ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY SCHOOL OF ENGINEERING DEPARTMENT OF COMPUTING

A KNOWLEDGE REPRESENTATION APPROCH FOR ETHIOPIAN TRADITIONAL MEDICINE

YOHANNES EPHREM

MAY, 2017

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY SCHOOL OF ENGINEERING DEPARTMENT OF COMPUTING

A KNOWLEDGE REPRESENTATION APPROCH FOR ETHIOPIAN TRADITIONAL MEDICINE

A Thesis Submitted to the Adama Science and Technology University School of Engineering Department of Computing in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information System

By

YOHANNES EPHREM

MAY, 2017

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY SCHOOL OF ENGINEERING DEPARTMENT OF COMPUTING

A KNOWLEDGE REPRESENTATION APPROCH FOR ETHIOPIAN TRADITIONAL MEDICINE

By

YOHANNES EPHREM

Name and Signature of Members of the Examining Board

Name	Title	Signiture	Date
____________________	__________	__________	__________
____________________	__________	__________	__________
____________________	__________	__________	__________

Adama-Ethiopia

DECLARATION

I declared that this thesis is my original work and has not been presented for a degree in any other Universities, and all sources of material used for the study have been accordingly acknowledged.

____

Yohannes Ephrem Djaleta

May, 2017

This Thesis work has been submitted for examination with my approval as a University advisor.

____

Dr. Vuda Srinivasarao

May, 2017

DEDICATION

I would like to dedicate this thesis to all those who contributed to the fruition of this work, especially my mother and all my family members. Thank you and May **The Most High YAHUWA** bless you in every way.

ACKNOWLEDGMENT

Most of all, I would like to thank **The Most High YAHUWA**, who enables me to realize all my goals sofar.

I would first like to thank my thesis advisor Dr. Vuda Srinivasarao of the Computing Department at Adama Science and Technology University. His office was always open whenever I ran into a trouble or had a question about my research. He consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it.

I also owe a great deal for all who contributed by providing academic inputs to this thesis. These are: Dr. Mariamawit Yonatan from Pharmaceutical Department at Addis Ababa University and all those who work at Addis Ababa University, Arat Killo Branch, and Herbarium Section.

Last but not least, I want to thank my mother and all my family members who helped me to finish this marathon all the way. Besides the never ending care of The Most High YAHUWA, your support and care while conducting this research is precious.

LIST OF ABBREVIATIONS

RDMS Relational Database Management System

KR Knowledge Representation

AI Artificial Intelligence

KBS Knowledge Base System

SPARQL SPARQL Protocol and RDF Query Language

DL Description Logic

OWL Web Ontology Language

RDF Resource Description Framework

RDFS Resource Description Framework Schema

KE Knowledge Engineering

SQL Structured Query Language

SNOWMED CT Systematized Nomenclature of Medicine Clinical Terms

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES AND GRAPH

ABSTRACT

Traditional herbal medicines are heavily used in third world nations to treat different types of diseases and Ethiopia is no exception. Even though usage of western medicine is prevalent, Ethiopians still rely more on traditional medicine. Garlic, ginger and others used to treat commonly recurring illnesses such as cold. Traditional medicine is very complex and wide spread In Ethiopia. Different ethnic groups uses different variety of herbs to cure diseases. However, there is a lack of computable data that contain plant species that are used as herbal medicine. This creates a struggle for researchers to gather and compile information about traditional medicine that can be used for different research purposes.

The purpose of this thesis is to model an otology based KR prototype by employing an ontology structure for Ethiopian Traditional Herbal Medicine. This allows domain experts who engaged in this specific area to capture and record semantically rich knowledge. Since it uses the OWL Semantic Web standard frame work, it can be integrated with in a linked-data platform to be shared over the web. The flexibility and scalability of ontology schema can greatly enhances the richness and expressiveness of the KR ontology model. Besides representing the domain area, the computing community in Ethiopia can take this research work as a supportive material in order to introduce ontologies as a capable KR modeling tool. The open source ontology editor and management platform called Protégé is used to build the prototype.

SPARQL query scripts are generated to extract a specific sets of information in the prototype knowledge base. Since SPARQL is the standard query language of the Semantic Web applications, it's very important to utilize it in the context of this thesis. Another very easy querying mechanism is the DL Query which is integrated with in the Protégé platform. This particular querying method allows a domain expert to formulate a query script and extract a specific piece of information from the ontology based KR model. Visualization is also important as it depicts the overall or specific architecture of the system in a glance. Third party application called AllegroGraph is used to generate the SPARQL query as well as the visualization that accompanies it.

The proposed prototype ontology is qualitatively evaluated by domain experts and it showed promising results in representing the domain knowledge. Eighty nine percent acceptability rate is gained after the evaluation session is done.

CHAPTER ONE
INTRODUCTION

1.1 Background

In today's data hungry world, finding an efficient, scalable, interoperable and easy knowledge representation method is very essential. These days, ontologies are gaining ground in capturing and representing domain knowledge. Semantics, flexibility, Interoperability and fast-learning curve are among the advantages that ontologies provide us over the other traditional knowledge capturing methods. Unfortunately, especially in the third world countries, employing ontologies for knowledge representation purpose is an almost non-existent phenomenon. The Purpose of this thesis it to showcase the benefits of using ontologies as a knowledge management system. The Ethiopian Traditional Herbal Medicine domain is used to build a prototype knowledge bases system which employs ontologies.

Currently ontologies are being used in many applications. Web Semantics and Artificial Intelligence are the major fields of study that use ontologies to capture and represent domain knowledge. Semantic Web is an area which focuses on meaning rather than word matching. It means that when we search for a particular subject or term on the web, web servers return results based on phrase or key-word matching rather than meaning or context. Producing search results based on meanings rather than text matching is what semantic web does. Imagine if a web server understands the meaning behind a phrase such as "I love Science" rather than just match the three words from a giant corpus in the index. Ontologies are the most important features of semantic web technologies. They are used to construct knowledge bases that are complex in nature. Ontologies allow sematic web applications to construct an ontology based KR model that understands contexts. By using ontologies we can create or model different domains in order to suit our needs.

Ontologies are composed of several components that are used to model a specific domain knowledge. These components include entities, classes, and relationships among classes. These components of an ontology model allow a certain KR systems to capture and record domain knowledge. These ontology based KR models can then be queried by users to answer complex questions, provide semantically rich and context aware results.

The volume of data consumption these days is very staggering. Today we are likely to use more data in a single day than people who accessed data in a life time a decade ago. Multiple data formats are making next to impossible in understanding the relationship among these different data sources.

Ontologies provide a remedy for undefined data or concept relationships. They are captured in a way that relationships between these classes are visible and understandable. **Classes** and **relationships** are at the heart of the two major standard ontology languages. These ontology languages are Resource Description Framework (RDF) and Web Ontology Language (OWL).

The combination of classes and relationships is known as a **Triple**. A triple is consisted of a **Subject**, a **Relation Ship** and an **Object**. Both OWL and RDF uses this template to construct and process ontologies. Below is a nice and simple example that describe the structure of a triple.

These days' ontologies are deployed by big companies such as Yahoo and Amazon. For instance Yahoo uses ontologies as taxonomies to categorize Websites. On the other hand, Amazon uses them to categorize products for sale and features.

1.2 Motivation

Many areas such as health, e-commerce and higher education research projects are reaping the benefits of ontologies as a knowledge capturing and representing schemes. This capability of ontologies allow us to model knowledge bases in different types of domains. Ontologies are part of the W3C standards stack for the Semantic Web, in which they are used to specify standard conceptual vocabularies to exchange data among systems, provide services for answering queries, publish reusable knowledge bases, and offer services to facilitate interoperability across multiple, heterogeneous systems and databases [1].

Almost all **AI** systems including mobile personal assistance for the iPhone called Siri uses ontology driven data to provide digital assistances to its users. Traditional data capturing mechanisms such as Relational Database Management Systems lack the smartness and flexibility of ontologies. The scientific community generally considers ontologies as the best method for representing world reality due to their capacity for semantic modeling of concepts. Ontologies use classes, properties, instances, aggregation relations, generalization relations, etc., and especially axioms represented mainly with logic languages, as descriptive logic or first order logic to add semantics to the models [2].

Ontologies give the same credence for model (schema) as to the data itself, while in **RDMS** the schema is secondary to the data. This means once the schema of RDMS set, it cannot change or evolve at will. This makes RDMS a much faster platform for simple data extraction purpose compare to ontologies despite its rigid schema. On the other hand, ontologies are much more expressive in nature because of the flexibility of their schemas as compare to RDMS. This makes them more suitable for constructing easy and quick scientific research KBS applications as compared to RDMS.

The following figure give a brief pictorial description of the difference between ontologies and RDMS schema in representing the same concept of Hydraulics:

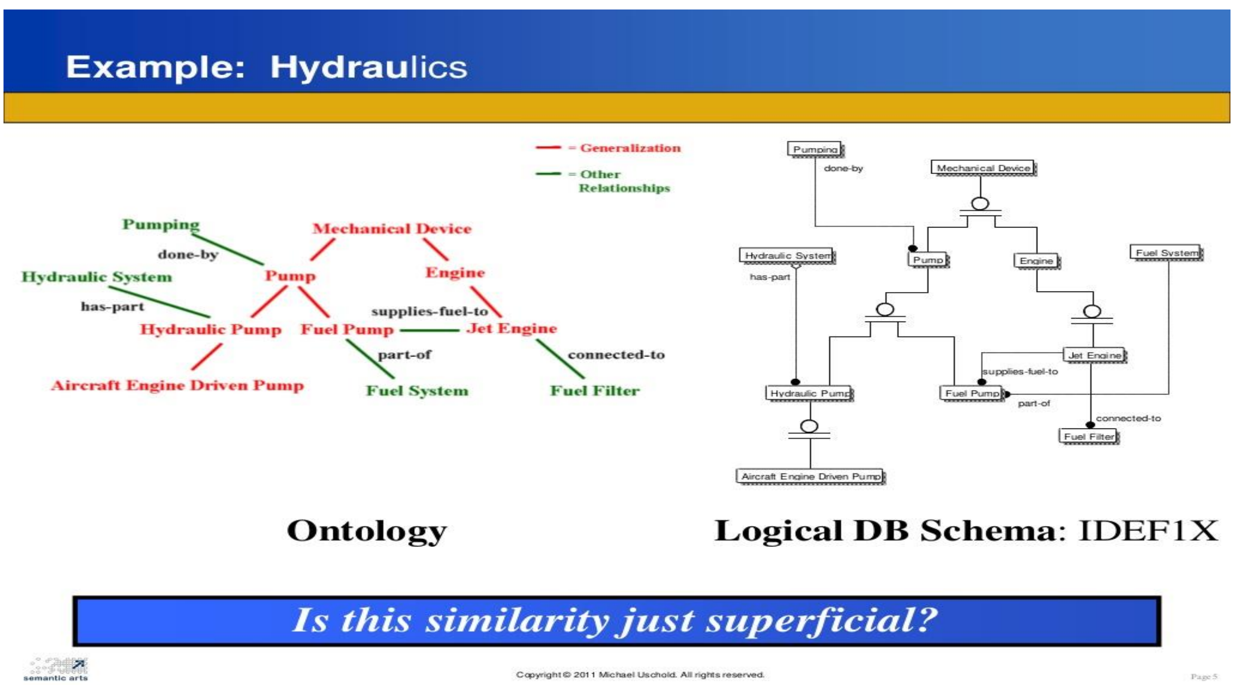


Figure 1: The difference between ontology and RDMS schema *[3]*

Modeling an ontology schema is a much easier task than that of RDMS. We can easily construct a quick mock-up schema for a particular research works or even a standalone KBS that can evolve its schema through time with limited restriction. Unlike RDMS, ontologies are semantically driven in expressing a certain world concept which can greatly help researchers to model their world view without hindrance. The semantic aspect of ontologies are essential to integrate them in a semantic web or linked-data applications.

In light of the above mentioned facts, the researcher's motivation to engage in this specific thesis arises from the fact that there is a lack of intelligent systems that use ontologies for knowledge management in Ethiopia. The availability of semantically driven apps that employ ontologies are almost non-existent in this country. The need to demonstrate the importance of ontologies for such a scenario is paramount.

The researcher believe that the prototype ontology developed for the Ethiopian traditional medicine domain in this thesis will help to stimulate the interest on ontologies.

1.3 Statement of the Problem

It's a well-known fact that Traditional Herbal Medicine are house hold sickness treatment mechanisms especially in third world nations. Here in Ethiopia, people are relying on local herbs for curing recurring sickness such as flu and common cold. These herbal plants are less invasive, carry much less side-effects and affordable compare to modern medicine for the average person.

Keeping in mind the above facts, here are brief descriptions of two problems identified as the reason for conducting this research:

- There are no knowledge base systems as well as centralized database set-ups that capture and record local traditional herbal medicines which are found in Ethiopia.
- The lack enough research works done in the area of Ontologies and Semantic Web as a whole in Ethiopian shows that there is an absence of familiarity in this subject matter.

1.4 Research Questions

The Following three questions describe why this research is intended to be conducted:

- What are the unique advantages of an ontology based KR system give us in representing the traditional herbal medicine domain?
- Can both SPARQL and DL query schemes adopted as a viable options for querying the ontology based KR system proposed in this research?
- Does the graph representation and visualization of the ontological model make the KR more intuitive and easy to access a specific data?

1.5 Objectives of the Study

1.5.1 General Objective

The general objective of this study is to develop a prototype ontology knowledge based model for Traditional Herbal Medicines found in Ethiopia and asses its feasibility.

1.5.2 Specific Objectives

- Conduct a thorough study on the domain knowledge of traditional medicine and its attributes.
- Model a blueprint of the ontology schema with all the classes, objects, attributes and relationships used for Ethiopian Traditional Herbal Medicine domain.
- Develop the prototype using the blue print model.
- Formulate a SPRQL and DL query scripts for querying the ontology.
- Asses the feasibility and the strength of the prototype by evaluating the performance of the ontology by using domain expert feedbacks.

