claim that existing ontologies are not suitable for hazelnut. However, the stakeholders of the agricultural domain need more sophisticated, publicly available, freely accessible, and in a machine-readable format data. This could be provided by using the power of semantic web technologies. Considering this power, we created Hazelnut Trait Ontology as initial work to build semantic annotation layer for open data processing model. A number of limitations of our study and areas for future research should be mentioned. For instance, ensuring ontology consistency, viability, reliability, difficulties in user and developers’ scenarios while evolving the process of the ontology might be a restriction and reluctance towards using this ontology. However, we submit that the contribution of our paper rests on knowledge engineering, semantic web, and building domain ontologies, we are not hazelnut domain experts. Thus, the contributions of hazelnut domain experts are crucial for future studies. A considerable amount of literature has been published on the evaluation of quality of ontologies. One of the main aims of this research is to assess the Hazelnut Trait Ontology by using the existing quality evaluation tools and methodologies. One question-based methodology and three different tools that measure similar metrics are used to evaluate Hazelnut Trait Ontology. Similar or distinct calculation results are obtained from these tools and methodology for Hazelnut Trait Ontology. These results are analyzed within the parts of this chapter. Analyzing the calculation results gives us a general opinion rather than making a certain decision regarding the quality of Hazelnut Trait Ontology. It is worth bearing in mind that including domain experts to evaluation processes of ontologies might provide a better understanding in compliance with their quality. In this chapter, domain experts are not included in the evaluation stage of Hazelnut Trait Ontology. However, our focus just is on introducing Hazelnut Trait Ontology, expressing the requirements of creating it, and evaluating the quality of it by using existing tools and methodologies. Therefore, handling the evaluation of the quality of Hazelnut Trait Ontology under the supervision of domain experts would provide important insights to spread its usage into a wide range. Further research should be done to investigate a new approach for evaluating the quality of ontologies including the domain experts.

Within the scope of chapter five, we proposed a generic ontology-based data acquisition form model which is constructed on MVC design pattern and developed a tool named OWL2MVC to create web forms dynamically. We used this tool to generate particular web forms in agriculture domain using Hazelnut Trait Ontology, which is a kind of trait dictionary. By this means, we have demonstrated how to use our model with a real-life scenario. While a data acquisition form is generated, one needs to make some changes at the back-end code of the project in case the data model has changed. In addition, the design of new data acquisition form depends on the requirements of users. However, our tool provides a flexible and easily useable form generation environment for any stakeholder of agriculture domain. Because model creation is implemented in compliance with the selection of ontology classes, stakeholders; in other words, the users of OWL2MVC Tool could generate data acquisition forms quickly and independently. It should be noted that our tool is a domain expert-centered tool, which means that any domain expert could upload a particular agricultural ontology (trait dictionary) on the platform and generate data acquisition forms in accordance with the requirements. OWLMVC Tool was evaluated in terms of usability by fifty-three respondents implementing the case-study scenario. The Global Usability Score (GUS) is 74.2 in compliance with this evaluation. This result shows us that, it has satisfactory usability score in terms of users' point of view. Considering the results of other scales of SUMI, OWL2MVC Tool has exceptional scores for each one of them, but learnability scale's score needs to increase a little bit. Nevertheless, the learnability score of OWL2MVC is 60%. This means that it has satisfactory learnability score as it is above the average value (50).

The chapter six sets out to propose an approach to integrating data collected from IoT devices to agricultural open data platforms from the perspective of semantic and syntactic interoperability. The proposed data integration approach has enhanced an understanding of data integration lifecycle for agricultural IoT applications in the context of open data platforms. The general tendency to provide data integration for data gathered from IoT devices is to use existing geospatial ontologies. However, the study has demonstrated (see chapter six), for the first time, that a crop-specific trait ontology by constructing mapping rules has been used to annotate the environmental data for a particular agricultural product with the aim of achieving semantic interoperability. The most obvious finding to emerge from this study is that semantic interoperability for data collected from IoT devices can be carried out using a crop-specific trait ontology by constructing mapping rules. Another important finding is to demonstrate how to use web services and APIs to provide syntactical interoperability for data gathered from IoT devices using crop-specific trait ontologies.

The research demonstrated the value of the proposed approach along with its implementation of software applications (see chapter six). Firstly, the open data platform which is developed based on the proposed approach enables agricultural domain stakeholders export linked open data in varied open formats or standards. It provides a visualization infrastructure for IoT devices and their data. Thus, stakeholders could access, and export data gathered from IoT devices belonging to a particular wireless sensor network established for the relevant agricultural product. Secondly, Web Services and APIs allow stakeholders to share data about a particular agricultural product between different kinds of software applications. By this means, syntactical interoperability has been carried out through these services and APIs.

The open data platform has been evaluated in terms of usability considering five different scales such as efficiency, affect, helpfulness, control, and learnability by using SUMI questionnaire. According to the usability results, the developed platform has satisfactory scores for each one of them. Furthermore, the score of global usability scale is quite reasonable for the open data platform.

The implementation part of the study focused on hazelnut agricultural product. However, the proposed approach is appropriate for different types of crop-specific trait ontologies, which are created using OWL and includes site-specific variables. The present study has only examined semantic and syntactic interoperability. However, further research might investigate other interoperability layers for open data platforms. One might be interested in making use of web services and APIs from mobile-based software applications to provide meaningful information for the relevant agricultural product and its production lifecycle.

REFERENCES

A. Gyrard, A. Zimmermann and A. Sheth, "Building IoT-Based Applications for Smart Cities: How Can Ontology Catalogs Help?," in IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3978-3990, Oct. 2018.

About AGROVOC Page. [online] Available at: http://aims.fao.org/standards/agrovoc/concept-scheme, last [Accessed 2019/01/15].

Agricultural Marketing. [online] Available at: https://en.wikipedia.org/wiki/Agricultural_marketing [Accessed 24.07.2018]

Agriculture and Food. [online] Available at: http://www.invest.gov.tr/en-US/sectors/Pages/Agriculture.aspx [Accessed 21.07.2018].

Ahlem Rhayem, Mohamed Ben Ahmed Mhiri, Mayssa Ben Salah, Faiez Gargouri, Ontology-based system for patient monitoring with connected objects, Procedia Computer Science, Volume 112, 2017, Pages 683-692, ISSN 1877-0509

Ahmad M. Nagib, Haitham S. Hamza, SIGHTED: A Framework for Semantic Integration of Heterogeneous Sensor Data on the Internet of Things, Procedia Computer Science, Volume 83, 2016, Pages 529-536, ISSN 1877-0509

Aisha Siddiqa, Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Mohsen Marjani, Shahabuddin Shamshirband, Abdullah Gani, Fariza Nasaruddin, A survey of big data management: Taxonomy and state-of-the-art, Journal of Network and Computer Applications, Volume 71, 2016, Pages 151-166, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2016.04.008.

Alani, Harith; Brewster, Christopher and Shadbolt, Nigel (2006). Ranking ontologies with AKTiveRank. In: The Semantic Web - ISWC 2006, Lecture Notes in Computer Science, Springer.

Alexandre Robin. OGC SWE Common Data Model Encoding Standard. Available online: http://portal.opengeospatial.org/files/?artifact_id=41157 (accessed on 22.03.2020).

Allemang, D., Hendler, J.: Semantic Web for the Working Ontologist.1st edn. Morgan Kaufman Publications, United States (2008).

Amann, Bernd & Scholl, Michel. (1996). Gram: A Graph Data Model and Query Language., Pages 201-211

Arnaud, Elizabeth, Laurel Cooper, Rosemary Shrestha, Naama Menda, Rex T. Nelson, Luca Matteis, Milko Skofic, Ruth Bastow, Pankaj Jaiswal, Lukas A. Mueller and Graham McLaren. “Towards a Reference Plant Trait Ontology for Modeling Knowledge of Plant Traits and Phenotypes.” KEOD (2012).

Bandeira, J., Bittencourt, I.I., Espinheira, P., & Isotani, S. (2016). FOCA: A Methodology for Ontology Evaluation. ArXiv, abs/1612.03353.

Berners-Lee, T., 2006, Linked Data, https://www.w3.org/DesignIssues/LinkedData.html [Accessed 23.07.2018].

Bertha Mazon-Olivo, Dixys Hernández-Rojas, José Maza-Salinas, Alberto Pan, Rules engine and complex event processor in the context of internet of things for precision agriculture, Computers and Electronics in Agriculture, Volume 154, 2018, Pages 347-360, ISSN 0168-1699

Bevan, Nigel. (1995). Measuring usability as quality of use. Software Quality Control. 4. 115-130. 10.1007/BF00402715.

Burton-Jones, A., Sorey, V. C., Sugumaran, V., & Ahluwalia, P. (2005). A semiotic metrics suite for assessing the quality of ontologies. Data & Knowledge Engineering, 55 , 84–102.

C. Bizer, T. Heath, T. Berners-Lee, Linked data—the story so far, Int. J. Semant.Web Inform. Syst. 5 (3) (2009) 1–22.

C. E. Kaed, I. Khan, A. Van Den Berg, H. Hossayni and C. Saint-Marcel, "SRE: Semantic Rules Engine for the Industrial Internet-Of-Things Gateways," in IEEE Transactions on Industrial Informatics, vol. 14, no. 2, pp. 715-724, Feb. 2018.

C. Hsu and R. Tsaih, "Applying design science research to innovations at ICT-enabled service," Proceedings of International Conference on Information, Communication Technology and System (ICTS) 2014, Surabaya, 2014, pp. 251-256, doi: 10.1109/ICTS.2014.7010592.

C. J. Matheus, D. Tribble, M. M. Kokar, M. G. Ceruti, and S. C. McGirr. Towards a formal pedigree ontology for level-one sensor fusion. In 10th International Command & Control Research and Technology Symposium, 2005.

Carla Bonina, Ben Eaton, Cultivating open government data platform ecosystems through governance: Lessons from Buenos Aires, Mexico City and Montevideo, Government Information Quarterly, Volume 37, Issue 3, 2020, 101479, ISSN 0740-624X, https://doi.org/10.1016/j.giq.2020.101479.

Carolan, L., Smithy, F., Protonotarios, V., Schaap, B., Broad, E., Hardinges, J., Gerry, W., 2015, How can we improve agriculture, food and nutrition with open data?, Open Data Institute

Charalampos C. Charalampidis, Euclid A. Keramopoulos, Semantic Web user interfaces – A model and a review, Data & Knowledge Engineering, Volume 115, 2018, Pages 214-227, ISSN 0169-023X, https://doi.org/10.1016/j.datak.2018.04.003.

Charalampos C. Charalampidis, Euclid A. Keramopoulos, Semantic Web user interfaces – A model and a review, Data & Knowledge Engineering, Volume 115, 2018, Pages 214-227, ISSN 0169-023X

Christoph Pflügler, Maximilian Schreieck, Gabriel Hernandez, Manuel Wiesche, Helmut Krcmar, A Concept for the Architecture of an Open Platform for Modular Mobility Services in the Smart City, Transportation Research Procedia, Volume 19, 2016, Pages 199-206, ISSN 2352-1465, https://doi.org/10.1016/j.trpro.2016.12.080.

Corcho O., Fernández-López M., Gómez-Pérez A., López-Cima A.: Building Legal Ontologies with METHONTOLOGY and WebODE. In: Benjamins V.R., Casanovas P., Breuker J., Gangemi A. (eds) Law and the Semantic Web. Lecture Notes in Computer Science, vol 3369. Springer, Berlin, Heidelberg, J., Gangemi A. (eds) Law and the Semantic Web. Lecture Notes in Computer Science, vol 3369. Springer, Berlin, Heidelberg (2005).

Cristani, M., Cuel, R.: A Survey on Ontology Creation Methodologies, Journal on Semantic Web & Information Systems, 1(2), 48-68, April-June (2005).

Cronbach, L.J. Coefficient alpha and the internal structure of tests. Psychometrika 16, 297–334 (1951). https://doi.org/10.1007/BF02310555

Crop Ontology Curation Tool, Hom the Project: http://www.cropontology.org/about, last accessed 2019/01/16.