1.6 Scope and Limitations of the Study

This thesis only deals with building the ontology framework and taxonomical structure for capturing and querying using the OWL ontology language. The ontology modelling and design is created with in the Protégé platform which is a free and open-source ontology engineering tool. Moreover, SPARQL queries are formulated through a third party application called AllegroGraph.

The main limitation of this thesis comes from a significant lack of data concerned with traditional herbal medicinal plants found in Ethiopia. The reason behind for the lack of data is that there are not enough organization responsible for actively gathering and recording herbal medicinal plants in Ethiopia.

1.7 Research Methodology

Raw files that are relevant for this research work has been collected from the archives of Addis Ababa University, Arat Kilo Branch, Biology Department. To be specific, the book Flora of Ethiopia and Eritrea 8th Edition, which is exclusively owned by the university and carries an extensive list of herbal plats found in Ethiopia and Eritrea is used as a data source for the ontology. To enhance the data gathered from the book, a web portal called **Useful Tropical Plants** is used to extract supportive data concerning the traditional herbal medicines found in Ethiopia.

Before building the knowledge base ontology prototype system is considered, several ontology languages has been taken in to consideration. However, the OWL ontology language is selected for its expressiveness, open world assumption and ease of use in modeling a certain domain. All the classes, objects and relationships are carefully defined in order to construct the ontology based KR system.

Since efficient and easy querying is one of the most important aspects of ontology based KBS, DL and SPARQL querying mechanism has been used for data extraction. DL query is used for quick and easy querying for users of the system. But SPARQL is also being used because it is the de-facto query language for OWL as well as RDF applications which reside in multiple platforms. A third party application called AllegroGraph is used to formulate SPARQL queries.

1.8 Significance of the Study

These days many AI systems and semantic web platforms are using ontologies extensively to represent a knowledge. Semantic web technologies, dubbed as web 3.0, is used for extracting context aware information from hugely cluttered and semi-structured data from the web. Ontologies are the back bone of semantic web technologies such as knowledge base search portals. These KR platforms allows us to construct complex queries and retrieve contextually relevant answers. One major example of a semantic search platform on the web is DBPedia. It uses every structured info-boxes available in Wikipedia pages to form a knowledge-base. It is available for public consumption and anyone can access it and query the DBPedia knowledge base using SPARQL for free.

Not many knowledge bases are using ontologies commercially or in educational institutions either in Ethiopia. Researchers need a platform that create a collaborative and highly scalable systems. Ontologies are a perfect tool for research works as well as enterprise application development that employ expansive KR system. This research will hopefully ignite an interest on ontologies and encourage different organizations such as higher educational institutions to develop ontology applications for different domains. The researcher believes that the success of this thesis and other handful research woks in this area will play a role in the adaptations of ontologies as tools to build KR systems.

1.9 Organization of the Thesis

The second chapter of this thesis contains citations of reviewed literatures that mostly deal with ontology and Web Sematic terms. Industry standards of ontologies such as RDF and OWL are dealt extensively in the literature review part. The third chapter discusses about the methodology of the research. This part includes the overview of the proposed system, data and prototype design. The fourth chapter discusses about data sources, data quality, data preparation, experimentation and prototype evaluation. Finally, the last chapter contains the conclusion, recommendation and future work.

CHAPTER TWO

LITERATURE REVIEW

This chapter discusses about all the reviewed literature which is concerned with knowledge representation. Topics which are related to KR such as knowledge acquisition, inference and validation are briefly elaborated here. Other relevant topics such as Ontologies and Semantic Web are also cited after reviewing several literatures.

2.1 Overview of Knowledge-Representation System

A knowledge representation is most fundamentally a surrogate, a substitute for the thing itself, used to enable an entity to determine consequences by reasoning about the world. It is a set of ontological commitments, i.e., an answer to the question: In what terms should I think about the world? It is a fragmentary theory of intelligent reasoning, expressed in terms of three components:

- The representation's fundamental conception of intelligent reasoning;
- The set of inferences the representation sanctions;
- The set of inferences it recommends. [4].

The best option to understand what Knowledge Representation is simply to mention what it is intended for. Its mission is to make knowledge as explicit as possible. This is necessary because knowledge is stored in implicit form, i.e. tacit knowledge non-observable from the outside, inside minds and spread around in community social habits. To facilitate knowledge sharing it is necessary to make it explicit [5].

The objective of a ***knowledge representation*** is to express knowledge in a computer tractable form, so that it can be used to enable our AI agents to perform well. A ***knowledge representation language*** is defined by two aspects:

1. **Syntax:** The syntax of a language defines which configurations of the components
of the language constitute valid sentences.

2. **Semantics:** The semantics defines which facts in the world the sentences refer to, and hence the statement about the world that each sentence makes. This is a very general idea, and not restricted to natural language.

Suppose the language is arithmetic, then ‘x’, ‘^{3}’ and ‘y’ are *components* (or symbols or words) of the language the *syntax* says that ‘$x\ ^3\ y$’ is a valid sentence in the language, but ‘$^3\ ^3$ x y’ is not the *semantics* say that ‘$x\ ^3\ y$’ is false if y is bigger than x, and true otherwise [6].

A Knowledge Representation or KR is the process of designing an expert system. It consists of three stages. These are knowledge acquisition, knowledge representation and knowledge validation.

In expressing the basic tenants of what role a *Knowledge Representation* can play Randall Davis, Howard Shrobe, and Peter Szolovits states the following “

- A knowledge representation (KR) is most fundamentally a surrogate, a substitute for the thing itself, used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than taking action in it.
- It is a set of ontological commitments, i.e., an answer to the question: In what terms should I think about the world?
- It is a fragmentary theory of intelligent reasoning, expressed in terms of three components: (i) the representation's fundamental conception of intelligent reasoning; (ii) the set of inferences the representation sanctions; and (iii) the set of inferences it recommends.
- It is a medium for pragmatically efficient computation, i.e., the computational environment in which thinking is accomplished. One contribution to this pragmatic efficiency is supplied by the guidance a representation provides for organizing information so as to facilitate making the recommended inferences.
- It is a medium of human expression, i.e., a language in which we say things about the world [7].”

An expert (knowledge based) system is a problem solving and decision making system based on knowledge of its task and logical rules or procedures for using knowledge. Both the knowledge and the logic is obtained from the experience of a specialist in the area (Business Expert). An expert system emulates the interaction a user might have with a human expert to solve a problem.

During the mid-1960's knowledge representation slowly emerged as a separate area of study. Several different approaches to knowledge representation began to manifest themselves and have resulted in the various formalisms in use today. The most important current approaches are semantic networks, first-order logic, frames, and production systems [8].

Over the years, perhaps the most popular kind of data structure has been a hierarchy of nodes connected by IS-A links. IS-A is a term used to describe the existence of a generalization relation between a sub concept like "dog" and its super concept "mammal," i.e., "Dog IS-A mammal." The most popular kind of inference has involved the inheritance of information from the top levels of the hierarchy downward along these IS-A links. Such an organization allows information to be shared among many nodes and thus leads to large-scale representational economies [9].

2.2 Objectives of Knowledge-Representation

- Rich enough to express the knowledge needed to solve the problem.
- As close to the problem as possible; it should be compact, natural, and maintainable. It should be easy to see the relationship between the representation and the domain being represented, so that it is easy to determine whether the knowledge represented is correct. A small change in the problem should result in a small change in the representation of the problem.
- Amenable to efficient computation, which usually means that it is able to express features of the problem that can be exploited for computational gain and able to trade off accuracy and computation time.
- Able to be acquired from people, data and past experiences [10].

The driving preoccupation of the field of knowledge representation should be understanding and describing the richness of the world. Yet in practice research that describes itself as core knowledge representation work has concentrated nearly all of its efforts in a much narrower channel, much of it centered around taxonomic and default reasoning.

It is not an accident that useful insight about finding a good set of temporal abstractions came from close examination of a realistic task set in a real-world domain. It underscores our conviction (shared by others) that attempting to describe the richness of the natural world is the appropriate forcing function for KR work

Our point here concerns both labeling and methodology: (i) work like and should be recognized by the KR community as of central relevance to KR research, not categorized as diagnosis or qualitative physics and seen as unrelated, and (ii) insights of the sort obtained in and come from studying the world, not from studying languages. We argue that those who choose to identify themselves as KR researchers should be developing theory and technology that facilitates projects like these; and conversely those who are building projects like these are engaged in a centrally important variety of KR research.

While tools and techniques are important, the field is and ought to be much richer than that, primarily because the world is much richer than that. We believe that understanding and describing that richness should be the central preoccupation of the field [11].

2.3 Components of Knowledge Engineering

The process of acquiring knowledge from experts and building a knowledge base is called **knowledge engineering.** Knowledge engineering involves the cooperation of human experts in the domain working with the knowledge engineer to codify and make explicit the rules (or other procedures) that a human expert uses to solve real problems. Knowledge engineering can be viewed from two perspectives: narrow and broad.

According to the narrow perspective, knowledge engineering deals with **knowledge acquisition**, **representation**, **validation**, **inference**, **explanation**, and **maintenance**. Alternatively; according to the broad perspective, the term describes the entire process of developing and maintaining intelligent systems. In this book, we use the narrow definition [12].

As we can see above KR is one of the integral components of knowledge engineering. Broadly speaking KR mainly deals with **knowledge acquisition**, **knowledge inference** and **knowledge validatio**n.

The figure below illustrates the instances and relationship among the components that form a knowledge engineering frame work.

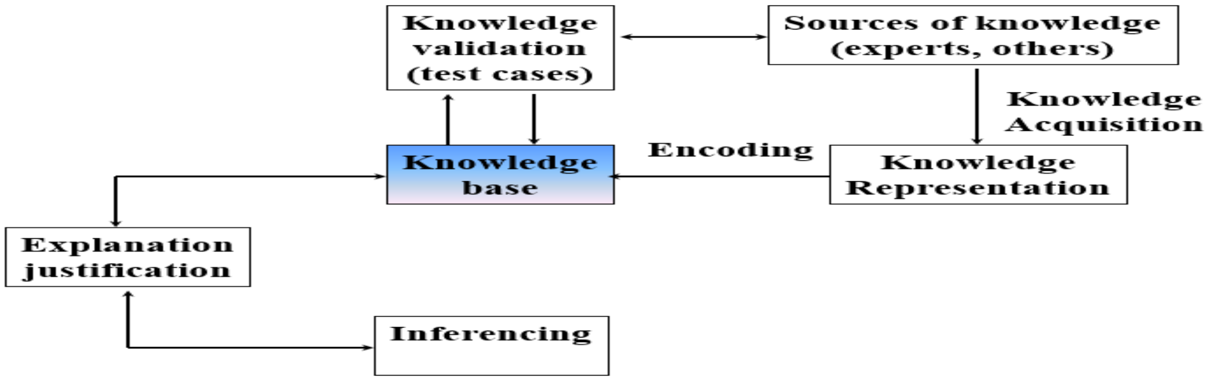


Figure 2: Diagrammatical depiction of the relationship of different KR components *[13]*

2.3.1 Knowledge Acquisition

Knowledge acquisition is the process of adding new knowledge to a knowledge base and refining or otherwise improving knowledge that was previously acquired. Acquisition is usually associated with some purpose such as expanding the capabilities of a system or improving its performance at some specified task. It is goal oriented creation and refinement of knowledge. It may consist of facts, rules, concepts, procedures, heuristics, formulas, relationships, statistics or other useful information. Sources of this knowledge may include one or more of the following.

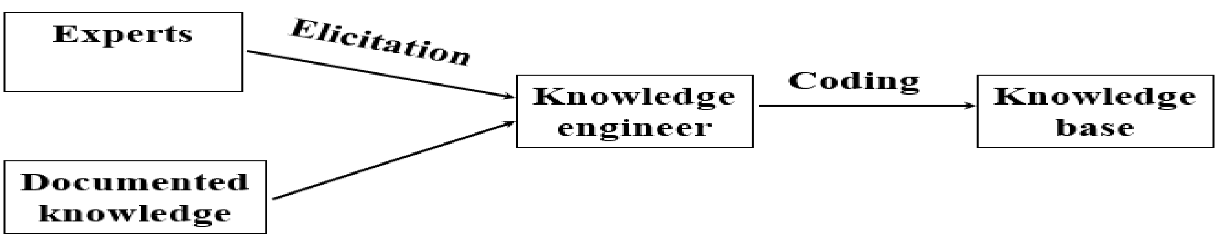


Figure 3: The development path that a particular knowledge base application has to follow *[14]*

Classification or **taxonomy** of learning types serves as a guide in studying or comparing a differences among them. One can develop learning taxonomies based on the type of knowledge representation used (predicate calculus, rules, frames), the type of knowledge learned (concepts, game playing, problem solving), or by the area of application (medical diagnosis, scheduling, prediction and so on).

The classification is intuitively more appealing and is one which has become popular among machine learning researchers. It is independent of the knowledge domain and the representation scheme is used. It is based on the type of inference strategy employed or the methods used in the learning process. The five different learning methods under this taxonomy are: **Memorization** (**rote learning**), **direct instruction** (**by being told**), **analogy**, **induction**, and **deduction.**

According to slide share web site depicted on the Artificial Intelligence, Knowledge Acquisition slide, Knowledge is broadly categorized under the following two categories:

Declarative knowledge: - Tells us out about facts

- Facts about objects and relationships
- Descriptive representation of knowledge
- It is often shallow knowledge

Procedural knowledge: - Tells us what to do

Knowledge about procedures involved in solving problems [15].

2.3.2 Knowledge Inference

Knowledge inference refers to acquiring new knowledge from existing facts based on certain rules and constraints. One way of representing these rules and constraints is through the use of logic rules, formally known as *knowledge representation*. The mechanism behind inferring new knowledge based on the existing facts and logic rules is typically known as *reasoning*. By encoding biological properties in the form of logic rules, the facts inferred by automated reasoning can be more biologically meaningful [16].