Crop Ontology Curation Tool, Home: http://www.cropontology.org/, last accessed 2019/01/16.

D. Russomanno, C. Kothari, and O. Thomas. Building a sensor ontology: a practical approach leveraging ISO and OGC models. In 2005 International Conference on Artificial Intelligence (vol 2), 2005.

d'Aquin, M., Schlicht, A., Stuckenschmidt, H., Sabou, M.: Criteria and evaluation for ontology modularization techniques. In: Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization, Lecture Notes in Computer Science, vol. 5445, pp. 67–89. Springer (2009)

Daraio, C., Lenzerini, M., Leporelli, C. et al. The advantages of an Ontology-Based Data Management approach: openness, interoperability and data quality. Scientometrics 108, 441–455 (2016). https://icproxy.khas.edu.tr:4652/10.1007/s11192-016-1913-6

David Beckett, Tim Berners-Lee, Eric Prud'hommeaux, Gavin Carothers. RDF 1.1 Turtle. Available online: https://www.w3.org/TR/turtle/ (accessed on 22.03.2020).

David Beckett. RDF 1.1 N-Triples. Available online: https://www.w3.org/TR/n-triples/ (accessed on 22.03.2020).

Dean Allemang and James Hendler. 2011. Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL (2 ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Dedi Iskandar Inan, Ghassan Beydoun, Disaster Knowledge Management Analysis Framework Utilizing Agent-Based Models: Design Science Research Approach, Procedia Computer Science, Volume 124, 2017, Pages 116-124, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2017.12.137.

Denny Vrandečić, Sofia Pinto, Christoph Tempich, York Sure: "The DILIGENT knowledge processes", Journal of Knowledge Management, Vol. 9 Issue: 5, pp.85-96, (2005).

DIGI International Inc., XBee/XBee-PRO S2C Zigbee RF Module User Guide, 2018

Dimitris S. Paraforos, Vangelis Vassiliadis, Dietrich Kortenbruck, Kostas Stamkopoulos, Vasileios Ziogas, Athanasios A. Sapounas, Hans W. Griepentrog, Multi-level automation of farm management information systems, Computers and Electronics in Agriculture, Volume 142, Part B, 2017, Pages 504-514, ISSN 0168- 699

Dobrev P., Kalaydjiev O., Angelova G. (2007) From Conceptual Structures to Semantic Interoperability of Content. In: Priss U., Polovina S., Hill R. (eds) Conceptual Structures: Knowledge Architectures for Smart Applications. ICCS 2007. Lecture Notes in Computer Science, vol 4604. Springer, Berlin, Heidelberg

dotNetRDF: An Open Source .NET Library for RDF. Available online: https://www.dotnetrdf.org/ (accessed on 26.03.2020)

Duque-Ramos, Astrid & Fernandez-Breis, Jesualdo & Stevens, Robert & Aussenac-Gilles, Nathalie. (2011). OQuaRE: A SQuaRE-based approach for evaluating the quality of ontologies. Journal of Research and Practice in Information Technology. 43. 159-176.

E. Begoli and J. Horey, "Design Principles for Effective Knowledge Discovery from Big Data," 2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, Helsinki, 2012, pp. 215-218, https://doi.org/10.1109/WICSA-ECSA.212.32.

Eric Costa, António Lucas Soares, Jorge Pinho de Sousa, Industrial business associations improving the internationalisation of SMEs with digital platforms: A design science research approach, International Journal of Information Management, Volume 53, 2020, 102070, ISSN 0268-4012, https://doi.org/10.1016/j.ijinfomgt.2020.102070.

Erik van Veenendaal. 2002. Low-cost usability testing. Software quality and software testing in internet times. Springer-Verlag, Berlin, Heidelberg, 153–164.

Eriksson, H., Puerta, A. R., & Musen, M. A. (1994). Generation of knowledge-acquisition tools from reusable domain ontologies. International Journal of Human-Computer Studies, 41(3), 425–453. doi: 10.1006/ijhc.1994.1067

F. Manola, E. Miller, RDF Primer, W3C Recommendation, World Wide Web consortium. Available from: <http://www.w3.org/TR/rdf-primer/> (February 10, 2004).

Farhan Ullah, Muhammad Asif Habib, Muhammad Farhan, Shehzad Khalid, Mehr Yahya Durrani, Sohail Jabbar, Semantic interoperability for big-data in heterogeneous IoT infrastructure for healthcare, Sustainable Cities and Society, Volume 34, 2017, Pages 90-96, ISSN 2210-6707

Fernandez-Lopez, M.: Overview of methodologies for building ontologies. In., Proceedings of IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends, in conjunction with the Sixteenth International Joint Conference on Artificial Intelligence, August, Stockholm, Sweden (1999).

Fındık Veriminde ve Üretiminde Temel Sorunlar. [online] Available at: http://www.findikihracati.com/findik/findik-veriminde-ve-uretiminde-temel-sorunlar.html [Accessed 21.07.2018].

Foughali Karim, Fathalah Karim, Ali frihida, Monitoring system using web of things in precision agriculture, Procedia Computer Science, Volume 110, 2017, Pages 402-409, ISSN 1877-0509

Fox, M., S. : The TOVE Project Towards a Common-Sense Model of the Enterprise, Conference: Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 5th International Conference, IEA/AIE - 92, Paderborn, Germany, June 9-12, 1992, Proceedings (1992)

Francesconi, Enrico & Montemagni, Simonetta & Peters, Wim & Tiscornia, Daniela : Integrating a Bottom–Up and Top–Down Methodology for Building Semantic Resources for the Multilingual Legal Domain. 95-121. 10.1007/978-3-642-12837-0_6 (2010).

Freeman, A. (2014). Pro ASP.NET MVC 5. Apress, Berkely, CA, USA.

Fu Zhang, Z.M. Ma, Weijun Li, Storing OWL ontologies in object-oriented databases, Knowledge-Based Systems, Volume 76, 2015, Pages 240-255, ISSN 0950-7051.

G. E. Iyawa, M. Herselman and A. Botha, "Building a Digital Health Innovation Ecosystem Framework through Design Science Research," 2019 Conference on Next Generation Computing Applications (NextComp), Mauritius, 2019, pp. 1-6, doi: 10.1109/NEXTCOMP.2019.8883650.

Gangemi, Aldo & Catenacci, Carola & Ciaramita, Massimiliano & Lehmann, Jos. (2006). Modelling Ontology Evaluation and Validation. 140-154. 10.1007/11762256_13.

Garvita Bajaj, Rachit Agarwal, Pushpendra Singh, Nikolaos Georgantas, Valerie Issarny. A study of existing Ontologies in the IoT-domain. 2017. ffhal-01556256f

Gatius, M. and Rodríguez, H.: A domain-restricted task-guided natural language interface generator. Proceedings of the Second Edition of the Workshop Flexible Query Answering Systems (FQAS'96), Denmark (1996).

Girardi, D., Arthofer, K., & Giretzlehner, M. (2012). An Ontology-based Data Acquisition Infrastructure - Using Ontologies to Create Domain-independent Software Systems. KEOD.

Glenn E. Krasner and Stephen T. Pope. 1988. A cookbook for using the model-view controller user interface paradigm in Smalltalk-80. J. Object Oriented Program. 1, 3 (August 1988), 26-49.

Gomez-Perez, Asuncion. (2001). Evaluation of ontologies. International Journal of Intelligent Systems. 16. 391 - 409. 10.1002/1098-111X (200103)16:3<391: AID-INT1014>3.0.CO; 2-2.

Gonçalves, R.S., Tu, S.W., Nyulas, C., Tierney, M.J., & Musen, M.A. (2015). Structured data acquisition with ontology-based web forms. ICBO.

Gonçalves, Rafael & Tu, Samson & Nyulas, Csongor & Tierney, Michael & Musen, Mark. (2017). An ontology-driven tool for structured data acquisition using Web forms. Journal of Biomedical Semantics. 8. 10.1186/s13326-017-0133-1.

Grigoris Antoniou and Frank van Harmelen. 2008. A Semantic Web Primer, 2nd Edition (Cooperative Information Systems) (2 ed.). The MIT Press.

Gruber, T. R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing, KSL-93-04, Knowledge Systems Laboratory, Stanford University (1993).

Guarino, Nicola & Welty, C. (2004). An overview of ONTOCLEAN. Handbook on Ontologies, International Handbooks on Information Systems. 151-171.

Guido L. Geerts, A design science research methodology and its application to accounting information systems research, International Journal of Accounting Information Systems, Volume 12, Issue 2, 2011, Pages 142-151, ISSN 1467-0895, https://doi.org/10.1016/j.accinf.2011.02.004.

Guting, R. H. 1994. GraphDB: modeling and querying graphs in databases. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB). Morgan Kaufmann, 297–308.

Gyssens, M., Paredaens, J., AND GUCHT, D. V. 1990b. A graph-oriented object model for database end-user interfaces. In Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data. ACM Press, 24–33.

H. Neuhaus and M. Compton. The semantic sensor network ontology: a generic language to describe sensor assets. In AGILE Workshop: Challenges in Geospatial Data Harmonisation, 2009.

Haller, Armin & Janowicz, Krzysztof & Cox, Simon & Lefrançois, Maxime & Taylor, Kerry & Phuoc, Danh & Lieberman, Joshua & García Castro, Raúl & Atkinson, Robert & Stadler, Claus. The Modular SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Sampling, and Actuation. Semantic Web., 2018, 10. 10.3233/SW-180320.

Hao-Yun Kao, Min-Chun Yu, Mehedi Masud, Wen-Hsiung Wu, Li-Ju Chen, Yen-Chun Jim Wu, Design and evaluation of hospital-based business intelligence system (HBIS): A foundation for design science research methodology, Computers in

Human Behavior, Volume 62, 2016, Pages 495-505, ISSN 0747-5632, https://doi.org/10.1016/j.chb.2016.04.021.

Harth, Andreas. (2004). An integration site for Semantic Web metadata. Web Semantics: Science, Services and Agents on the World Wide Web. 1. 229-234. 10.1016/j.websem.2003.12.002.

Hevner, A., & Chatterjee, S. (2010). Design research in information systems: Theory and practice, springer, Vol. 22. New York, USA: Springer Science & Business Media.

Hevner, A.R., March, S.T., and Park, J. Design Research in Information Systems Research. MIS Quarterly, 28, 1 (2004), 75-105.

Hidders, J. And Paredaens, J. 1993. GOAL, A graph-based object and association language. Advances in Database Systems: Implementations and Applications, CISM, 247–265.

I. Szilagyi and P. Wira, "Ontologies and Semantic Web for the Internet of Things - a survey," IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, 2016, pp. 6949-6954.https://doi.org/10.1109/IECON.2016.7793744.

I. Taleb, R. Dssouli and M. A. Serhani, "Big Data Pre-processing: A Quality Framework," 2015 IEEE International Congress on Big Data, New York, NY, 2015, pp. 191-198. https://doi.org/10.1109/BigDataCongress.2015.35.

Ian Davis, Talis Thomas Steiner, Arnaud J Le Hors. RDF 1.1 JSON Alternate Serialization (RDF/JSON). Available online: https://www.w3.org/TR/rdf-json/ (accessed on 22.03.2020).

Introduction to XML. Available online: https://www.w3schools.com/xml/xml_whatis.asp (accessed on 22.03.2020).

Islam M.S., Grönlund Å. (2012) Applying Design Science Approach in ICT4D Research. In: Helfert M., Donnellan B. (eds) Practical Aspects of Design Science. EDSS 2011. Communications in Computer and Information Science, vol 286. Springer, Berlin, Heidelberg.

ISO IEC 9126-1:2001, “Software Engineering-Product Quality-Part 1-Quality Model,” 2001.

Izumi, N. and Yamaguchi, T.: Building Business Applications By Integrating Heterogeneous Repositories Based on Ontologies, (2001).

Jabbar, Sohail, Farhan Ullah, Shehzad Khalid, Murad Khan and Ki Jun Han. “Semantic Interoperability in Heterogeneous IoT Infrastructure for Healthcare.” Wireless Communications and Mobile Computing 2017 (2017): n. pag.

Jabbar, Sohail, Farhan Ullah, Shehzad Khalid, Murad Khan and Ki Jun Han. “Semantic Interoperability in Heterogeneous IoT Infrastructure for Healthcare.” Wireless

Communications and Mobile Computing 2017 (2017).

Jaime González-Buesa, María L. Salvador, An Arduino-based low cost device for the measurement of the respiration rates of fruits and vegetables, Computers and Electronics in Agriculture, Volume 162, 2019, Pages 14-20, ISSN 0168-1699

Jakob Nielsen. 1993. Usability Engineering. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Jani Holopainen, Osmo Mattila, Essi Pöyry, Petri Parvinen, Applying design science research methodology in the development of virtual reality forest management services, Forest Policy and Economics, Volume 116, 2020, 102190, ISSN 1389-9341, https://doi.org/10.1016/j.forpol.2020.102190.

Jason J. Jung, Semantic preprocessing for mining sensor streams from heterogeneous environments, Expert Systems with Applications, Volume 38, Issue 5, 2011, Pages 6107-6111, ISSN 0957-4174 https://doi.org/10.1016/j.eswa.2010.11.017.

Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Future Generation Computer Systems, Volume 29, Issue 7, 2013, Pages 1645-1660, ISSN 0167-739X

Jirapond Muangprathub, Nathaphon Boonnam, Siriwan Kajornkasirat, Narongsak Lekbangpong, Apirat Wanichsombat, Pichetwut Nillaor, IoT and agriculture data analysis for smart farm, Computers and Electronics in Agriculture, Volume 156, 2019, Pages 467-474, ISSN 0168-1699

Johannesson P., Perjons E. (2014) Introduction. In: An Introduction to Design Science. Springer, Cham.

Jones, D., Bench-capon, T., and Visser, P. : Methodologies For Ontology Development In J. Cuena, editor, Proc. ITi and KNOWS Conference of the 15th IFIP World Computer Congress, pages 62–75, Chapman and Hall Ltd. London, UK, (1998).

K. J. Witt, J. Stanley, D. Smithbauer, D. Mandl, V. Ly, A. Underbrink, and M. Metheny. Enabling Sensor Webs by utilizing SWAMO for autonomous operations. In 8th NASA Earth Science Technology Conference, 2008.

Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger & Samir Chatterjee (2007) A Design Science Research Methodology for Information Systems Research, Journal of Management Information Systems, 24:3, 45-77, DOI: 10.2753/MIS0742-1222240302.

Khan, Z.C. (2018). Applying evaluation criteria to ontology modules. In: Joint Ontology Workshops 2018, Cape Town, South Africa, 17-21 September.

Kirakowski, J. (1995) The software usability measurement inventory: background and usage, in Usability Evaluation in Industry, Jordan, P., Thomas, B. and Weerdmeester, B. (eds) (Taylor & Frances, London).

Kirakowski, J. and Corbett, M. (1993), SUMI: the Software Usability Measurement Inventory. British Journal of Educational Technology, 24: 210-212. doi:10.1111/j.1467-8535.1993.tb00076.x

Koksal, A.,I., Gunes, N., Descriptors for Hazelnut, Bioversity International and FAO, 2008.

Kostovski, M., Jovanovik, M., & Trajanov, D. (2012). Open data portal based on semantic Web technologies. The Proceedings of the 7th South East European Doctoral Student Conference (SEEDSC'12) (pp. 1–13). Thessaloniki, Greece: SEERC.

L. Chen, Z. Xu and L. Ni, "WF2OML: A Modeling Language for Mapping Web Forms to Ontology," 2013 10th Web Information System and Application Conference, Yangzhou, 2013, pp. 456-461.doi: 10.1109/WISA.2013.92

L. De Silva et al., "Design science research based blended approach for usability driven requirements gathering and application development," 2014 IEEE 2nd International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE), Karlskrona, 2014, pp. 17-24, doi: 10.1109/UsARE.2014.6890996.

Lata, Swaran & Sinha, Bhaskar & Kumar, Ela & Chandra, Somnath & Arora, Raghu, Semantic Web Query on e-Governance Data and Designing Ontology for Agriculture Domain. International Journal of Web & Semantic Technology. 4., 2013 http://doi.org/10.5121/ijwest.2013.4307.

Li Minbo, Zhu Zhu & Chen Guangyu (2013) Information Service System Of Agriculture IoT, Automatika, 54:4, 415-426, DOI: 10.7305/automatika.54-4.413

Liyanage H, Krause P, De Lusignan S. Using ontologies to improve semantic interoperability in health data. J Innov Health Inform. 2015 Jul 10;22(2):309-15. doi: 10.14236/jhi.v22i2.159. PubMed PMID: 26245245.

Lozano-Tello, A & Gomez-Perez, Asuncion. (2004). ONTOMETRIC: a method to choose the appropriate ontology. J. Database Manag.. 15. 1-18. 10.4018/jdm.2004040101.

Luigi Atzori, Antonio Iera, Giacomo Morabito, The Internet of Things: A survey, Computer Networks, Volume 54, Issue 15, 2010, Pages 2787-2805, ISSN 1389-1286

Luo Y., Picalausa F., Fletcher G.H.L., Hidders J., Vansummeren S., Storing and Indexing Massive RDF Datasets. In: De Virgilio R., Guerra F., Velegrakis Y. (eds) Semantic Search over the Web. Data-Centric Systems and Applications. Springer, Berlin, Heidelberg, 2012, https://doi.org/10.1007/978-3-642-25008-8_2.

M. Eid, R. Liscano, and A. E. Saddik. A novel ontology for sensor networks data. In IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2006.

M. Graves, E. R. Bergeman and C. B. Lawrence, "A graph-theoretic data model for genome mapping databases," Proceedings of the Twenty-Eighth Annual Hawaii International Conference on System Sciences, Wailea, HI, USA, 1995, pp. 32-41 vol.5.

M. Xie, J. Hu and S. Guo, "Segment-Based Anomaly Detection with Approximated Sample Covariance Matrix in Wireless Sensor Networks," in IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 2, pp. 574-583, Feb. 2015.

Madhura Jayaratne, Dinithi Nallaperuma, Daswin De Silva, Damminda Alahakoon, Brian Devitt, Kate E. Webster, Naveen Chilamkurti, A data integration platform for patient-centered e-healthcare and clinical decision support, Future Generation Computer Systems, Volume 92, 2019, Pages 996-1008, ISSN 0167-739X, https://doi.org/10.1016/j.future.2018.07.061.

Mahmud Al-Osta, Bali Ahmed, Gherbi Abdelouahed, A Lightweight Semantic Web-based Approach for Data Annotation on IoT Gateways, Procedia Computer Science, Volume 113, 2017, Pages 186-193, ISSN 1877-0509

Maria Ganzha, Marcin Paprzycki, Wiesław Pawłowski, Paweł Szmeja, Katarzyna Wasielewska, Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective, Journal of Network and Computer Applications, Volume 81, 2017, Pages 111-124, ISSN 1084-8045

Mark Schildhauer, Matthew B. Jones, Shawn Bowers, Joshua Madin, Sergeui Krivov, Deana Pennington, Ferdinando Villa, Benjamin Leinfelder, Christopher Jones, Margaret O'Brien. The Extensible Observation Ontology. Available online: http://agroportal.lirmm.fr/ontologies/OBOE (accessed on 22.03.2020).

Masolo, C., Gangemi, A., Guarino, N., Oltramari, A., and Schneider, L.: WonderWeb Deliverable D17: The WonderWeb Library of Foundational Ontologies. (2002)

Matthew MacDonald and Adam Freeman. 2010. Pro ASP.NET 4 in C# 2010, Fourth Edition (4th ed.). Apress, Berkely, CA, USA.

Mcdaniel, Melinda & Storey, Veda. (2019). Evaluating Domain Ontologies: Clarification, Classification, and Challenges -- Computing Surveys: Abstract References Preprint.

Mehrdokht Pournader, Amin Akhavan Tabassi, Peter Baloh, A three-step design science approach to develop a novel human resource-planning framework in projects: the cases of construction projects in USA, Europe, and Iran, International Journal of Project Management, Volume 33, Issue 2, 2015, Pages 419-434, ISSN 0263-7863, https://doi.org/10.1016/j.ijproman.2014.06.009.

Melissa J. Perry, Agricultural Health and Safety, Editor(s): Stella R. Quah, International Encyclopedia of Public Health (Second Edition), Academic Press, 2017, Pages 38-44, ISBN 9780128037089

Mell, P., G., Tim, 2018, National Institute of Standards and Technology, Information Technology Laboratory – a nonregulatory federal agency within the U.S.

Department of Commerce – http://csrc.nist.gov/groups/SNS/cloud-computing.

Michael Compton, Payam Barnaghi, Luis Bermudez, Raúl García-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, Vincent Huang, Krzysztof Janowicz, W. David Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin Page, Alexandre Passant, Amit Sheth, Kerry Taylor, The SSN ontology of the W3C semantic sensor network incubator group, Journal of Web Semantics, Volume 17, 2012, Pages 25-32, ISSN 1570-8268

Michael Shirer and Carrie MacGillivray. The Growth in Connected IoT Devices Is Expected to Generate 79.4ZB of Data in 2025, According to a New IDC Forecast. Available online: https://www.idc.com/getdoc.jsp?containerId=prUS45213219 (accessed on 18.03.2020).

Minhong Wang, Doug Vogel, Weijia Ran, Creating a performance-oriented e-learning environment: A design science approach, Information & Management, Volume 48, Issue 7, 2011, Pages 260-269, ISSN 0378-7206, https://doi.org/10.1016/j.im.2011.06.003.

Musen, M.A. The Protégé project: A look back and a look forward. AI Matters. Association of Computing Machinery Specific Interest Group in Artificial Intelligence, 1(4), June 2015. DOI: 10.1145/2557001.25757003.

N. F. Noy and C. D. Hafner, “The State of the Art in Ontology Design: A Survey and Comparative Review”, AIMag, vol. 18, no. 3, p. 53, Sep. 1997, https://doi.org/10.1609/aimag.v18i3.1306 .

Niels Martin, Jochen Bergs, Dorien Eerdekens, Benoît Depaire, Sandra Verelst, Developing an emergency department crowding dashboard: A design science approach, International Emergency Nursing, Volume 39, 2018, Pages 68-76, ISSN 1755-599X, https://doi.org/10.1016/j.ienj.2017.08.001.

Noura M., Atiquzzaman M., Gaedke M. (2018) Interoperability in Internet of Things Infrastructure: Classification, Challenges, and Future Work. In: Lin YB., Deng DJ., You I., Lin CC. (eds) IoT as a Service. IoTaaS 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 246. Springer, Cham

Noura, M., Atiquzzaman, M. & Gaedke, M. Interoperability in Internet of Things: Taxonomies and Open Challenges. Mobile Netw Appl 24, 796–809 (2019).

Noy, N.F., McGuinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. (2001).

Odette Sangupamba Mwilu, Isabelle Comyn-Wattiau, Nicolas Prat, Design science research contribution to business intelligence in the cloud — A systematic literature review, Future Generation Computer Systems, Volume 63, 2016, Pages 108-122, ISSN 0167-739X, https://doi.org/10.1016/j.future.2015.11.014.

P. P. Jayaraman, D. Palmer, A. Zaslavsky and D. Georgakopoulos, "Do-it-Yourself Digital Agriculture applications with semantically enhanced IoT platform," 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 2015, pp. 1-6.

P. Vcelak, M. Kryl, M. Kratochvil and J. Kleckova.(2017) "Ontology-based web forms — Acquisition and modification of structured data," 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, pp. 1-5. doi: 10.1109/CISP-BMEI.2017.8302291.