An **Inference Engine** is a tool from artificial intelligence. The first inference engines were components of expert systems. The typical expert system consisted of a knowledge base and an inference engine. The knowledge base stored facts about the world. The inference engine applied logical rules to the knowledge base and deduced new knowledge. This process would iterate as each new fact in the knowledge base could trigger additional rules in the inference engine. Inference engines work primarily in one of two modes either special rule or facts: forward chaining and backward chaining. Forward chaining starts with the known facts and asserts new facts. Backward chaining starts with goals, and works backward to determine what facts must be asserted so that the goals can be achieved [17].

When rules are examined by the inference engine, actions are executed if the information supplied by the user satisfies the conditions in the rules. Two methods of inference often are used, *forward* and *backward chaining.* Forward chaining is a top-down method which takes facts as they become available and attempts to draw conclusions (from satisfied conditions in rules) which lead to actions being executed. Backward chaining is the reverse. It is a bottom-up procedure which starts with *goals* (or actions) and queries the user about information which may satisfy the conditions contained in the rules. It is a verification process rather than an exploration process. An example of backward chaining is MYCIN, and an example of forward chaining is Expert. A system which uses both is Prospector [18].

2.3.3 Knowledge Validation

Among the software community, validation is interpreted as *"building the right product"*, verification as *"building the product right".* After we assume the validation process can be considered as some determined composition of two kinds of tasks:

- Activities that intend to reach the structural correctness of the KB *(verification),*
- Activities that intend to demonstrate the KB ability to reach correct conclusions *(evaluation).*

On the other hand, validation refers to different components of a knowledge-based system. We can validate a knowledge base, inference engine, a user interface etc [19].

Validation and verification (V&V) comprise a set of techniques used in software engineering (and, therefore, in knowledge engineering) to evaluate the quality of software systems (including KBS). There is often confusion about the distinction between validation and verification, but the conventional view is that *verification* is the process of checking whether the software system meets the *specified* requirements of the users, while *validation* is the process of checking whether the software system meets the *actual* requirements of the users [20].

This is no less important where KBS technology is used in knowledge management: Validation &Verification techniques provide ways to measure the quality of knowledge in a knowledge base, and to indicate where work needs to be done to rectify anomalous knowledge. In this context, verification tells us whether or not the knowledge bases are flawed as software artifacts, while validation tells us whether or not the content of the knowledge base accurately represents the knowledge of the human experts that supplied it. Both are clearly important. It is worth noting in this context that verification is essentially an objective test: there are absolute measures of the correctness of a piece of software. However, validation is typically subjective to a certain extent, where we must compare formally-represented knowledge to informal statements [21].

2.4 Semantic Web and Ontologies

2.4.1 Introduction to the Semantic Web

The **Semantic Web** is an extension of the Web through standards by the World Wide Web Consortium (W3C).The standards promote common data formats and exchange protocols on the Web, most fundamentally the Resource Description Framework (RDF). According to the W3C, "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries" [22].

The basic idea of the Semantic Web is to describe the meaning (i.e. the semantics)
of Web content in a way that can be interpreted by computers. Programs can then exploit (more) human knowledge to execute tasks, enabling powerful applications not only on the Web. The first step for this is to cast knowledge into a machine-processable form. The resulting descriptions are often called ontologies, and the machine-readable formalisms that they are based on are called ontology languages [23].

One of the approaches of the semantic web according to Pascal Hitzler, Markus Krötzsch and Sebastian Rudolph on their book Foundations of Semantic Web is the vision of representing knowledge in a way that allows machines to automatically come to reasonable conclusions, maybe even to "think," has been a driving force for decades of research and development, long before the WWW was imagined [24].

The Semantic Web currently cannot exist without the implementation of ontologies which is the back bone of all the semantic web applications today. The following citations depicts the basic precepts of ontologies and their use.

2.4.2 Ontologies

The Internet and other open connectivity environments create a strong demand for sharing the semantics of data. *Ontologies* are becoming increasingly essential for nearly all computer science applications. Organizations are looking towards them as vital machine-processable semantic resources for many application areas. An ontology is an agreed understanding (i.e. semantics) of a

certain domain, axiomatized and represented formally as logical theory in the form of a computer based resource. By sharing an ontology, autonomous and distributed applications can *meaningfully* communicate to exchange data and thus make transactions *interoperate* independently of their internal technologies.
Research on ontologies has turned into an interdisciplinary subject. It combines elements of Philosophy, Linguistics, Logics, and Computer Science. Within computer science, the research on ontologies emerged “mainly” within two sub-communities: artificial intelligence (among scientists largely committed to building shared knowledge bases) and database (among scientists and members of industry who are largely committed to building conceptual data schemes, also called semantic data models. This legacy in computer science brings indeed successful techniques and methods to enrich the art of ontology engineering [25].

In computer science and information science, an **ontology** is a formal naming and definition of the types, properties, and interrelationships of the entities that really or fundamentally exist for a particular domain of discourse. It is thus a practical application of philosophical ontology, with a taxonomy.

An ontology compartmentalizes the variables needed for some set of computations and establishes the relationships between them.

The fields of artificial intelligence, the Semantic Web, systems engineering, software engineering, biomedical informatics, library science, enterprise bookmarking, and information architecture all create ontologies to limit complexity and to organize information. The ontology can then be applied to problem solving [26].

2.4.3 Ontology Languages

There are several markup based languages used for creating and developing an ontology but we cited the two most widely used languages used to construct an ontology. These are OWL (Web Ontology Language) and RDF (Resource Description Framework).

2.4.3.1 OWL (Web Ontology Language)

The **O***ntology* **W***eb* **L***anguage* (OWL) is a set of markup languages which are designed for use by applications that need to process the content of information instead of just presenting information to humans.

The OWL ontology describe the hierarchical organization of ideas in a domain, in a way that can be parsed and understood by software.

OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web. OWL is part of the W3C recommendations related to the Semantic Web [27].

The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language such that knowledge expressed in OWL can be exploited by computer programs, e.g., to verify the consistency of that knowledge or to make implicit knowledge explicit. OWL documents, known as ontologies, can be published in the World Wide Web and may refer to or be referred from other OWL ontologies. OWL is part of the W3C's Semantic Web technology stack, which includes RDF, RDFS, SPARQL, etc.

The current version of OWL, also referred to as "OWL 2", was developed by the W3C OWL Working Group (now closed) and published in 2009, with a Second Edition published in 2012. OWL 2 is an extension and revision of the 2004 version of OWL developed by the W3C Web Ontology Working Group (now closed) and published in 2004. The deliverables that make up the OWL 2 specification include a Document Overview, which serves as an introduction to OWL 2, describes the relationship between OWL 1 and OWL 2, and provides an entry point to the remaining deliverables via a Documentation Roadmap [28].

2.4.3.2 RDF (Resource Description Framework)

Resource Description Framework is another language related to creating an ontology which is extracted from web sources. The following citations paints a picture on the purpose and use of RDF.

The **Resource Description Framework** (**RDF**) is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax notations and data serialization formats. It is also used in knowledge management applications.

The RDF data model is similar to classical conceptual modeling approaches such as entity–relationship or class diagrams, as it is based upon the idea of making statements about resources (in particular web resources) in the form of subject–predicate–object expressions. These expressions are known as *triples* in RDF terminology. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object. For example, one way to represent the notion "The sky has the color blue" in RDF is as the triple: a subject denoting "the sky", a predicate denoting "has the color", and an object denoting "blue". Therefore, RDF swaps object for subject that would be used in the classical notation of an entity–attribute–value model within object-oriented design; Entity (sky), attribute (color) and value (blue). RDF is an abstract model with several serialization formats (i.e., file formats), and so the particular way in which a resource or triple is encoded varies from format to format.

This mechanism for describing resources is a major component in the W3C's Semantic Web activity: an evolutionary stage of the World Wide Web in which automated software can store, exchange, and use machine-readable information distributed throughout the Web, in turn enabling users to deal with the information with greater efficiency and certainty. RDF's simple data model and ability to model disparate, abstract concepts has also led to its increasing use in knowledge management applications unrelated to Semantic Web activity [29].

2.5 Related Works

There are several related works which are relevant to this research work. One of these is done in India, Sri Ramaswamy Memorial University, Kattankulathur. Vavidu, G. and Waheeta Hopper [30] developed a knowledge framework for Indian Medicinal Plants (KIMP). Knowledge frame work includes the ontology creation, and user interface for querying the system. Jena is used to build semantic web applications with the ontology representation of Resource Description Framework (RDF) and Web Ontology Language (OWL). SPARQL Protocol and RDF Query Language (SPARQL) is used to retrieve various query patterns. Automated mapping is achieved by considering lexical and edge based relatedness. The user interface is demonstrated for five thousand concepts, which gives the related information from Wikipedia web page in three languages. Mapping recommendation by the lexical similarity Jaccard algorithm gives 27% and Jaro Winkler algorithm gives 60%. Edge based relationship using WuPalmer algorithm gives 93% mapping recommendation. These are analyzed and compared with our algorithm based on WuPalmer gives more specific mapping results than WuPalmer with 71% [31].

The other related work done on ontology development using the herbal medicine domain was produced by Mohd Suffian Sulaiman, Sharifalillah Nordin, ursuriati Jamil and Alfian Abdul Halin [32]. This paper presents a multi-modal ontology model that integrates textual and visual information of herbal medicinal plants. This model was designed base on the human perception of images where retrieval is performed at semantic level.

Two types of information extraction need to be done namely analyzing the textual description information and interpreting the semantic visual feature information. The textual description information describes the high-level concepts about the image

The visual features information describes the low-level feature information which can be distinguished based on color, shape and texture about the particular herbal medicinal plant image. Acts from monographs describing the particular herbal medicinal plant images [33].

An ontology with a set of individual instances is referred to as a knowledge base (Noy & McGuinness,2000). In order to construct the ontology, three important things must be defined which are classes, properties and instances. To determine the hierarchy of classes, a number of approaches can be used such as either top-down, bottom-up or combination of both approaches (Uschold & Gruninger, 1996) [34].

CHAPTER THREE
METHODOLOGY

3.1 Research Design

There are several justification scenarios on why the proposed ontology based KBS is preferred rather than let's say RDMS. The success of any ontology application is highly dependent on how well defined the integral components of the ontology such as classes, instances, relationships and etc. Ontology consistency is also as much an essential part as the other main components.

RDMS is a tried and tested data management system that can capture data and present it for efficient and fast querying. Yet, in terms of providing semantically rich and highly expressive data, ontology driven KBS applications are much more useful than RDMS. Moreover, the flexibility provided using ontology in terms of schema evolution is another advantage that sways the favor for ontologies. This aspect of ontologies enable them to scale their schemas whenever there is a need to add new concepts and enrich the meanings of the knowledge base.

Biometric initiatives such as the Gene Ontology framework and SNOWMED CT which is used for clinical terms are testaments of ontological application used today. SNOMED CT (Systematized Nomenclature of Medicine -- Clinical Terms) is a standardized, multilingual vocabulary of clinical terminology that is used by physicians and other health care providers for the electronic exchange of clinical health information. [35]. SNOWMED CT is very scalable because of the flexible nature of ontology schemas.

3.2 Process Model

The domain area chosen for this research also benefits from the scalability and expressiveness of the ontology schema. Capturing and recording herbal medicines used throughout a big country like Ethiopia, with very diverse types of vegetation, while introducing new concepts and semantic data related to the domain is not possible with RDMS. This is due to mainly because the government and private organizations who're working on this domain don't have fixed set of vocabularies. This characteristics of the domain needs flexibility in order to accommodate new concepts. RDMS needs a rigid and unchanged schema but ontology fit perfectly to the ever changing scenario of the domain mentioned above.

As it is alluded to earlier, the lack of common and well-structured vocabulary terms dedicated for herbal traditional plants in Ethiopia is one of the launch-pads for this research work. This KBS can be a great platform for traditional healers as well as modern medicine practitioners in contributing different sets of vocabularies, terms and annotations to the KBS. This enables the KBS to grow semantically in describing certain set of terms.

3.2.1 Practice to be preserved from the existing system

Since the domain area revolves around herbal plants and their medicinal use, we retained some aspects of the subjects which are relevant for traditional medicinal plants. These are:

- Plant family, their distribution and characteristics
- Plant Species and their attributes.
- Healing properties that these plant species are said to provide. These are anecdotal information most of the time.

3.2.2 The Proposed Framework Overview

The OWL framework is used to build the ontology in this research. The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language such that knowledge expressed in OWL can be exploited by computer programs, e.g., to verify the consistency of that knowledge or to make implicit knowledge explicit. OWL documents, known as ontologies, can be published in the World Wide Web and may refer to or be referred from other OWL ontologies [36].

Below is a representation of the OWL frame work. This figure includes the different components used in the ontology development using the OWL language.