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V. & Bragge, J.(2006). The Design Science Research Process : A Model for Producing and Presenting Information Systems Research. In 1st International Conference, DESRIST 2006 Proceedings. (pp. 83-106). Claremont Graduate University.

Peter Murray-Rust, Open Data in Science, Serials Review, 34:1, 52-64, 2008, https://doi.org/10.1080/00987913.2008.10765152.

Pollock, J., T.: Semantic Web for Dummies. 1st edn. Wiley Publishing, Inc., Canada ,2009, ISBN: 978-0-470-39679-7 .

Poveda-Villalón, María & Gomez-Perez, Asuncion & Suárez-Figueroa, Mari Carmen. (2014). OOPS! (OntOlogy Pitfall Scanner!): An on-line tool for ontology evaluation. International Journal on Semantic Web and Information Systems. 10. 7-34. 10.4018/ijswis.2014040102.

Reenskaug, T. (1979). The original MVC reports.

Reenskaug, T. (2003). The Model-View-Controller (MVC) Its Past and Present.

Resource Description Framework (RDF). Available online: https://www.w3.org/RDF/ (accessed on 22.03.2020).

Resource Description Framework. [online] Available at: https://www.w3.org/RDF/ [Accessed 23.07.2018]

RFID Journal Webpage: Available online: https://www.rfidjournal.com/articles/view?4986 (accessed on 18.03.2020).

Robinson, I. A. N. (2016). Graph Databases: new opportunities for connected data. Place of publication not identified: SHROFF Publishers & DISTR.

Rubin, Daniel & Hewett, Micheal & Oliver, Diane & Klein, Teri & Altman, Russ. (2002). Automating Data Acquisition Into Ontologies From Pharmacogenetics Relational Data Sources Using Declarative Object Definitions And XML. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing. 88-99. 10.1142/9789812799623_0009.

Rueda, C.; Galbraith, N.; Morris, R. A.; Bermudez, L. E.; Graybeal, J.; Arko, R. A.; Mmi

Device Ontology Working Group. The MMI Device Ontology: Enabling Sensor Integration. Available online: https://ui.adsabs.harvard.edu/abs/2010AGUFMIN44B..08R/abstract (accessed on 22.03.2020).

S. Avancha, C. Patel and A. Joshi, "Ontology-driven adaptive sensor networks," The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004., Boston, MA, USA, 2004, pp. 194-202.

S. K. Sowe and K. Zettsu, "Towards an Open Data Development Model for Linking Heterogeneous Data Sources," 2015 Seventh International Conference on Knowledge and Systems Engineering (KSE), Ho Chi Minh City, 2015, pp. 344-347, doi: 10.1109/KSE.2015.56.

Schlicht, A., Stuckenschmidt, H.: Towards structural criteria for ontology modularization. In: Proceedings of the 1st InternationalWorkshop on Modular Ontologies, (WoMO'06). CEURWorkshop Proceedings, vol. 232. CEUR-WS.org (2006), iSWC'06 November 5, 2006, Athens, Georgia, USA

Sensor Model Language (SensorML). Available online: https://www.ogc.org/standards/sensorml (accessed on 22.03.2020).

Shastri L Nimmagadda, Amboge Samson, Neel Mani, Torsten Reiners, Design Science Information System Framework for Managing the Articulations of Digital Agroecosystems, Procedia Computer Science, Volume 159, 2019, Pages 1198-1207, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2019.09.289.

SKOS Simple Knowledge Organization System eXtension for Labels (SKOS-XL) Namespace Document - HTML Variant. [online] Available at: https://www.w3.org/TR/skos-reference/skos-xl.html, [Accessed 2019/01/15].

SPARQL 1.1 Query Language. [online] Available at: https://www.w3.org/TR/2013/REC-sparql11-query-20130321/ [Accessed 23.07.2018].

Stvilia, Besiki. (2007). A model for ontology quality evaluation. First Monday. 12. 10.5210/fm.v12i12.2043.

Sure, Y. and Studer, R.: On-To-Knowledge Methodology-Final Version. (2002)

Susanna Pantsar-Syväniemi, Anu Purhonen, Eila Ovaska, Jarkko Kuusijärvi, and Antti Evesti. 2012. Situation-based and self-adaptive applications for the smart environment. J. Ambient Intell. Smart Environ. 4, 6 (November 2012), 491–516.

Szilaghyi, I., Wira, P.: Ontologies and Semantic Web for the Internet of Things – A Survey, IEEE, 6949-6954 (2016).

T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American, May-2001, pp. 29–37.

T. Elsaleh, M. Bermudez-Edo, S. Enshaeifar, S. T. Acton, R. Rezvani and P. Barnaghi, "IoT-Stream: A Lightweight Ontology for Internet of Things Data Streams," 2019 Global IoT Summit (GIoTS), Aarhus, Denmark, 2019, pp. 1-6. doi: 10.1109/GIOTS.2019.8766367

Tamoghna Ojha, Sudip Misra, Narendra Singh Raghuwanshi, Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges, Computers and Electronics in Agriculture, Volume 118, 2015, Pages 66-84, ISSN 0168-1699

Tartir, S., Arpinar, I. B., Moore, M., Sheth, A. P., & Aleman-Meza, B. (2005). OntoQA: Metric-Based Ontology Quality Analysis. .https://corescholar.libraries.wright.edu/knoesis/660

Tavakol, Mohsen, and Reg Dennick. “Making sense of Cronbach's alpha.” International journal of medical education vol. 2 53-55. 27 Jun. 2011, doi:10.5116/ijme.4dfb.8dfd

The Food and Agriculture Organization (FAO). (2014).Statistical Yearbook of the Food and Agriculture Organization for the United Nations. Retrieved from FAO website: http://www.fao.org/3/a-i3621e.pdf

Thomas R. Gruber, Toward principles for the design of ontologies used for knowledge sharing?, International Journal of Human-Computer Studies, Volume 43, Issues 5–6, 1995, Pages 907-928, ISSN 1071-5819, https://doi.org/10.1006/ijhc.1995.1081.

Tim Berners-Lee, Dan Connolly. Notation3 (N3): A readable RDF syntax. Available online: https://www.w3.org/TeamSubmission/n3/ (accessed on 22.03.2020).

Tran, Van-Anh & Johnson, Nathaniel & Redline, Susan & Zhang, Guo-Qiang. (2011). OnWARD: Ontology-driven Web-based Framework for Multi-center Clinical Studies. Journal of biomedical informatics. 44 Suppl 1. S48-53. 10.1016/j.jbi.2011.08.019.

TUIK Duties and Authorities Available at: http://www.turkstat.gov.tr/UstMenu.do?metod=gorevYetki [Accessed 24.07.2018].

Turkey's Hazelnut. [online] Available at: http://www.ftg.org.tr/en/turkish-hazelnutturkeys- hazelnut.html [Accessed 21.07.2018].

U. Ungkawa, D. H. Widyantoro and B. Hendradjaya, "Methodology for constructing form ontology," 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, 2017, pp. 1-6. doi: 10.1109/EECSI.2017.8239196

Underbrink, A., Witt, K., Stanley, J., Mandl, D., 2008.Autonomous mission operations for sensor webs. AGU Fall Meet.Abstr., C5.

Uschold, M., and Gruninger, M.: Ontologies: Principles: Methods and Applications, Knowledge Engineering Review 11(2):93-155 (1996).

V. M. Tayur and R. Suchithra, "A Comprehensive Ontology for Internet of Things (COIoT)," 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India, 2019, pp. 1-6. doi: 10.1109/ICACCP.2019.8882936

Veenendaal, van, E. P. W. M. (2001). Questionnaire based usability testing. In M. Oivo, & P. Kuvaja (Eds.), Infill systems for residential open building: comparison and status report of developments in four countries Oulu, Finland: VTT (Technical Research Centre of Finland).

W3C, OWL, 11-Dec-2013, [Online], Available: https://www.w3.org/2001/sw/wiki/OWL. (Accessed: 30-Nov-2019)

W3C, Semantic Web Main Page, 1-Aug-2019, [Online], Available: https://www.w3.org/2001/sw/wiki/Main_Page. (Accessed: 30-Nov-2019)

Walter, Matthias. "An Approach to Transforming Requirements into Evaluable UI Design for Contextual Practice - a Design Science Research Perspective." 2018 Federated Conference on Computer Science and Information Systems (FedCSIS) (2018): 715-724.

Wenguang Wang, Andreas Tolk, and Weiping Wang. 2009. The levels of conceptual interoperability model: applying systems engineering principles to M&S. In Proceedings of the 2009 Spring Simulation Multiconference (SpringSim '09). Society for Computer Simulation International, San Diego, CA, USA, Article 168, 1–9.

What is Arduino? Available online: https://www.arduino.cc/en/guide/introduction (accessed on 22.03.2020)

What is HTML? Available online: https://www.w3schools.com/whatis/whatis_html.asp (accessed on 22.03.2020).

What is JSON? Available online: https://www.w3schools.com/whatis/whatis_json.asp (accessed on 22.03.2020).

What is SUMI? Available online: http://sumi.uxp.ie/about/whatis.html (accessed on 19.04.2020)

XBee Zigbee Mesh Kit. Available online: https://www.digi.com/resources/documentation/digidocs/pdfs/90001942-13.pdf (accessed on 24.03.2020)

XBee/XBee-PRO S2C Zigbee. Available online: https://www.digi.com/resources/documentation/digidocs/pdfs/90002002.pdf (accessed on 24.03.2020)

Xiang Su, Hao Zhang, Jukka Riekki, Ari Keränen, Jukka K. Nurminen, Libin Du, Connecting IoT Sensors to Knowledge-based Systems by Transforming SenML to RDF, Procedia Computer Science, Volume 32, 2014, Pages 215-222, ISSN 1877-

0509

XML Web Services. [online] Available at: https://www.w3schools.com/xml/xml_services.asp [Accessed 25.07.2018].

Yan Dang, Yulei Zhang, Paul Jen-Hwa Hu, Susan A. Brown, Hsinchun Chen, Knowledge mapping for rapidly evolving domains: A design science approach, Decision Support Systems, Volume 50, Issue 2, 2011, Pages 415- 427, ISSN 0167-9236, https://doi.org/10.1016/j.dss.2010.10.003.

YuanYuan Li and L. E. Parker, "A spatial-temporal imputation technique for classification with missing data in a wireless sensor network," 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, 2008, pp. 3272-3279.

Zheng, Jie & Harb, Omar & Jr, Christian. (2011). Ontology Driven Data Collection for EuPathDB. International Conference on Biomedical Ontology (ICBO).

Zhong Y, Fong S, Hu S, Wong R, Lin W. A Novel Sensor Data Pre-Processing Methodology for the Internet of Things Using Anomaly Detection and Transfer-By-Subspace-Similarity Transformation. Sensors (Basel). 2019;19(20):4536. Published 2019 Oct 18. https://doi.org/10.3390/s19204536.

Zhu, H., Liu, D., Bayley, I., Aldea, A., Yang, Y., & Chen, Y. (2017). Quality Model and Metrics of Ontology for Semantic Descriptions of Web Services. TSINGHUA SCIENCE AND TECHNOLOGY, 22(3), 254–272.

Zimmermann, Antoine & Lopes, Nuno & Polleres, Axel & Straccia, Umberto. (2011). A General Framework for Representing, Reasoning and Querying with Annotated Semantic Web Data. Journal of Web Semantics. 11. 10.1016/j.websem.2011.08.006.