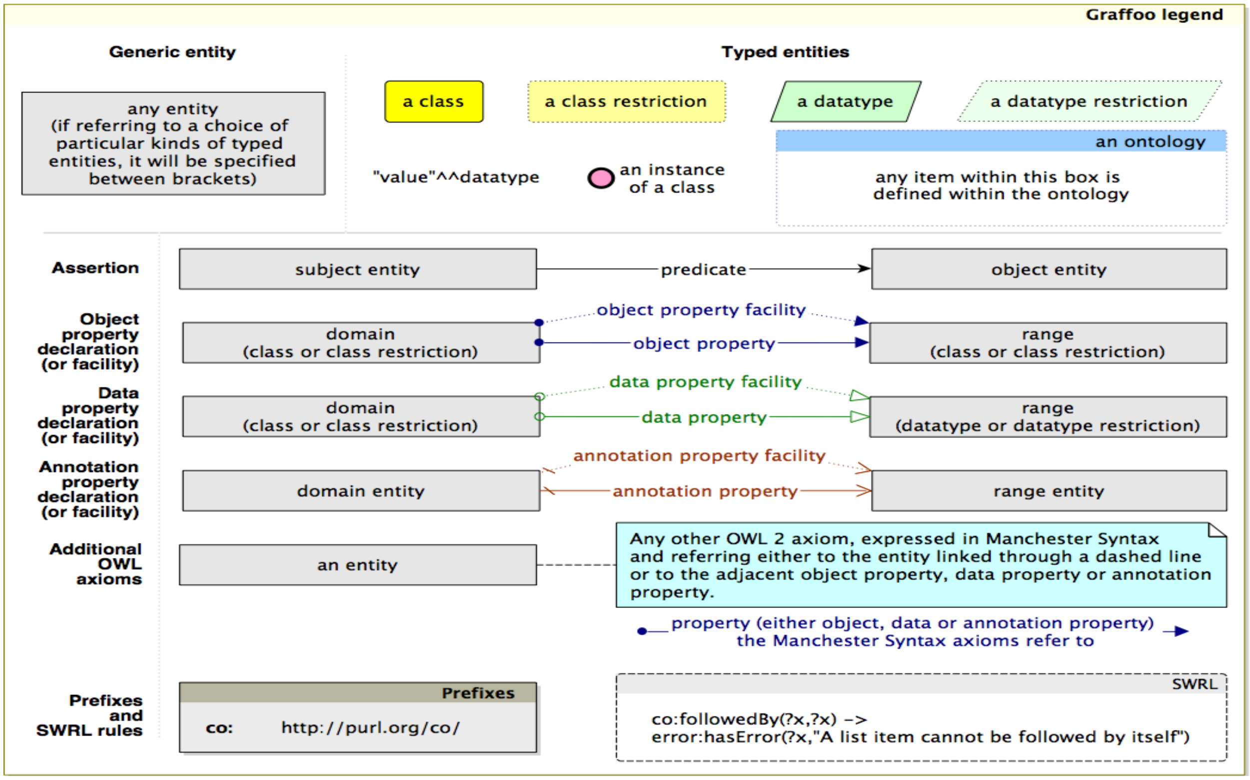


Figure 4: Graphical representation of an OWL framework *[37]*

The main attributes of the OWL ontology language are described in the list below:

- Class, objects, relationship, instances (individuals) and attributes are the major components for an ontology KBS.
- Classes are containers of instances also called individuals. These instances are the ones that contain the data pertaining to the Plant family, plant species, different types of ailments or diseases.
- Object properties create relationships between the main classes, subclasses and instances.
- Visualization gives snapshot of the entire class structure and relationships with subclass and instances graphically. These visualization components are integrated with in the Protégé frame work.

3.2.3 Proposed Framework Requirements:

1. **Functional Requirements**

   Functional requirements describe the main goals that this KBS ontology needs to accomplish and interaction scenarios between the KBS and end user. The following list explain what these requirements are:

- Model the schema of classes, objects, attributes (data properties) and relationships (object properties).
- Expand the schema whenever the need to add a new concepts in to the ontology.
- Capture and record plant specimens and their attributes which are discovered in the different areas of Ethiopia used for traditional medicinal purposes.
- Represent each plant specimens in terms of their relationship with other components of the ontology. This can be done with the object properties defined in the ontology.
- Query and retrieve data from the ontology using SPARQL and DL query that provide a semantically expressive results.
- Create a visualization of the entire as well as specific segments of the ontology in order to get a bird's eye view of what the ontology is composed of.

2. **None-functional requirements**

   These are perhaps not directly related to user-system interaction but they indirectly impact the quality of the system if not carefully considered. A couple of non-functional requirements in relation to the KBS ontology system are:

- Create a collaborative atmosphere by sharing it through the web for specific individuals or by making it public so that all who are interested in the domain can contribute. This allows the collaborators working on this ontology to contribute and expand the ontology further when there are new discoveries.
- User friendly platform is necessary especially for non-tech savvy researchers who work on the traditional medicine domain.

3.3 Data Source

In order to test the usability and performance of the prototype, raw data that can be populated with in the KBS ontology was needed. Yet, data pertaining to this domain are fragmented all over the place and it is very difficult to get hold on it. The problem of having an efficient and reliable data source is exacerbated by the fact that there are just a handful of governmental organization which hold botanical data concerned with traditional medicines. Despite several failed attempts to hold on a dataset useful enough to begin building the prototype, the Addis Ababa University, Arat Killo branch provided the researcher a book which contains an extensive list at around 9000 plant species. Among these species that are included in this book, there were several herbs that are discovered to be used as a medicinal plant throughout the country. So, the book called **Flora of Ethiopia and Eritrea** and is effectively the main source of data.

This book is seen several revision over time and the 8th volume which is the last edition of the book is used to extract the data sets. It contains plenty of details about plant species. These include all the taxonomy of botanical nodes such as family, genus and species.

3.4 Data Collection Techniques

As it is alluded to above, the researcher has used both primary and secondary sources. The secondary source are the data sets that are taken from the book Flora of Ethiopia and Eritrea. These datasets consisted of plant species, plant family, plant parts used for medicinal purposes and supportive annotations that describe the usage of these plant species.

Primary source data was used while conducting the prototype evaluation sessions. This is done by preparing questions that are believed to be representative in evaluating a particular KBS system. Questioners was given to five carefully selected domain experts and qualitative feedback data was retrieved from the prototype evaluation session. Details pertaining to the evaluation questions and the analysis results can be found in the next chapter.

3.5 Issues of Reliability and Validity

The data collected and the methods used to conduct this research can have a major impact on its success. Even though there are several factors and variables which can greatly affect the results of this thesis, the following components are taken to hold significant weight for the success or failure of the research. These are:

- Reliability of the data collected.
- Methodology used for the research.
- Subjects used to conduct the evaluation of the prototype.
- The researcher’s level of knowledge in the domain area.

3.6 Development of the Prototype

The purpose of building a prototype is to test whether the proposed system is working as intended. A prototype is generally used to evaluate a new design to enhance precision by system analysts and users. Prototyping serves to provide specifications for a real, working system rather than a theoretical one. In some design workflow models, creating a prototype (a process sometimes called **materialization**) is the step **between** the formalization and the evaluation of an idea [38].

The prototype for this particular research is developed by following the ontology development life cycle. From defining the concepts of the ontology up until the creation of instances or individuals is the route that is followed by the researcher to establish the layout for the ontology. The Following figure illustrates the ontology lifecycle path.

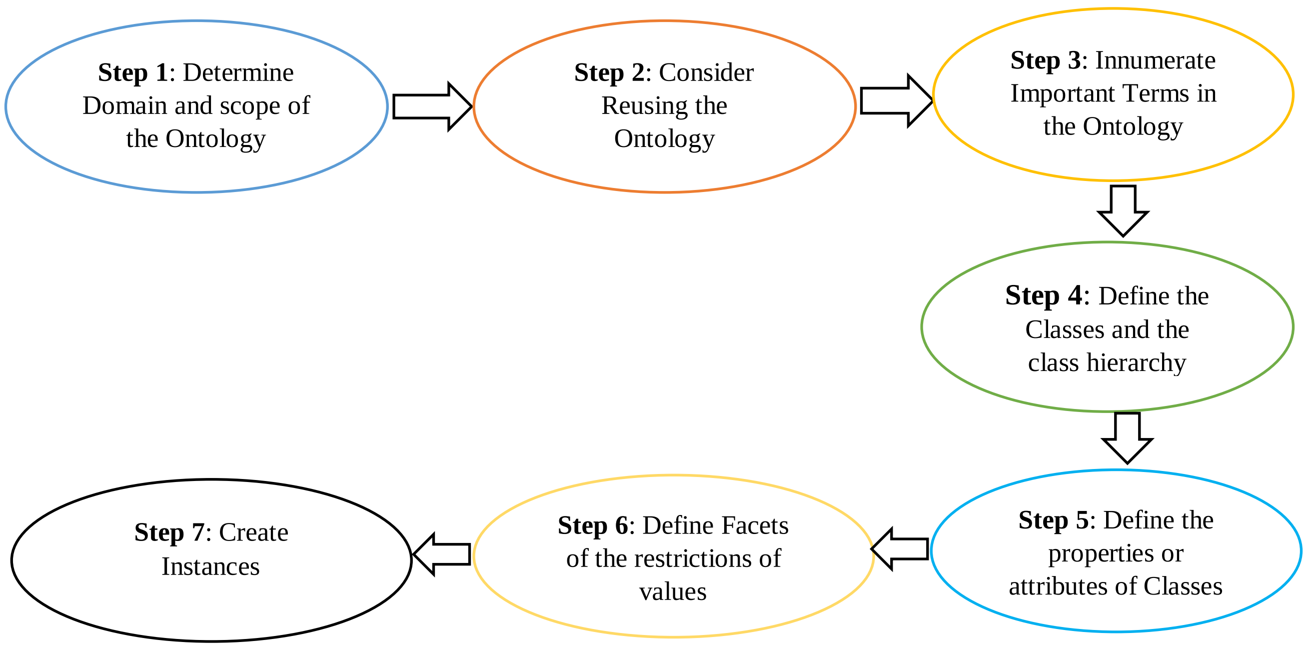


Figure 5: Ontology development life-cycle *[39]*

3.6.1. The Architecture of the System

Laying out the foundational architecture for the KBS ontology is important before building the actual ontology itself. Super classes, sub-classes, instances, attributes and relationships, which are the building blocks of ontological KBS has to be clearly defined first. Brief descriptions of these main components of an ontology presented below:

1. **Classes**: Classes are the core components of an ontology much like the classes you find in object oriented programming paradigm. They are also called concepts because they represent a collection or a group of different individuals that share a common characteristics. Classes can also be more or less specific in nature.

For example, (most) humans share certain characteristics, such as related DNA, a set of specific body parts, the ability to speak a complex language. Likewise, all mammals share these characteristics, except for the ability to speak [40].

A class or a concept can be a subclass or a sub concept of a superclass. This means if for instance Class 2 is a subclass of Class 1, then any individual or member of Class 2 is also a member of Class 1 by inheritance. The OWL ontology language allow us to explicitly state the class and subclass arrangement.

2. **Individuals:** also called instances are the ones that the ontology try to describe. They are related to tangible objects such as people, nations, mammals and etc. They also model abstract objects such as personal behaviors or politics.
3. **Relations:** In an ontology, relations are used to describe how one individual or instance is related to another. Although it is dependent on the ontology language, it is often possible to express different categories of relationships between Concepts. Consider, for example, “person has father person”. This is an **existentially quantified** relationship; it is the case that every person has a father, and that this individual is also a person. This can be contrasted from “person is father of person”; this is a **universal quantified** relationship. It is true that every individual which is father of a person is, themselves, a person; however, it would be wrong to assert that every person is the father of another [41].

In the case of the prototype developed for this research, we have two super classes that contain another three top-tier subclasses and 21 bottom-tier subclasses. All these classes contains 297 individuals or instances. Interms of relationships or Object Properties, there are **five relationship creators** and there are also **nine data properties** that describe the attributes of these instances. The figure below illustrates the main design layout or structure of the ontology. This model describes the relationships between the four top-most super classes.

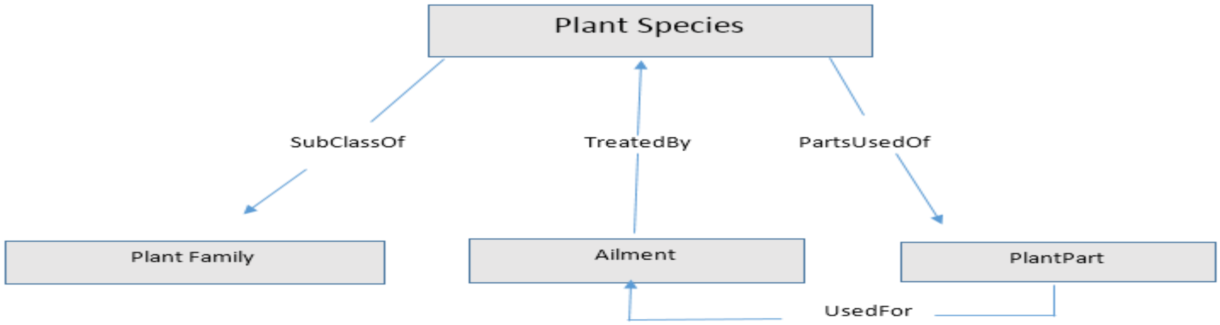


Figure 6: Ontological relationship structure between top-tier classes

To briefly describe the class concepts found in the above figure: the **Plant Species** class represents all the individual plant instances in the ontology. Moreover, the **PlantPart** class contains what Plant Parts are used to treat the illnesses and it is the sub class of the Plant Species class. The **Plant Family** class also contains instance of plant families which includes at least one Plant Species within them since the Plant Species class is a sub class of the Plant family class.

The **Ailment** class contains all the diseases instances which are thought to be cured by the plant species found in the ontology.

The relationships that exists between the classes are there to create the sematic link between the instances. For example the **TreatedBy** object property link can join illness called **x1** to a plant species called **y1**, so in this way it creates one semantically defined OWL structure.

All the above classes depicted in the model contain individuals or instances related to their respective concepts. For instance, the Ailment class includes twelve different subclass categories of diseases such as cancer, urological problems, neurologic diseases and etc. The plant species class contains 82 different medicinal plant species samples that are gathered from the book Flora of Ethiopia and Eritrea.

The ontology tree visualization found below is represented using the **is_a** relationship creator in order to show parent-child class relationship. As it can be witnessed also from the figure below, every ontology structure start with the **Thing** super class including the one we see below.

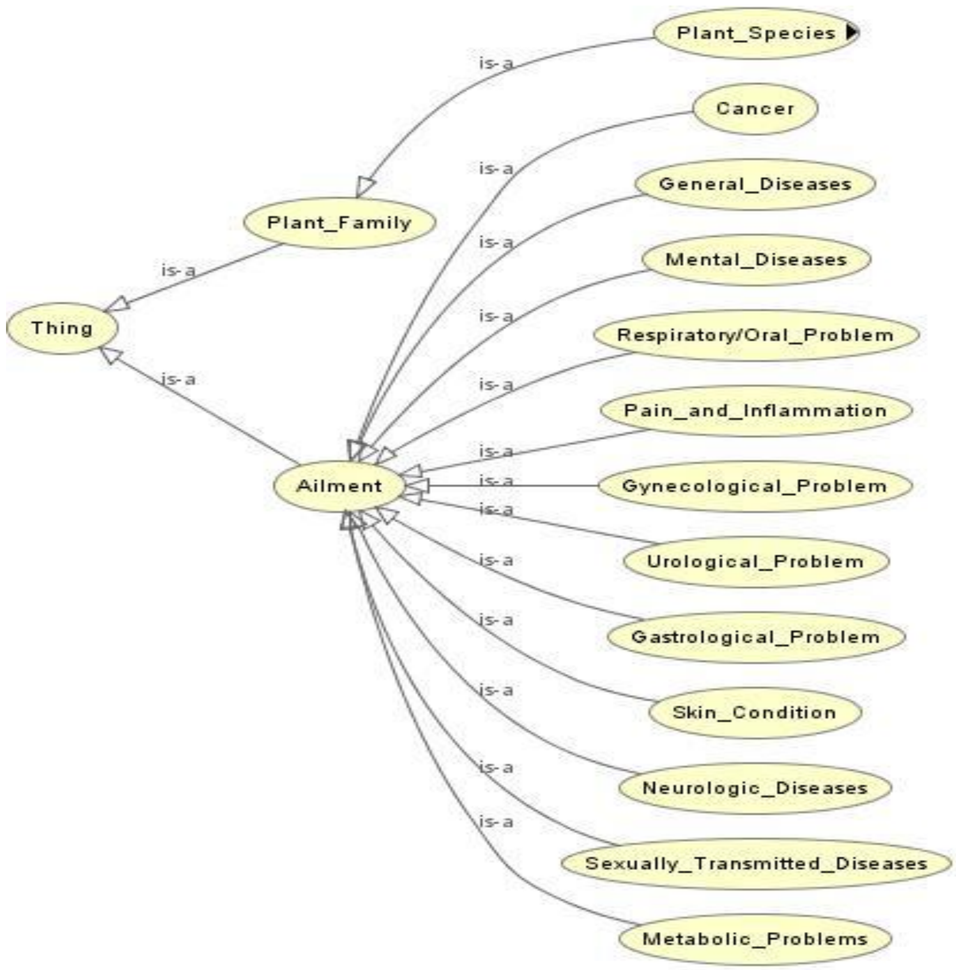


Figure 7: Ethiopian Herbal Medicinal Plants Ontology tree structure visualization

The overall system architecture for the ontology KBS depicts the several major components that are included in the system.

For instance the domain experts who interacts with the ontology should handle tasks related to the schema and other aspects of the knowledge base such as the class instances.

The following figure provide a snapshot of the system architecture of the prototype.

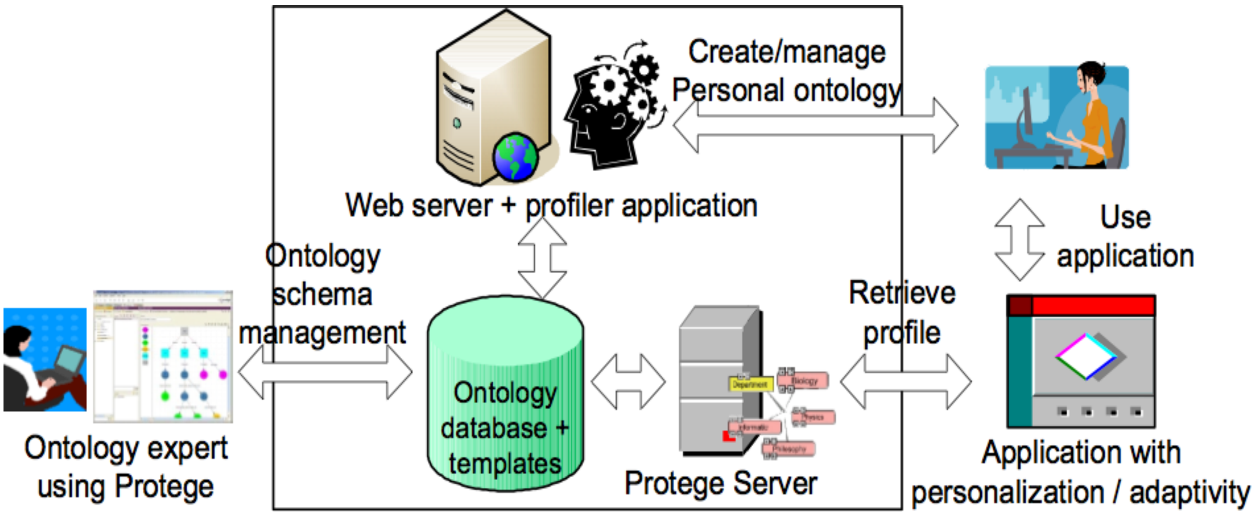


Figure 8: Overall System Architecture *[42]*

3.6.2. Programing Language Used and Justification

The programming language that is used to build the ontology is called OWL or Web Ontology Language. The reason it is called OWL is because it's better to pronounce the acronym by giving it a familiar name. OWL is issued by the World Wide Web Consortium (W3C), who standardize all web related protocols and languages.

The justification for using OWL rather than let's say RDFS or Resource Description Framework Schema as an ontology language is as follows:

I. **Rich Vocabulary:** OWL provides you a far better and very extensive vocabulary than RDFS.

II. **Rigidity:** When you're using OWL, you know what vocabulary to use and what not to use unlike RDFS. This aspect of OWL insures the integrity of the ontology. Integrity is very important in modelling the real world entities in order not to create an ambiguous semantics.

For building the prototype, the researcher selected a knowledge management platform called Protégé. **Protégé** is written in Java and it supports all the popular operating systems such as Windows and Mac OS. Protégé is being developed at Stanford University and is made available under the BSD 2-clause license [43]. There are several reasons why Protégé is used rather than other platforms out there. These are:

- Its free and open source platform.
- Its compliance with the W3C.
- Is optimized for collaborative activities through the web.
- Support for all semantic web stack including OWL.
- According Protégé's developers, it is supported by academic, government and corporate users.
- It is being used by more than 300,000 users to build smart applications and makes it the most widely used ontology modeling platform.

Easy and efficient querying ability is one of the main things to consider when we develop a KBS. Specific to ontologies, there are several querying schemes used to extract data. The following querying methods are used to retrieve data from the ontology.

There are other free and paid ontology building tools. The following table illustrates the general descriptions of other alternatives for ontology building and management.

Feature	**Appollo**	**OntoEdit**	**Protégé**	**Swoop**	**TopBraid Composer**
Developers	KMI(Open University)	Ontoprise	SMI(Stanford University)	MND(University of Maryland)	TopQuadrant
Availability	Open Source	Software License	Open Source	Open Source	Software License

Table 1: General description of the tools *[44]*

I. **SPARQL**: Is a standard language used to query all semantic applications that use ontology programming languages such as RDF and OWL. It's more like SQL query for relational databases application.

SPARQL allows users to write queries against what can loosely be called "key-value" data or, more specifically, data that follows the RDF specification of the W3C. The entire database is thus a set of "subject-predicate-object" triples. This is analogous to some NoSQL databases' usage of the term "document-key-value", such as MongoDB. RDF data can also be considered in SQL relational database terms as a table with three columns – the **subject** column, the **predicate** column, and the **object** column. Unlike relational databases, the object column is heterogeneous: the per-cell data type is usually implied (or specified in the ontology) by the predicate value.

Alternately, again comparing to SQL relations, all of the triples for a given subject could be represented as a row, with the subject being the primary key and each possible predicate being a column and the object is the value in the cell. However, SPARQL/RDF becomes easier and more powerful for columns that could contain multiple values (like "children") for the same key, and where the column itself could be a joinable variable in the query, rather than directly specified [45].

Yet, writing SPAPQL queries all the time may sometimes pose a difficulty for domain experts who are not adept at formulating string level queries. The difficulty of forming SPARQL queries rises when the complexity and the scale of the ontology grows. Here is where graphical SPARQL query creator platform called **AllegroGraph** comes handy. In this research we formulated SPARQL queries scripts using the **AllegroGrpah**. Moreover, the **AllegroGraph** platform provides diagrammatical visualization of the specifically formulated SPARQL queries. This feature enhances the data query experience even more and the platform itself is free. The queries themselves are discussed in the next chapter when we analyze and evaluate the prototype.

II. **DL Query**: For ease of use and convenience sake, we also used the DL Querying mechanism which is embedded with in Protégé platform. This querying mechanism allows the domain expert to retrieve data easily by linking instances with object properties to get a specific results. However, DL Query only works with in the Protégé frame work and is not as powerful as SPARQL for retrieving sematic data from different platforms. The results from both querying platform are discussed in the next chapter.

CHAPTER FOUR

PRESENTATION, ANALYSIS, AND INTERPRETATION OF DATA

4.1 Introduction

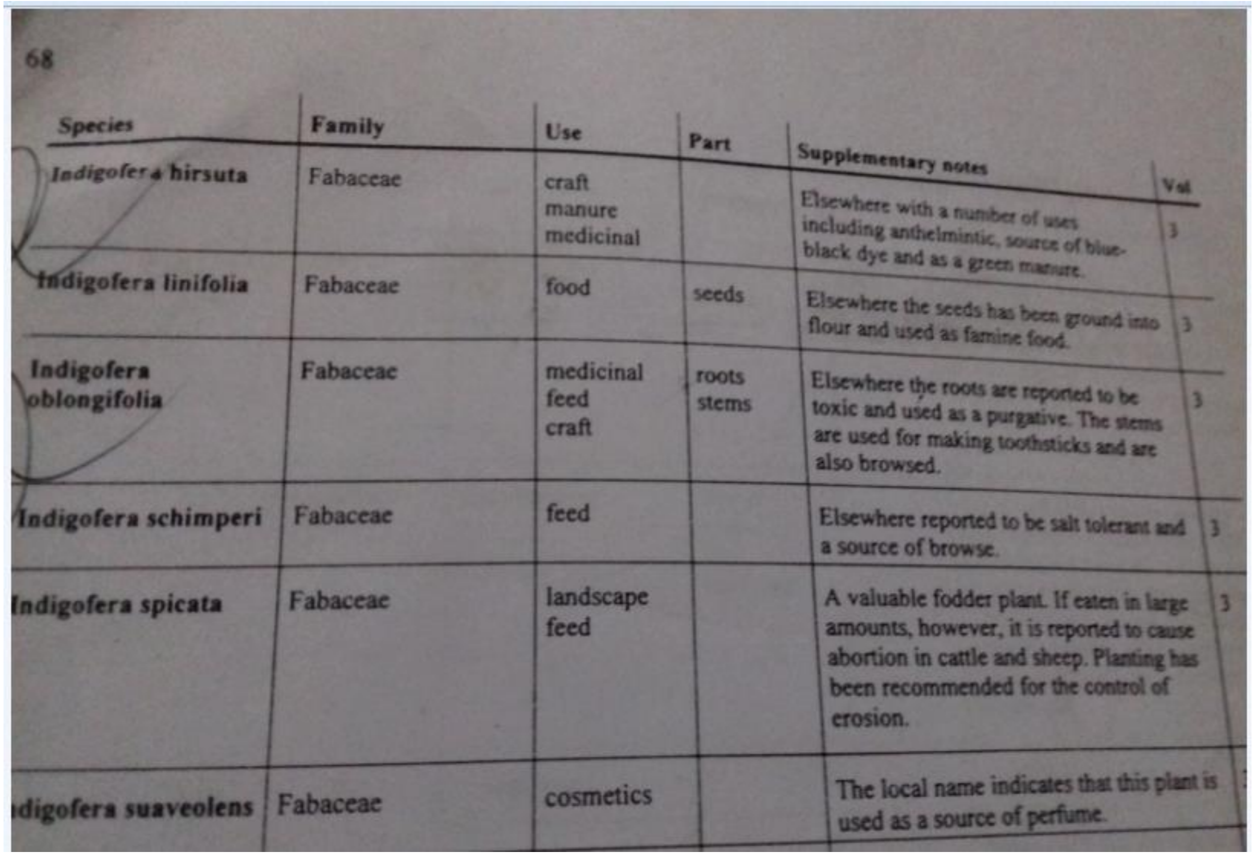
In this chapter, data and prototype related aspects are discussed. In terms data, the different components of the dataset gathered from secondary sources are described here. As it is mentioned in the previous chapter, the data set used to populate the KBS ontology mainly comprised of data from the book called Flora of Ethiopia and Eritrea, which is owned by the Addis Ababa University, Arat Kill Branch.

The other aspect of this chapter is the prototype developed to represent the Ethiopian Traditional Herbal Medicine domain. Querying and data visualization facets are subjects for analysis and interpretation. After that, evaluation of the prototype which is conducted in order to ascertain its performance is also discussed.

4.2 Description of the data collected

One major difficulty faced during the data collection phase was finding a reliable source that can provide a substantial and accurate data in relation to the domain. The main reason for lack of data is that there are very few scientific studies conducted concerning the traditional medicine domain area in Ethiopia. Plus, finding a structured archives that contain medicinal plants discovered in different parts of Ethiopia is very difficult. We crawled different government and non-governmental sources for a dataset to be used in the prototype model. Fortunately, we found a book called Flora of Ethiopia and Eritrea which contains an extensive list of plants species found in Ethiopia and Eritrea. This book is currently found in Addis Ababa University, Arat Killo branch. The stock of all published volumes of this book is owned by the university. The book contains many information about biodiversity, vegetation types, location, and use.