CURRICULUM VITAE

Personal Information

Name Surname : Şahin AYDIN

Place and Date of Birth : Ulubey - ORDU / 28.01.1986

Education

Undergraduate Education :
Marmara University
Ataturk Education of Faculty
B.Sc. in Computer Education and Instructional Technologies

Graduate Education :
Istanbul University
Institute of Sciences
M.Sc. in Informatics

Foreign Language Skills : English

Work Experience

Name of Employer and Dates of Employment:
Istanbul Ayvansaray University
Lecturer
Computer Programming
Computer Engineering
2011 - Present

APPENDIX A

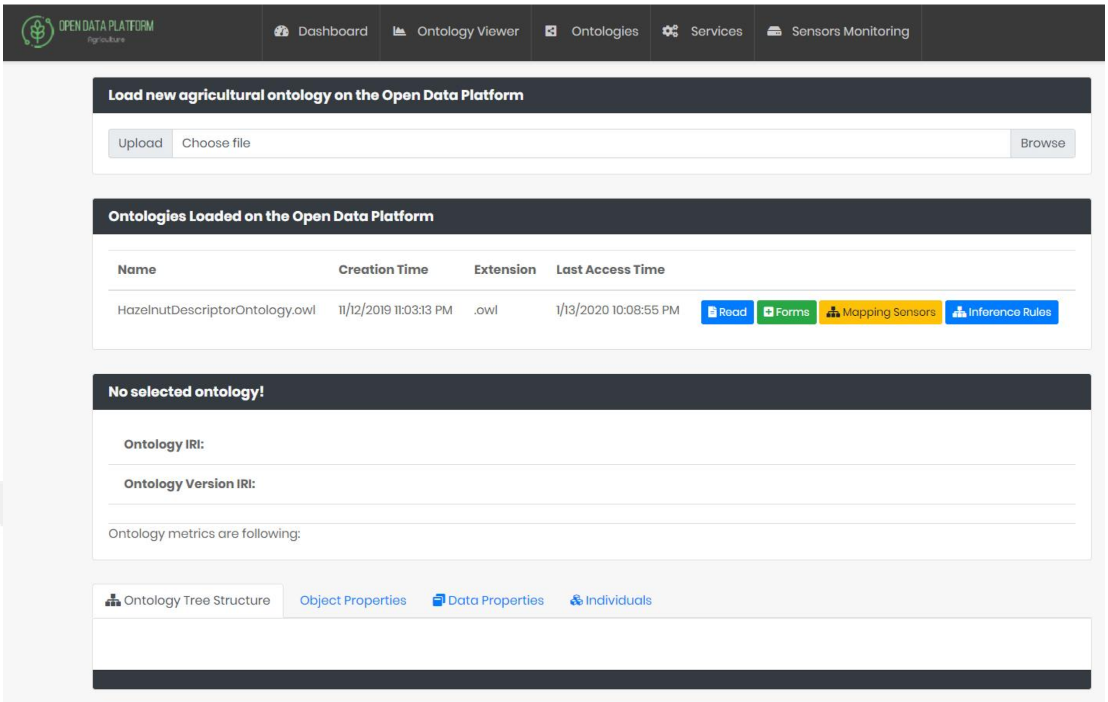


Figure A.1 Screenshot of Ontology Viewing and Loading

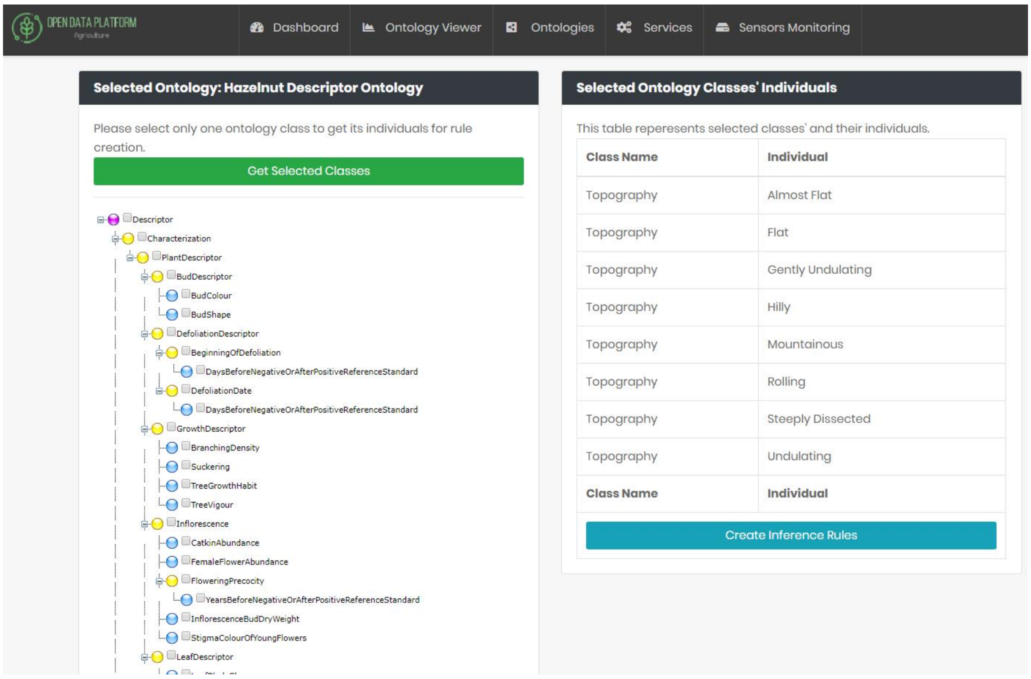


Figure A.2 Screenshot of Selecting Individuals of Traits for Inference Rules Creation