For the purpose of this research, we used datasets pertaining to plant family, plant species, and plant part used as traditional medicine. Annotations that describe details of usage of these plant specimens to cure a certain types of illnesses also included in the KBS. Since the book itself lack information about attributes and characteristics these plants, we are encouraged to use another secondary source. This source is a web portal called **Useful Tropical Plants**, and its URL is http://tropical.theferns.info/. It contains huge list of plant species and their attributes located all around the world. Snap shot figure of the data sample extracted from the book is depicted below.



68

Species	Family	Use	Part	Supplementary notes	Vol
Indigofera hirsuta	Fabaceae	craft manure medicinal		Elsewhere with a number of uses including anthelmintic, source of blue-black dye and as a green manure.	3
Indigofera linifolia	Fabaceae	food	seeds	Elsewhere the seeds has been ground into flour and used as famine food.	3
Indigofera oblongifolia	Fabaceae	medicinal feed craft	roots stems	Elsewhere the roots are reported to be toxic and used as a purgative. The stems are used for making toothsticks and are also browsed.	3
Indigofera schimperi	Fabaceae	feed		Elsewhere reported to be salt tolerant and a source of browse.	3
Indigofera spicata	Fabaceae	landscape feed		A valuable fodder plant. If eaten in large amounts, however, it is reported to cause abortion in cattle and sheep. Planting has been recommended for the control of erosion.	3
Indigofera suaveolens	Fabaceae	cosmetics		The local name indicates that this plant is used as a source of perfume.	[illegible]

Figure 9: Plant species list and their attributes

Here are a brief description about the dataset fields and their details as it is depicted above:

- **Plant Species:** Plants are the only organisms that can convert light energy from the sun into food. And plants produce ALL of the food that animals, including people, eat. Plantae [46]. Plant species are the actual plant specimens instances included in the ontology. They are represented with their scientific names. Hierarchically, plant species are below Plant family.
- **Plant Family:** In biological classification, family is one of the eight major taxonomic ranks. A family may be divide into subfamilies, which are intermediate ranks above the

rank Genus. In vernacular usage, a family may be named after one of its common members; for example, walnuts and hickory trees belong to the family Juglandaceae, commonly known as walnut family. Plant Families are higher above plant species in the hierarchicy of biological classification taxonomy [47].

- **Plant Parts used:** Plants parts are identifications of what part of the Plant Species is used to treat a certain sickness. Leaf, root, bark, fruits, resin, wood, flowers, bulbs and seeds are considered to be parts of a plant species. In some instances, every plant part of a certain plant species are used for medicinal purposes.
- **Usage:** Annotates or describes how these plants are used and their preparation in order to treat sicknesses. For instance, some plant parts used in conjunction with other plants to increase their potency.

The data that are used to annotate the ailment subclasses and instances are came from different trusted secondary web sources. Below are a list of some URL's used to capture annotation data used in the ontology to describe members of the Ailment super-class.

- http://www.webmd.com/cancer
- http://www.epilepsy.com/learn/epilepsy-101/what-epilepsy
- https://medlineplus.gov/neurologicdiseases.html
- http://www.lifeextension.com/
- https://medlineplus.gov/sexuallytransmitteddiseases.html
- http://www.healthline.com/
- http://www.medicinenet.com/script/main/art.asp?articlekey=2285
- http://www.everydayhealth.com/pms/long-periods.aspx
- http://www.nhs.uk/conditions/Multiple-sclerosis/Pages/Introduction.aspx

Primary data was also collected when conducting the questioner session for the purpose of gathering feedback of the proposed prototype. There are eight question provided for five carefully selected medical students in Adama General Hospital and they've produced a qualitative evaluation scores. Details about the questions and the overall result are on the experimentation section of this chapter.

4.3 Data quality assurance

Some data gathered from the book pertaining to the annotations of the medicinal plants usage and preparation lacks details and authenticity. This is because the usage section is comprised of many anecdotal information about the preparation and application of these medicinal plants. It is common to witness that some medicinal plants used differently and for different purposes in different areas. Because of this variation, it’s difficult to annotate their usage and create a classification based on preparations and usages in different areas of the country. Less studies done on this specific domain is one of the main factors for the problem stated above.

4.4 Preparation of the data

Selective gathering, sorting and filtering is essential when you deal with several fragmented sets of data. We have to carefully map the model of the ontology with the dataset gathered in order to avoid inconsistencies in the ontology. Web sources mentioned above used for annotation information selected based on their accuracy and popularity. Only segments are taken from these web sources to annotate the classes and instances of the ontology.

4.5 Experimentations and prototype evaluation

4.5.1 Experimentation I: Querying Plant Family Instances

Case 1: This segment of the experiment shows querying using SPARQL query scripts. We used both SPARQL and DL Query found within the Protégé platform. The justification why both platforms chosen instead of just one to query the ontology is presented on the previous chapter. The data that we want to extract is about a plant family called **Asteraceae** and how many plant species instances are contained within it. The SPARQL query script to query this specific data is as follows:

```
select ?Plant_Species_1 where
{ ?Plant_Species_1 <http://www.semanticweb.org/john/ontologies/2017/2/untitled-ontology-24#PartOf>
<http://www.semanticweb.org/john/ontologies/2017/2/untitled-ontology-24#
Asteraceae> . }
limit 32
```

Table 2: SPARQL query string pertaining to the Asteraceae plant family

In the above query, we created a variable called **Plant_Species_1** and we want to place a query result which contains members of a plant species which are related to a plant family called **Asteraceae** in the variable**.** We use the object property called **Partof** as a relationship creator. As the query results shown below, we got four plant species associated with the **Asteraceae** plant family.

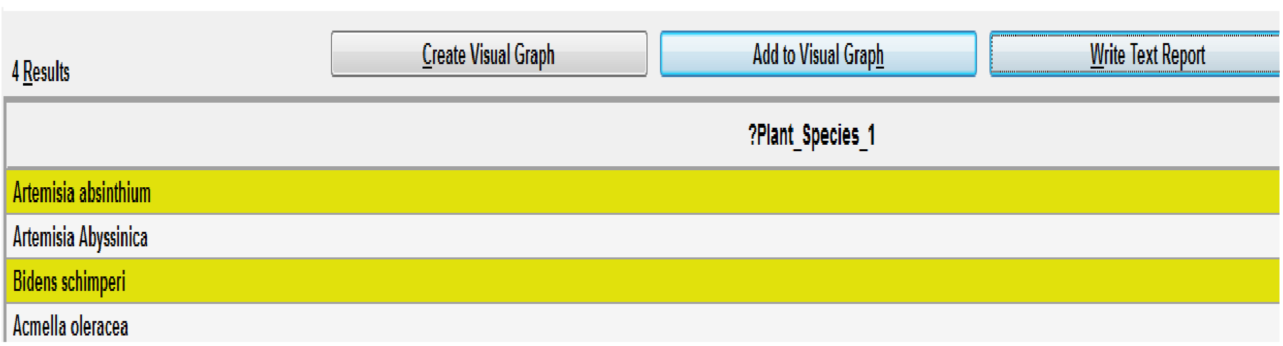


Figure 10: SPARQL query results of the Asteraceae plant family

As we can see, we retrieved four instances that are related to the plant family used in the query string above. Different parts of these plant species used for medicinal purposes. Yet, the SPARQL query results didn't show which parts of the plant of these plant species are used to treat illnesses in the ontology. But here is where the visualization of the generated SPARQL query comes handy.

The visualization below gives even more expressive results of the query. It adds extra information about what part of the plant species is associated for medicine and their relation with the plant family.

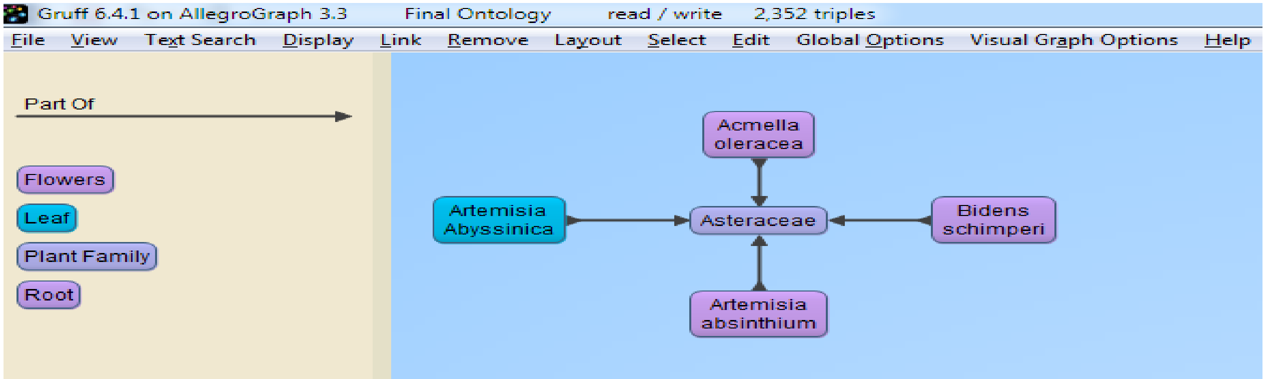


Figure 11: The visualized query results for the Asteraceae plant family

As we can see from the above figure, we are treated with extra bits of information concerning which part of the plant is associated with the plant species specified in the query results. We can know which of which by comparing the corresponding colors on the left and the right panes of the views on the **Gruff** platform.

DL Query is also used to query the same information pertaining to the plant family **Asteraceae.** The query is done using a simple one line string that contains the name of an instance or class, object property and the **value** key word. For this particular query, which pertains to the plant family **Asteraceae** and what plant species it contains, the query string look like the following:

PartOf ***value*** *Asteraceae*

The figure bellow shows the DL query string and the retrieved data after executing the query with in Protégé:

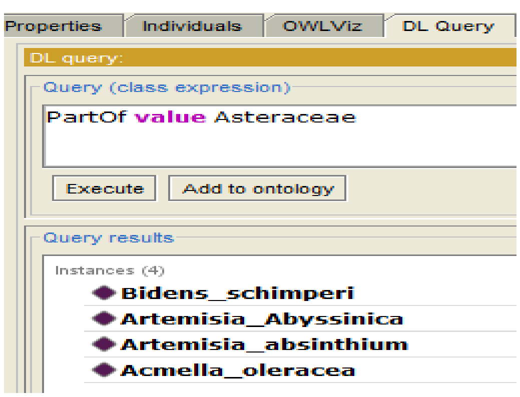


Figure 12: DL query results of the Asteraceae plant family

Case 2: In this specific case, a query is formulated to extract data concerning a plant family called **Lamiaceae**. As it is done in the previous case, the query is presented in both SPARQL and DL query format. The SPARQL query string used to extract the instances of the **Lamiaceae** plant family is as follows:

```
select ?PlantFamilyInfo where
{ ?PlantFamilyInfo <http://www.semanticweb.org/john/ontologies/2017/2/untitled-ontology-24#PartOf>
<http://www.semanticweb.org/john/ontologies/2017/2/untitled-ontology-24#Lamiaceae> . }
limit 32
```

Table 3: SPRQL query string for extracting plant species found in the Lamiaceae family

After executing the above query, there were three results produced. This means the **Lamiaceae** plant family contains three plant species with in it. The query results is shown below:

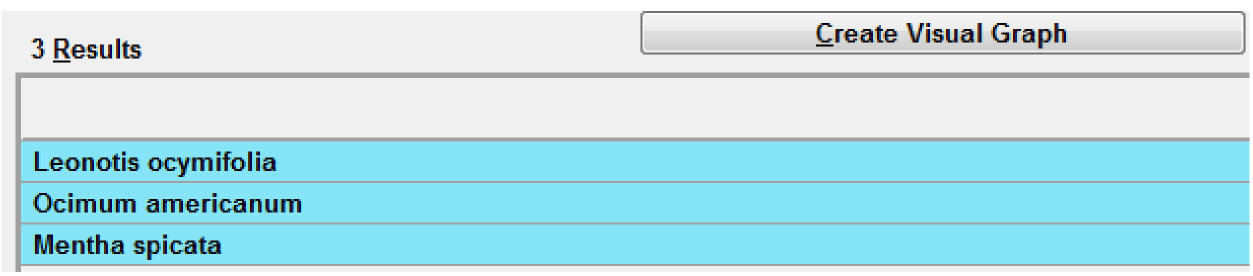


Figure 13: SPARQL query results of the Lamiaceae plant family

The graph visualization also depicts all the data concerning the plant family that is being queried and its instances. It also shows the object properties used to link the plant family to the plant species. As it shown from the visualization graph below, the **PartOf** object property link, depicted using the orange arrow, is used to link the Plant family called **Lamiaceae** with plant species instances with in **Lamiaceae**.

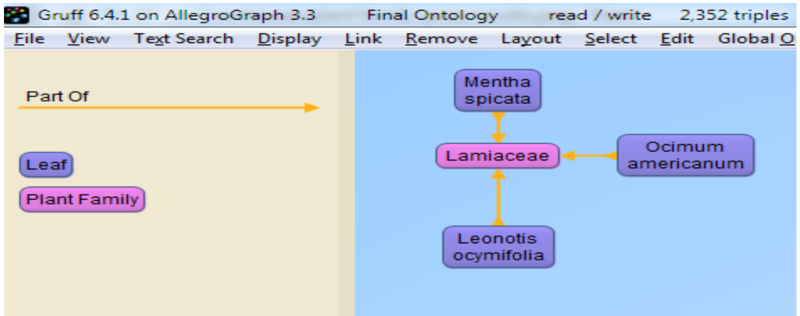


Figure 14: The visualized query results for the Lamiaceae plant family

The DL query is formulated using a single line of query string. This query involves an object property link creator **PartOf**, a key word called **value** and the name of the plant species. The query string and the result instances of the query is shown in the figure below:

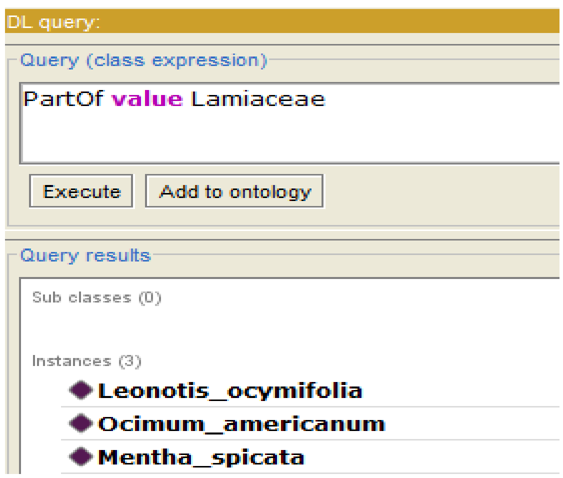


Figure 15: DL query results of the Lamiaceae plant family

4.5.2 Experimentation II: Querying Disease Instances

Case 1: This case deals with extracting a data which is related with a disease called as **Malaria** and which plant species are responsible to have a healing property in relation to this disease. The SPARQL query below used to extract which plant species are used to treat **Malaria**.

```
select? Plant_Species where

{ ?Plant_Species <http://www.semanticweb.org/john/ontologies/2017/2/untitled-ontology-24#UsedFor>
<http://www.semanticweb.org/john/ontologies/2017/2/untitled-ontology-24#Malaria> . }

limit 32
```

Table 4: SPRQL query string for extracting plant specimens used to treat Malaria

On the above SPARQL query string, we have the Plant_Species variable, **UsedFor** as an object property and the instance of the Ailment class **Malaria**. If we put it in one sentence, the query would look something like this: ***Plant_Species*** *UsedFor* ***Malaria***. The results of the above query is shown below.

7 Results | Create Visual Graph | Add to Visual Graph

?Plant_Species
Artemisia absinthium
Albizia coriaria
Buckollia volubilis
Senna petersiana
Tamarindus indica
Lagenaria siceraria
Eucalyptus robusta

Figure 16: SPARQL query results containing instances of Plant species used to treat Malaria

The visualization for this particular query further enhances the information we get from the query result. Below, we can see the visualization graph describes the relations that the plant species have with the **Malaria** instance.

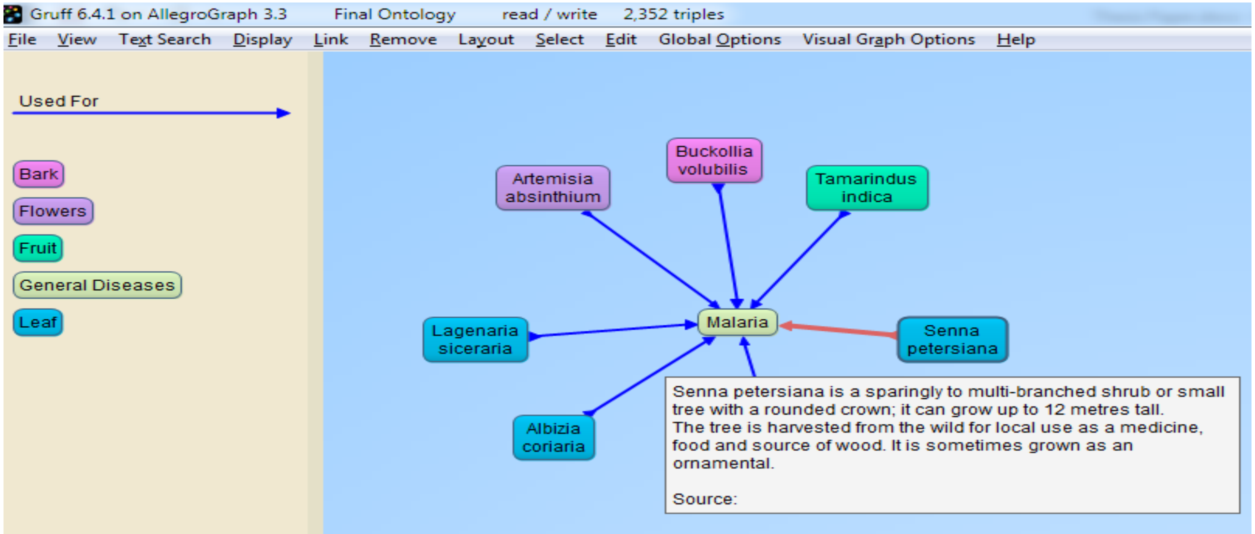


Figure 17: Visualized representation of the SPARQL query results generated for Malaria

Again visualization provides semantically enriched information which the plain SPARQL query can't. The **AllegroGraph** platform even allows us to hover on any on the nodes of instances and see the annotated information of the plant species to gain further information about them on the fly.

The DL query, even though it's not as powerful as SPARQL, it has its own useful. Plus, for most domain experts, who are not tech savvy, it's easier than SPARQL. Below is the DL query equivalent of the above SPARQL query.

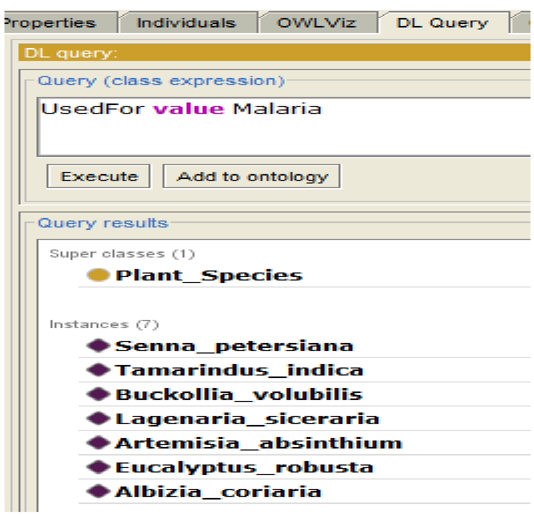


Figure 18: DL query results of Malaria

As the above figure for experimentation II shows, there are seven instance or plant species that are used to treat the **Malaria** in the ontology KBS.

Case 2: **Diabetes** is another disease that is represented in the knowledge base as an instance of the Ailment class. We can again use SPARQL and DL query to extract which plant species instances are used to treat diabetes. The SPARQL query is shown in the following table:

```
select ?Disease_Info2 where

{ ?Disease_Info2 <http://www.semanticweb.org/john/ontologies/2017/2/untitled-ontology-24#UsedFor>
<http://www.semanticweb.org/john/ontologies/2017/2/untitled-ontology-24#Diabetes> . }

limit 32
```

Table 5: SPRQL query string for extracting plant specimens used to treat Diabetes

On the above SPARQL query, we have the Plant_Species variable, **UsedFor** as an **object property** and the instance **Diabetes**. If we put in one sentence, the query would look something like this: ***Plant_Species*** *UsedFor* ***Diabetes***. The results of the above query is shown the figure below.

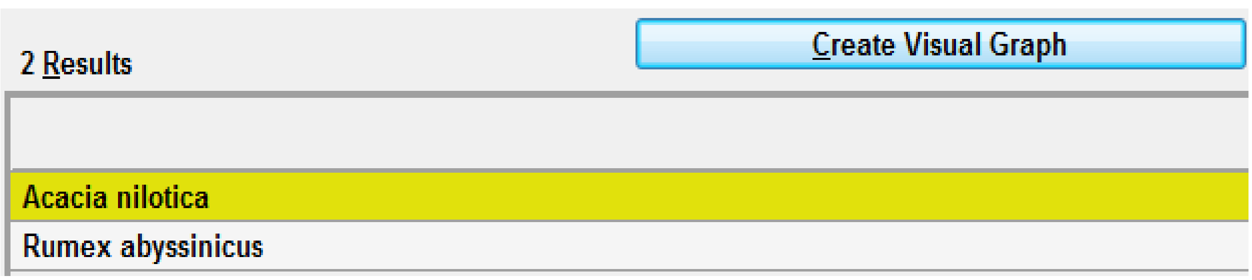


Figure 19: SPARQL query results for the Diabetes disease instance

The DL query equivalent of the above SPARQL query is even easier to formulate in order to extract data which is pertaining to the Diabetes instance and the plant species associated with it. The figure below shows the single line DL query and the result instances.

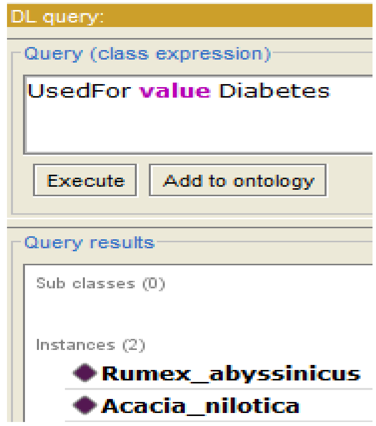


Figure 20: DL query results for the Diabetes disease instance

Again visualizing a query yields an even better results in providing extra bits of details concerning the query. In case of this specific visualization below, we can see that the **Diabetes** disease instance is part of the **Metabolic_Problem** class as it can be seen from matching colors. We can also see from the matching colors that the Bark of the **Acacia_nioltica** and the Root of the **Rumex_abysinicus** plant species are used to treat Diabetes. The visualization figure is shown below:

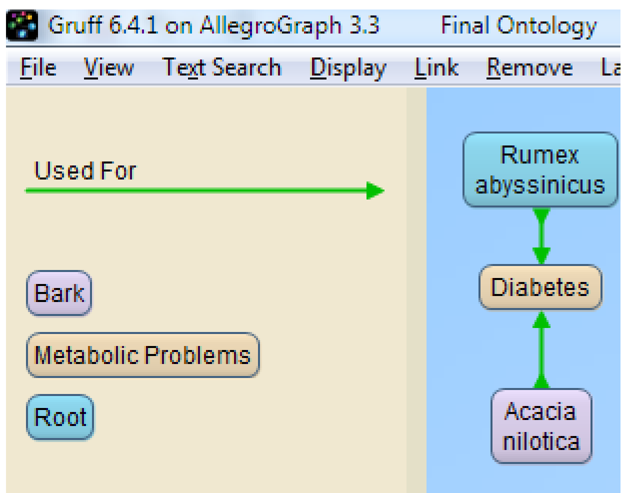


Figure 21: Visualized SPRQL query results for the Diabetes disease instance

DL Query Instance Data Attributes

These plant species are represented with attributes just like every instances in the ontology. These attributes for the plants include:

- **Height**: How high the plants grows until it reaches maturation.
- **Range**: Which areas in the world do these plants are reported to be widely available or natively discovered.
- **Habitat**: Which ecological area that a specific plant species inhabits. For instance, a plant species scientifically called Punica **grantatum** or commonly known as Pomegranate grows in dry limestone soil at elevation up to 2700 meters in the Himalayas.

The figure below shows how a specific plant is represented using data attributes and object property values with in Protégé.

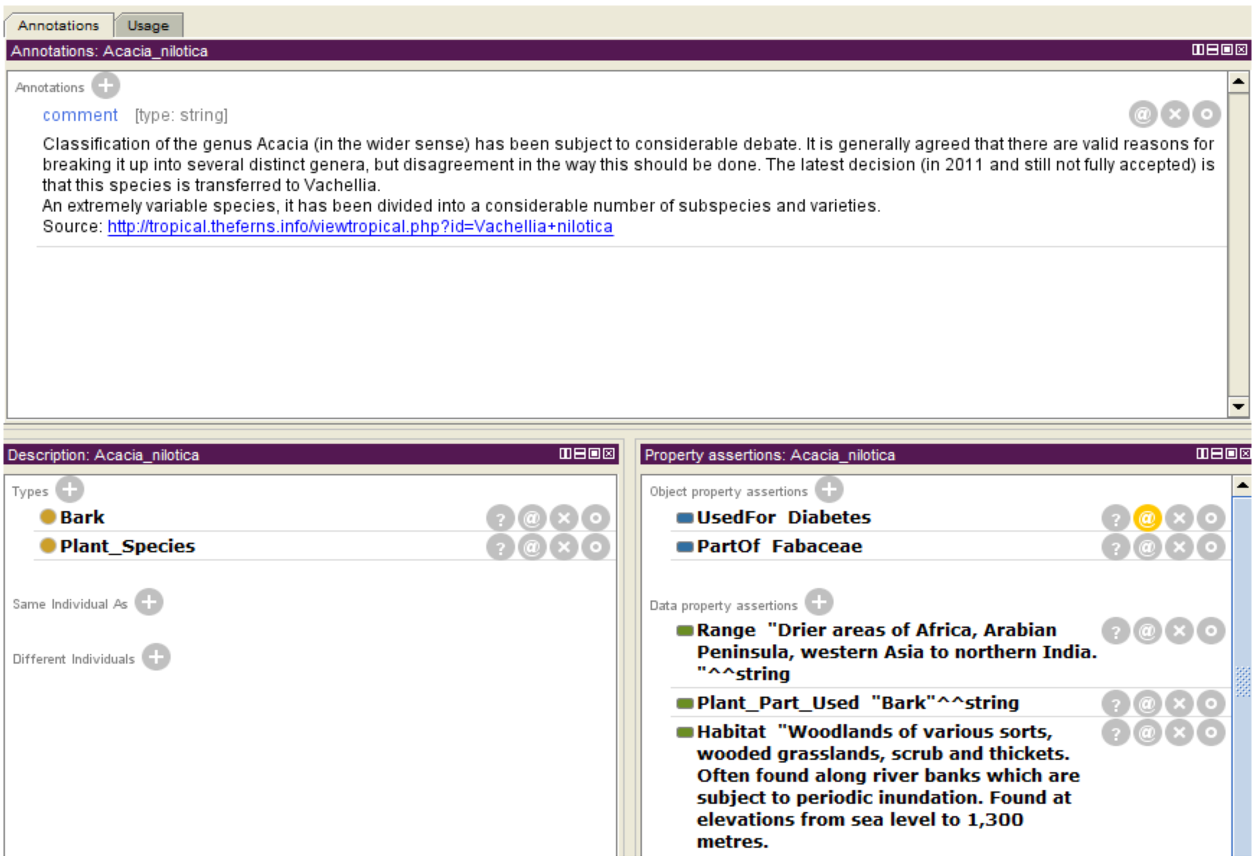


Figure 22: Data property, object property, description and annotation view for the plant species Acacia_nilotica

4.5.3. Prototype ontology evaluation

Prototype evaluation is one of the essential phases in getting a feedback on how the prototype is performed and how much of the domain area is represented in the knowledge base. Eight questions were prepared in order to qualitatively evaluate the prototype. Five domain experts are selected from medical students who are studying in Adama General Hospital Medical College. These students are on the advanced stage of their school program and they are identified as knowledgeable in the domain area of traditional medicine. These questions covers areas such as usability, effectiveness of knowledge representation of the domain, querying ability and overall performance. The following table show what questions are presented to the students:

Questions number	**Questions**
Question 1	A person with no computer skills can be able to use this KBS?
Question 2	It was simple to use the prototype system?
Question 3	It was easy to find the information that I needed?
Question 4	The Classes, the Subclasses, instances, attributes and relationships aspects of the ontology are well defined?
Question 5	The visualization mechanisms sufficiently depicts an overall snapshot of the various component of the ontology?
Question 6	It's better than any knowledge management system I have encountered in the past that concerned with traditional herbal medicine?
Question 7	It lacks some details about the ontology in relation to medicinal plant specimens?
Question 8	I would recommend this ontology based KBS to domain experts who work on the traditional medicines and other related areas?