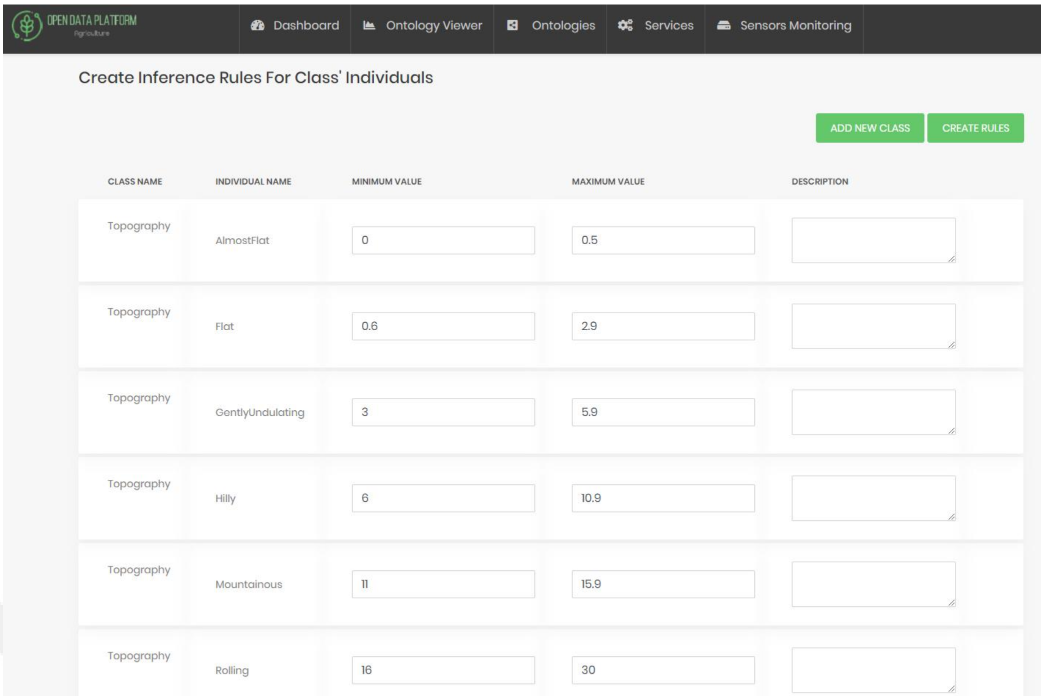


Figure A.3 Screenshot of Creating Rules Definitions for Individuals

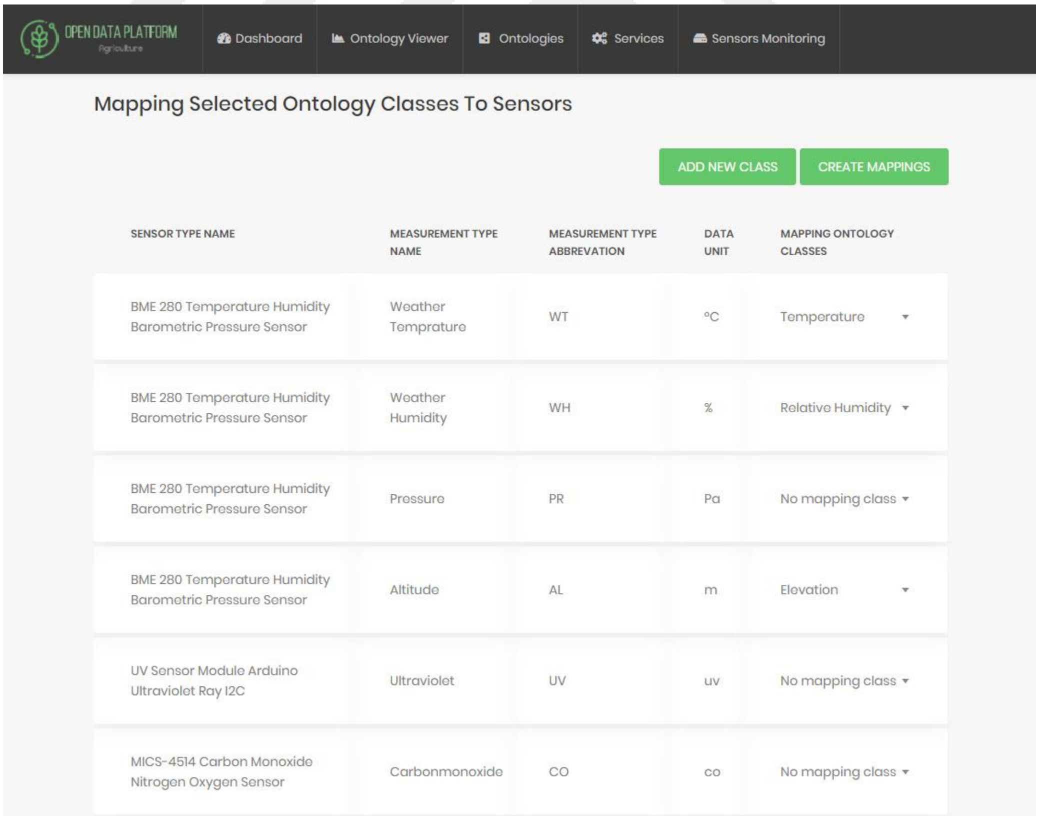


Figure A.4 Screenshot of Creating Mapping Rules

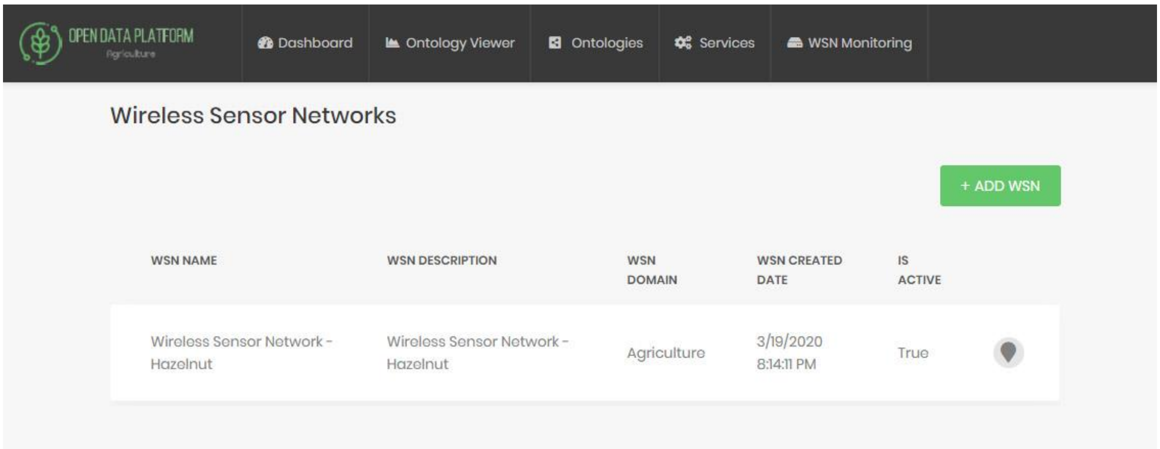


Figure A.5 Screenshot of Wireless Sensor Networks Management

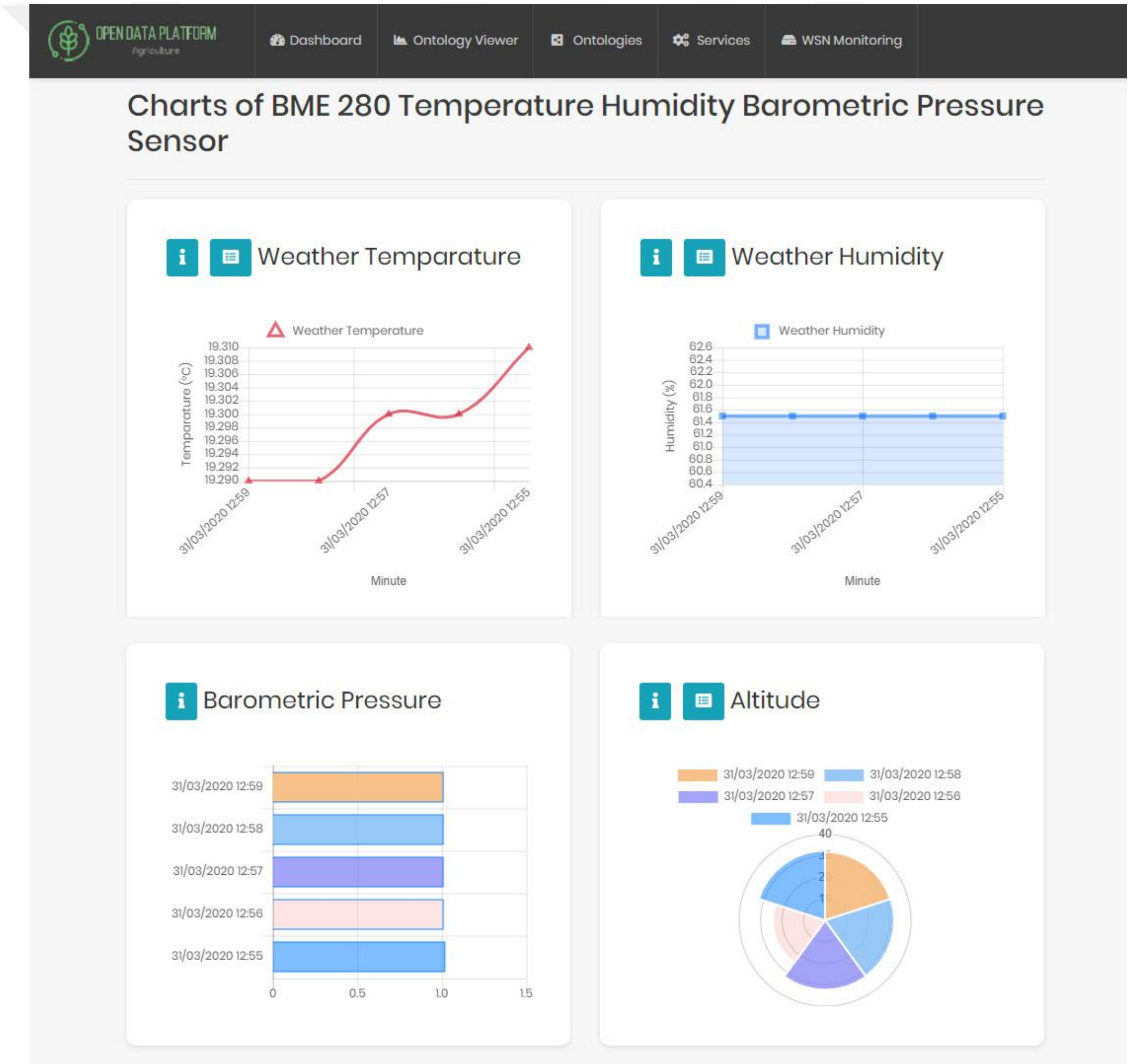


Figure A.6 Screenshot of Charting and Managing Mapped Sensor Measurements

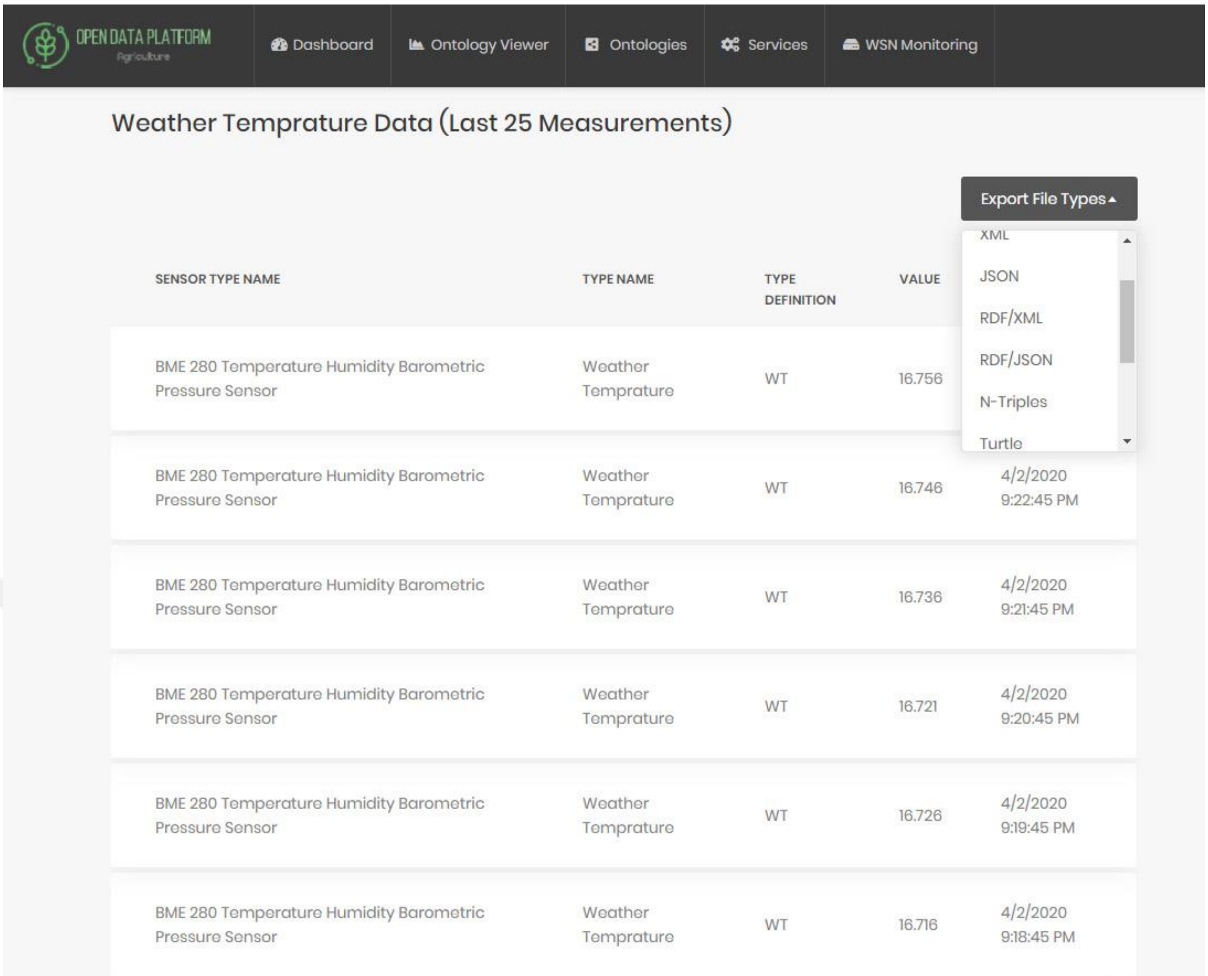


Figure A.7 Screenshot of Exporting Linked Sensor Measurement Value

APPENDIX B

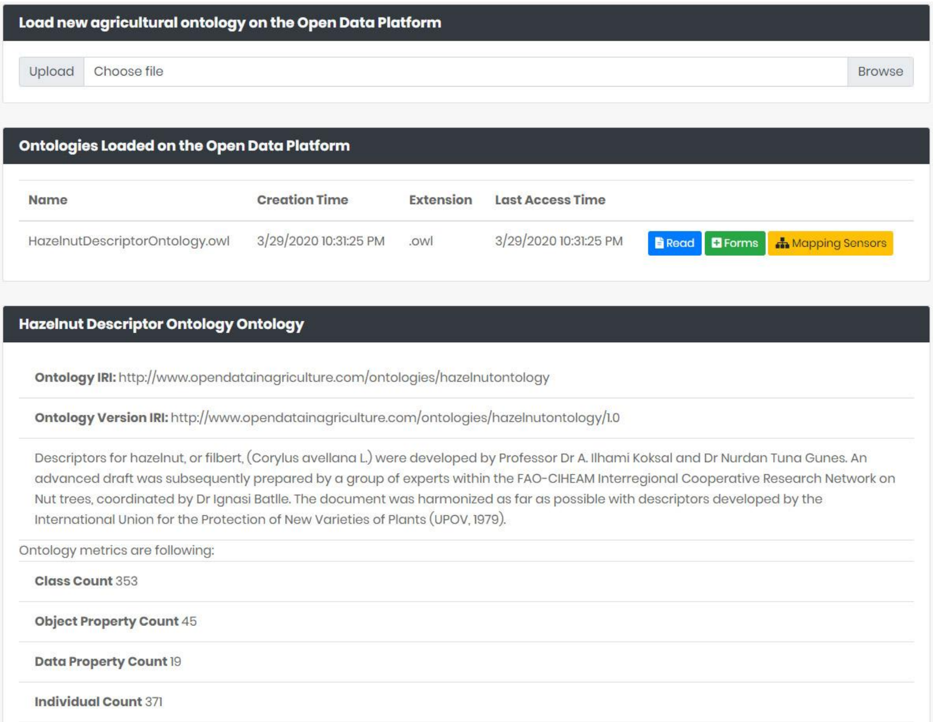


Figure B.1 The module of Uploading and Viewing Ontologies for Data Acquisition Form Generation

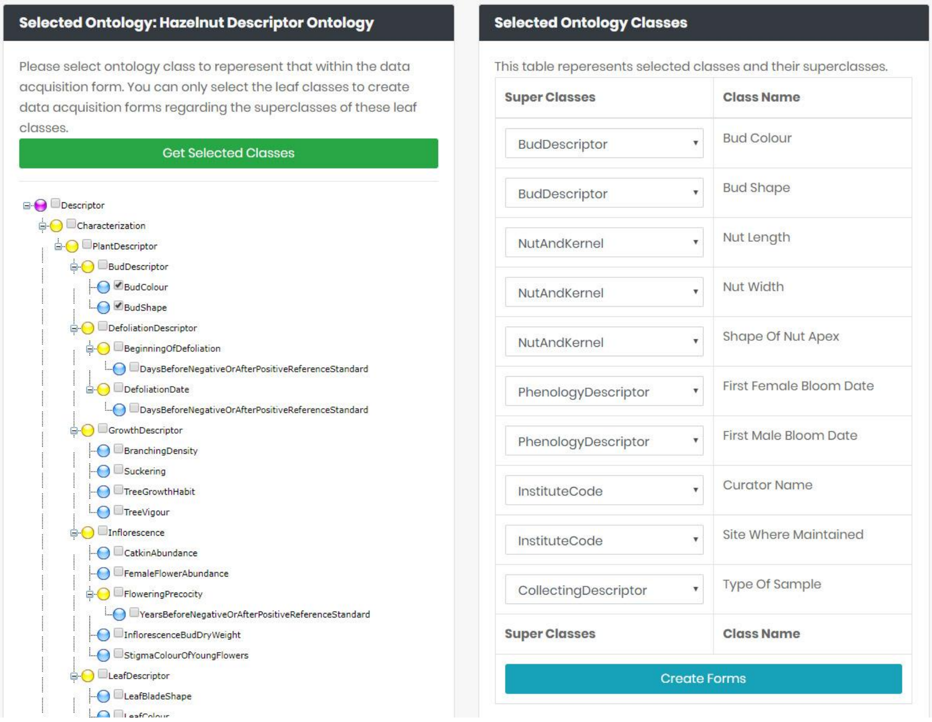


Figure B.2 The Module of Viewing and Selecting Ontology Classes for Data Acquisition Form Generation

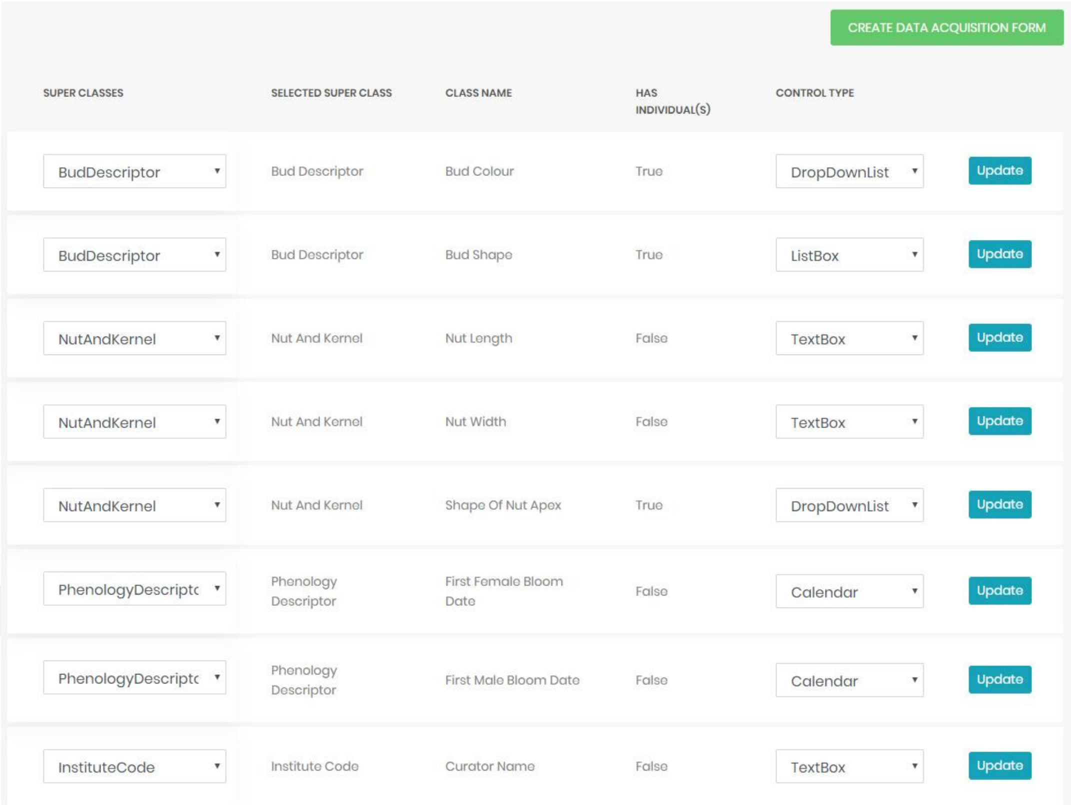


Figure B.3 The Module of Providing to Map ASP.NET Controls to Ontology Classes

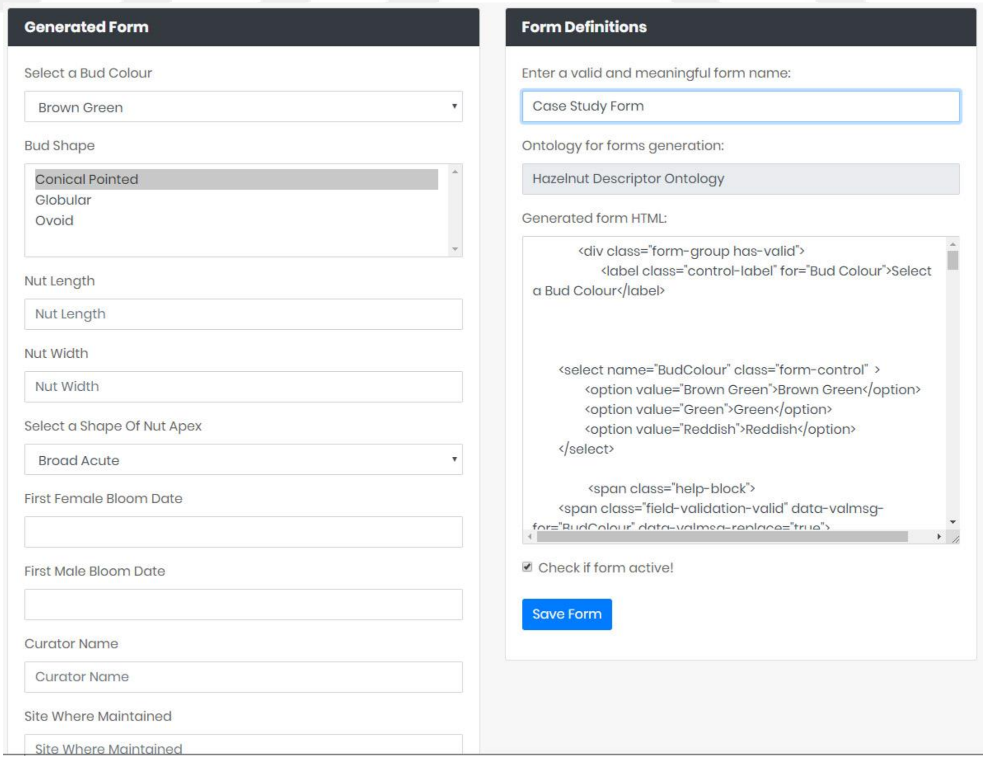


Figure B.4 The Module of Viewing and Editing the Generated Data Acquisition Form

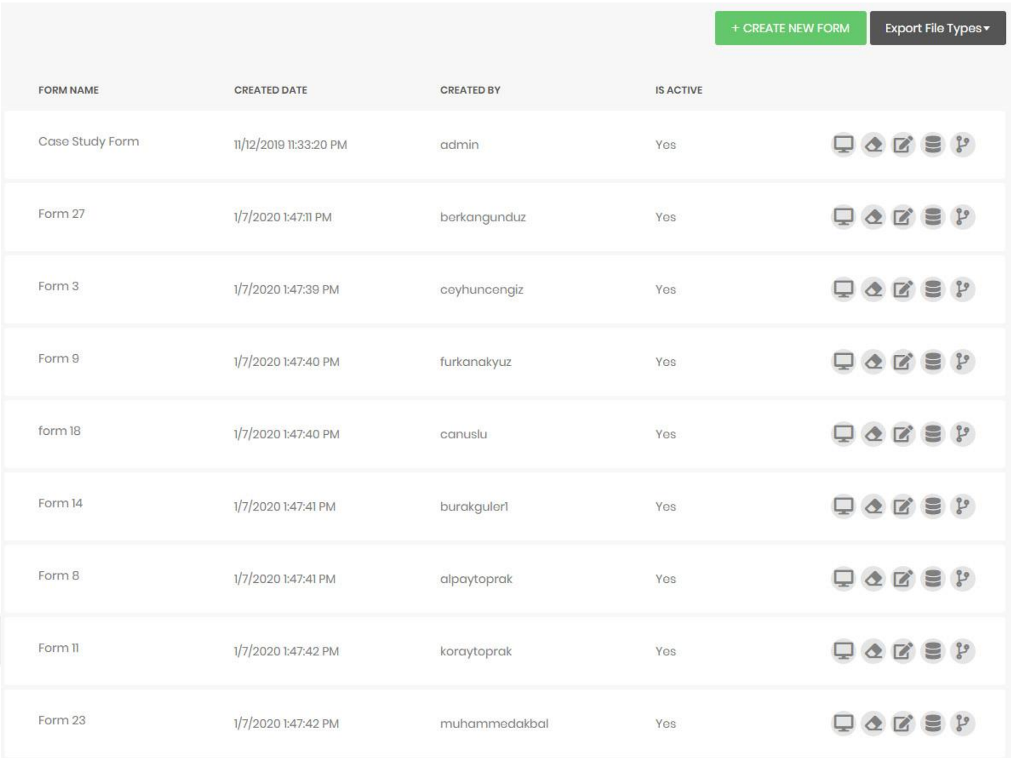


Figure B.5 The Module of Listing Generated Data Acquisition Forms

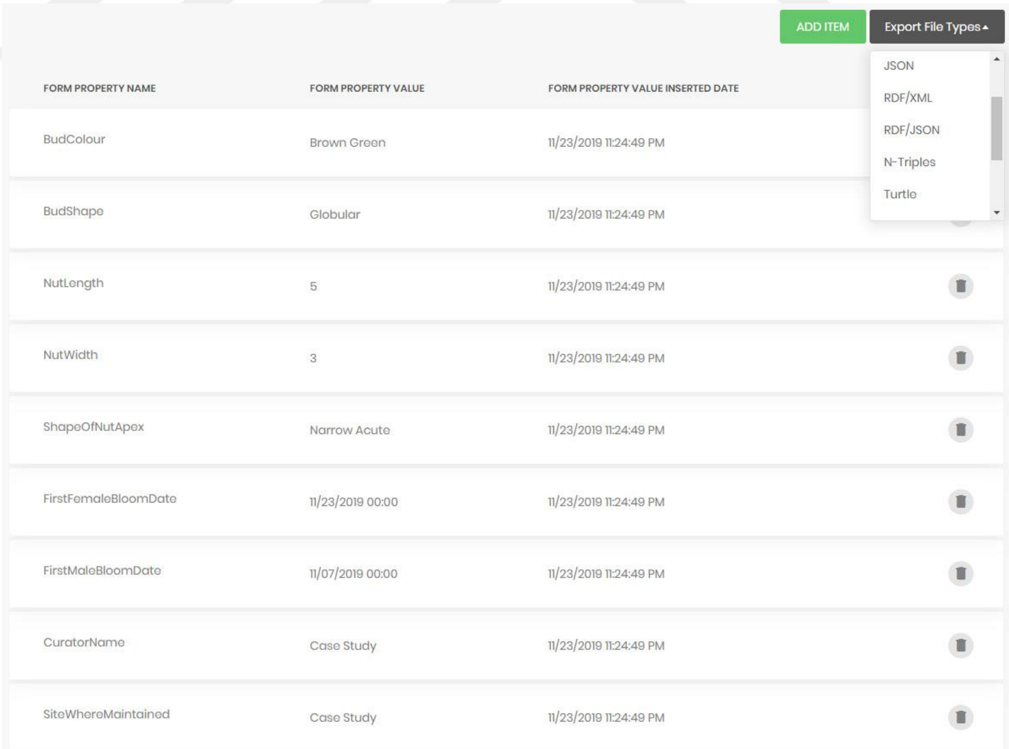


Figure B.6 The Module of Listing and Exporting Data Collected Using Generated Data Acquisition Form

APPENDIX C

Table C.1 SUMI User Records for OWL2MVC

Respondent ID	Affect	Controllability	Efficiency	Helpfulness	Learnability	Global Usability Score
1	94	94	100	63	91	88
2	46	29	49	25	29	35
3	92	61	57	63	7	56
4	94	94	100	92	91	94
5	94	94	100	63	91	88
6	89	94	100	92	68	88
7	14	14	40	18	22	21
8	94	61	100	92	58	81
9	54	67	70	48	42	56
10	92	61	57	63	7	56
11	46	44	47	43	32	42
12	67	63	41	62	33	53
13	47	48	80	76	24	55
14	6	5	8	32	27	15
15	80	73	88	88	71	80
16	94	94	37	63	40	65
17	58	31	83	73	33	55
18	84	94	88	88	69	84
19	11	9	33	45	19	23
20	85	91	100	88	41	81
21	78	74	80	57	49	67
22	89	73	51	92	58	72
23	71	67	83	79	59	71
24	80	53	91	82	69	75
25	89	90	79	25	74	71
26	94	91	88	92	59	84
27	89	77	88	88	68	82
28	89	69	100	63	77	79
29	94	94	100	92	91	94
30	58	90	100	88	78	82
31	94	91	91	81	27	76
32	94	94	100	92	78	91
33	94	94	100	92	58	87
34	94	84	100	92	77	89
35	89	94	100	92	77	90
36	83	94	100	82	77	87
37	47	94	88	92	78	79
38	73	74	88	64	77	75
39	83	94	100	92	69	87
40	70	94	100	87	61	82
41	83	94	100	92	77	89
42	94	69	100	92	77	86
43	94	94	100	92	77	91
44	58	90	88	53	45	66
45	94	90	88	92	69	86
46	94	94	100	92	77	91
47	94	94	100	92	69	89
48	94	61	100	92	77	84

49	94	94	100	92	91	94
50	58	50	100	92	69	73
51	83	84	100	63	40	74
52	89	94	100	92	77	90
53	83	94	81	92	77	85

Table C.2 SUMI User Records for Semantic Interoperability Tools

Participant	Global	Efficiency	Affect	Helpfulness	Control	Learnability
1	75	67	69	72	74	62
2	75	67	54	72	66	63
3	74	60	64	72	66	53
4	74	64	65	72	64	61
5	74	60	61	70	65	53
6	74	67	56	72	65	69
7	72	64	69	72	70	57
8	71	67	59	70	60	62
9	70	63	69	65	66	55
10	70	66	62	65	65	53
11	70	67	66	66	66	61
12	69	64	69	65	74	51
13	69	61	69	65	65	48
14	68	64	69	62	66	55
15	68	57	64	65	65	62
16	68	64	69	62	65	55
17	68	67	59	62	66	62
18	68	66	59	62	65	61
19	67	67	69	62	66	58
20	67	58	64	62	66	61
21	67	60	63	61	58	54
22	67	64	66	62	69	67
23	66	65	59	60	66	69
24	65	64	69	58	74	51
25	53	69	70	46	61	45
26	53	48	63	43	49	40
27	45	60	67	38	48	45

APPENDIX D

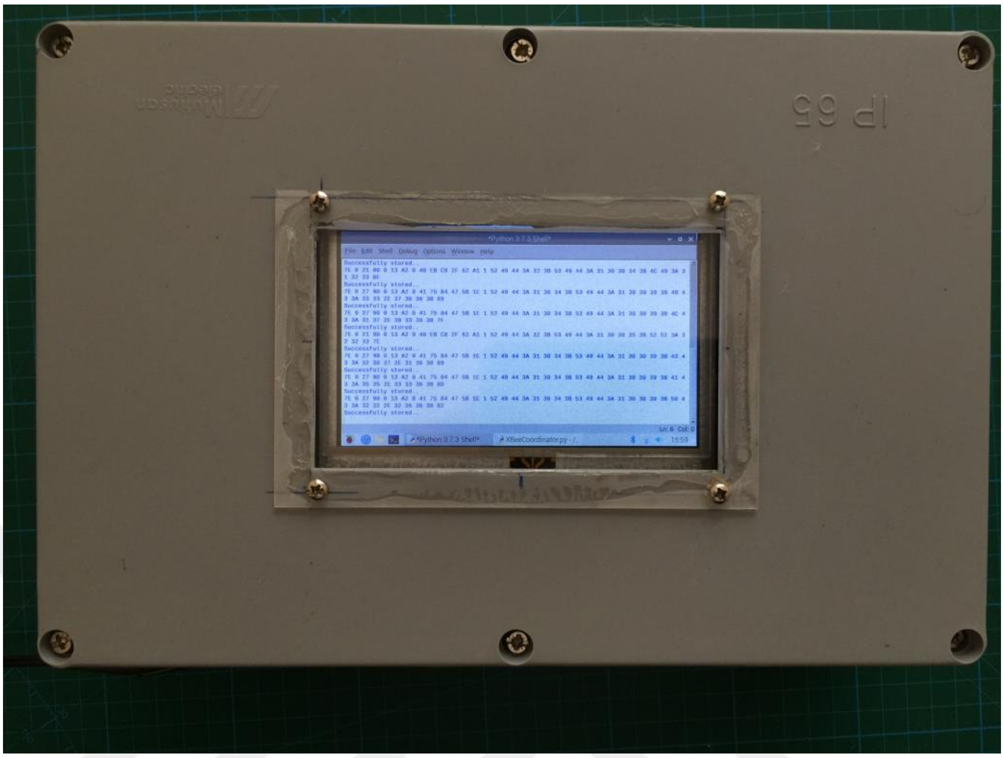


Figure D.1 Coordinator Device

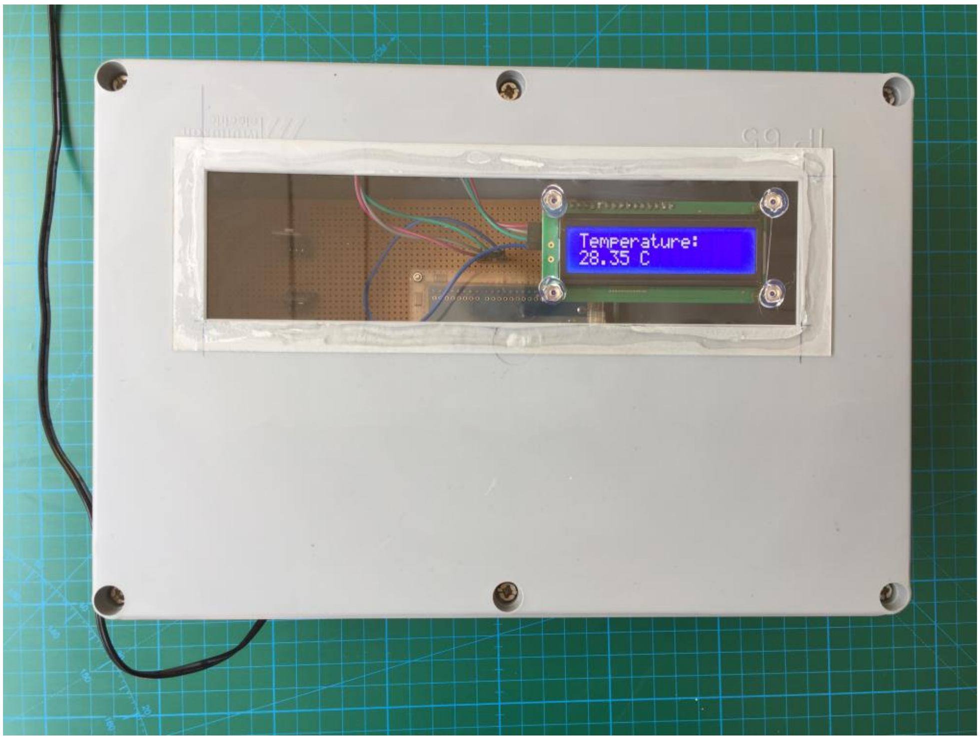


Figure D.2 Router 1 Device

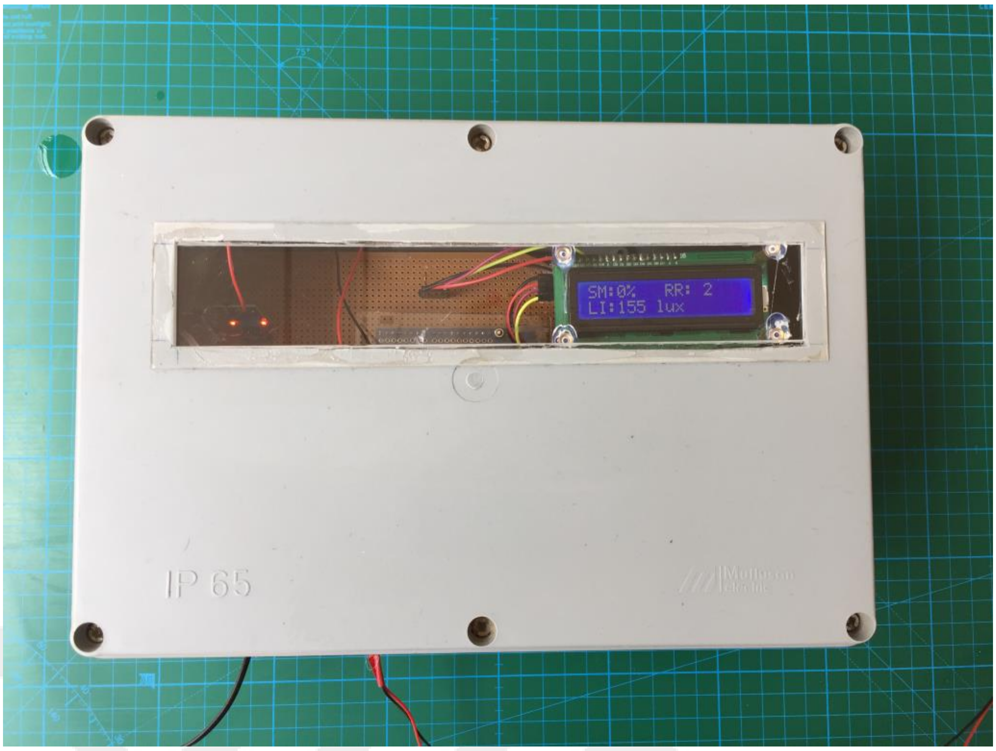


Figure D.3 Router 2 Device

Figure D.4 Router 3 Device

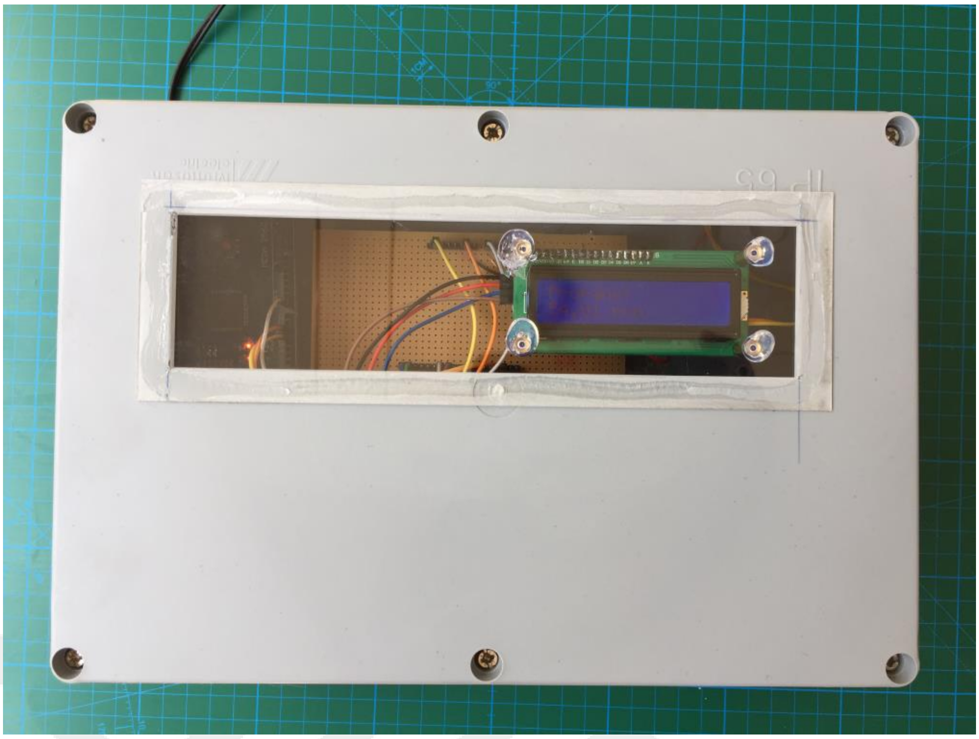


Figure D.5 Router 4 Device