Table 6: Prototype evaluation feedback questions

The domain experts used five qualitative evaluation marks to provide a feedback based on each questions provided in the questioner. These marks are **Strongly Agree**, **Agree**, **Neutral, Disagree**, and **Strongly Disagree**. For instance, if the majority of questions given a strongly agree and agree marks, it shows that overall, the system proposed is accepted by the domain experts. The following table shows the feedback result generated by the domain experts for the questioner above.

Questions Number	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	-	-	2	3	-
2	-	4	1	-	-
3	4	-	1	-	-
4	4	1	-	-	-
5	3	1	1	-	-
6	3	1	1	-	-
7	-	2	-	3	-
8	2	3	-	-	-

Table 7: Evaluation results

The above table shows how many of the five respondents chose which evaluation marks for each questions. For instance, in case of question number 4, four respondent chose to strongly agree and one respondent chose neutral for that specific question after the assessment. The pie chart figure below depicts the overall acceptability in percentage of the proposed KBS produced after assessing the feedback data above.

The format of the evaluation question is taken from a thesis project called **A Knowledge Based Expert System for Medical Advice Provision.** This thesis project is conducted by Kulani Makhubele in the University of Cape Town 2012

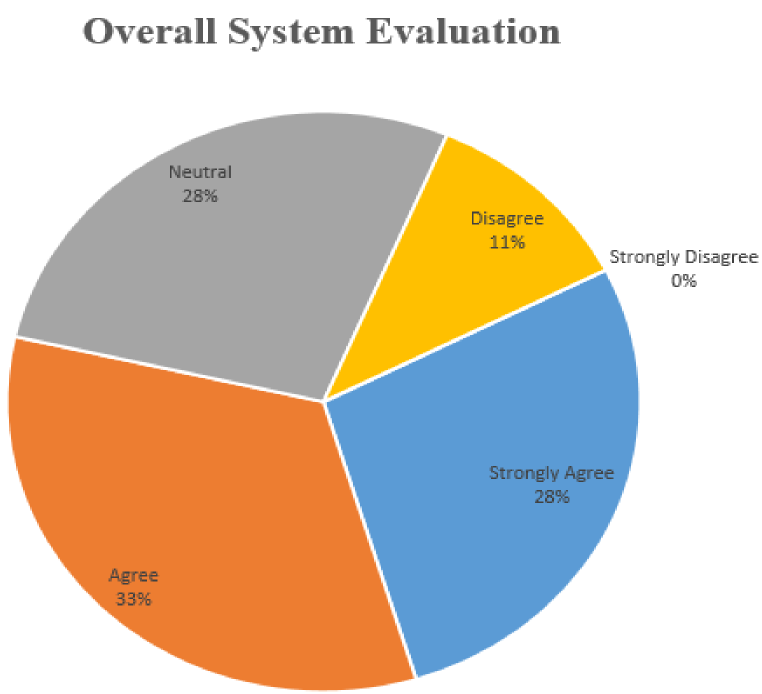


Figure 23: This figure shows the acceptability percentage of the prototype model

The above chart clearly illustrates that the domain experts, specifically the medical students participated in evaluating the prototype model found it to be overall satisfactory. As you can see from the pie chart above, strongly agree represents 28%, agree represents 33% and neutral represents 28 %, which produced a combined result of 89 %. This combined results mean that especially on the main components of the ontology such as overall domain representation, visualization of the instances as well as in accurately defining the classes in the ontology, the domain experts found the ontology to be very useful.

4.6 Summary:

Overall the main aspects of this chapter tells us that ontology models are a very useful mechanisms in representing a certain type's knowledge in a very expressive manner. Despite the high performance and data driven nature of traditional data management schemes such as RDMS, providing semantics, interoperability and schema flexibility are what makes ontologies unique compare to other platforms. The domain area chosen in this research, which is the Ethiopian herbal traditional medicine shown to benefit from the unique features provided by ontological models.

CHAPTER FIVE

DISCUSSION, CONTRIBUTION AND RECOMMANDATION

Brief summary of what are discussed so far in the previous chapters, limitations faced while conducting this research and recommending future works that can benefit from the fruits of this thesis included in this chapter.

5.1. Summary of the thesis

Ontologies, now a days are heavily integrated in the Sematic Web as well as Artificial Intelligence applications as a knowledge representation schemes. Schema flexibility allows ontologies to model a certain concept without restriction and easily introduce new concepts to the KBS. In this research, it is demonstrated that what the specific domain areas ontologies are useful and what are their unique features. Scientific research projects in higher educations and large scale government supported projects and many more benefits from an ontological driven applications.

This thesis provides an evidence on the richness and expressiveness of ontologies. The traditional medicine domain used in this thesis benefited from the unique query and visualization methods of accessing an ontology for producing a certain results. The results retrieved from evaluating the KBS prototype indicated that the overall accuracy in representing the domain area found to be good enough.

5.2. Contribution

This thesis has contributed in disseminating a fairly significant understanding on ontologies as a core building blocks of a knowledge base systems, Semantic Web and AI applications. The fact of the matter is that not much research projects has been conducted in Ethiopia that incorporates ontologies and the Sematic Web field of studies. To be specific, research works or journal articles that are focused on querying mechanisms for ontologies such as SPARQL are few and far between. Considering that querying schemes and visualizations of semantic data is a very significant portion of any ontology study, the contribution of this research in his area is substantial.

The domain area used for building this ontology based KBS also greatly benefited as it showed on the prototype evaluation results in the previous chapter. No known system in Ethiopia uses ontology based applications to capture and record data that require schema flexibility to accommodate new concepts and follows an open world assumption. The OWL framework used to accomplish both these feats showed a great promise and hopefully provided encouragement for future research projects in this subject. The researcher also hopes that future research works that are based on ontologies and Semantic Web use this thesis as a launch pad to strengthen their work.

5.3. Recommendations for Future Research

The main motivating factor to do this research, as it is stated above on the previous chapter is to arouse an interest in this subject in Ethiopia. The hope is that Ethiopian researchers, Sematic Web application developers, and the AI community would exploit the benefits of ontologies and Semantic Web applications as a whole. Some of the future works that are recommended by the researcher are as follows:

- Further develop the ontology using open-source semantic application development platforms such as Jena. Expand the ontology model by creating new classes and instances that can enrich the ontology in terms of design. This would of course require a further study on the traditional medicine domain area and have access to a substantial amount of peer-reviewed works on the domain.
- Merge other domain related ontologies with this one in order to have even more richness in vocabulary, relationships and annotations. And also test the integrity of this ontology after merging it with other related ontologies.
- Converting it to an RDF triple store and share it to the triple store web repositories.
- It may not be directly linked to this thesis but the DBPedia project is a very powerful tools for data miners and researchers these days. What DBPedia essentially do is use the semi-structure data available in Wikipedia and make it available to the public for querying using SPARQL. It means DBPedia allows us to ask complex queries against Wikipedia because it uses ontologies in its core. The researcher recommends a possible future research work that studies and investigates the capability of DBPedia in aiding researches and system developers in Ethiopia.

REFERENCE

[1] G. Tom, "Ontology," web.dfc.unibo, 2008. [Online]. Available: http://web.dfc.unibo.it/buzzetti/IUcorso2007-08/mdidattici/ontology-definition-2007.htm. [Accessed 2017].

[2] C. Martinez-Cruz, I. J. Blanco and M. A. Vila, "Ontologies versus relational databases: are they so different? A comparison," 2011.

[3] M. Uschold, "slideshare," 2011. [Online]. Available: https://www.slideshare.net/UscholdM/ontologies-and-db-schema-whats-the-difference. [Accessed 2017].

[4] A. M. Florea, "Knowledge Representation and Reasoning," University "Poli tehnica" of Bucharest Department of Computer Science, 2009.

[5] R. García, "A Semantic Web approach," 2005.

[6] J. A. Bullinaria, "IAI : Knowledge Representation," 2005.

[7] R. Davis, H. Shrobe and a. P. Szolovits, "What is a Knowledge Representation?," 1993.

[8] Cercone and G. M. a. Nick, "Approches to Knowledge Represenation," 1983.

[9] G. M. a. N. Cercone, "Approches to Knowledge Representation," 1983.

[10] Mackworth and D. P. a. Alan, "artint.info," 2010. [Online]. Available: http://artint.info/html/ArtInt_8.html.

[11] R. Davis, H. Shrobe and a. P. Szolovits, "What is a Knowledge Representation?," 1993.

[12] s-hammami, "Knowledge acquisition, representation and reasoning".

[13] E. T. a. J. E., Decision Support Systems and Intelligent Systems, Upper Saddle River, NJ: Prentice Hall, 2001.

[14] E. T. a. J. E. Aronson, "Decision Support Systems and Intelligent Systems," Upper Saddle River, NJ, 2001.

[15] E. T. a. J. E. Aronson, "slideshare," 29 August 2014. [Online]. Available: http://www.slideshare.net/theintegralworm/knowledge-acquisition.

[16] T. Luis, "Springer Link," 2013. [Online]. Available: http://link.springer.com/referenceworkentry/10.1007%2F978-1-4419-9863-7_166. [Accessed 2017].

[17] F. Hayes-Roth, D. Waterman and D. Lenat, Building Expert Systems, 1983.

[18] Lewis and N. L. G. a. F. D., "A Rule-Based Inference Engine Which is Optimal and VLSI Implementable," *Lexington.*

[19] M. L. Owoc and M. O. a. T. Gladysz, "Validation and Verification of Knowledge Based Systems," 1999.

[20] A. Preece, "Evaluating Verification and Validation Methods in Knowledge Engineering," 2001.

[21] A. Preece, "Evaluating Verification and Validation Methods in Knowledge Engineering," 2001.

[22] B. N., T. C., G. N., S. I. and a. C. S, "XML and Semantic Web Worlds W3C Standards Timeline," 2012.

[23] M. K. a. S. R. Pascal Hitzler, "Foundations of Semantic Web," Taylor and Francis Group, LLC, 2010.

[24] M. K. a. S. R. Pascal Hitzler, "Foundations of Semantic Web," Taylor and Francis Group, LLC, 2010.

[25] M. J. a. R. Meersman, "Ontology Engineering – The DOGMA Approach," 2008.

[26] M. Hegde, "An Entity-centric Approach for Overcoming Knowledge Graph Sparsity," 2015.

[27] V. Beal, "webopedia/Ontology_Web_Language," 2016. [Online]. Available: http://www.webopedia.com/TERM/O/Ontology_Web_Language.html.

[28] https://www.w3.org/2001/sw/#owl, "Web Ontology Language," 2016. [Online]. Available: https://www.w3.org/OWL/.

[29] A. H. a. S. Decker, "Optimized Index Structures for Querying RDF from the Web," 2005.

[30] G. a. S. W. H. Vavidu, "Ontology Mapping of Indian Medicinal Plants With Standardize Medical," 2012.

[31] G. a. S. W. H. Vavidu, "Ontology Mapping of Indian Medicinal Plants With Standardize Medical," 2012.

[32] S. o. u. J. A. H. Mohd Suffian Sulaiman, "Multi-Modality Ontologyfor Herbal Medicinal Plant Semantic Based," 2014.

[33] S. o. u. J. a. A. A. H. Mohd Suffian Sulaiman, "Multi-Modality Ontologyfor Herbal Medicinal Plant Semantic Based Image Retrival," 2014.

[34] S. o. N. J. A. H. Mohd Suffian Sulaiman, "Multi-Modality Ontologyfor Herbal Medicinal Plant Semantic Based Image Retrival," 2014.

[35] M. Rouse, "http://searchhealthit.techtarget.com," 2010. [Online]. Available: http://searchhealthit.techtarget.com/definition/SNOMED-CT. [Accessed 2017].

[36] O. W. Group, "w3.org," 2012. [Online]. Available: https://www.w3.org/OWL/. [Accessed 2017].

[37] S. Peroni, "essepuntato.it," [Online]. Available: http://www.essepuntato.it/graffoo/. [Accessed 2017].

[38] M. M. S. a. F. Rebelo, " Advances in Usability Evaluation," 2012.

[39] A. Lalingkar, C. Ramnathan and S. Ramani, "MONTO: A Machine-Readable Ontology for Teaching Word Problems in Mathematics," 2015.

[40] P. Lord, "Ontogenesis," 2010. [Online]. Available: http://ontogenesis.knowledgeblog.org/514. [Accessed 2017].

[41] P. Lord, "Ontogenesis," 2010. [Online]. Available: http://ontogenesis.knowledgeblog.org/514. [Accessed 2017].

[42] A. Katifori, C. Vassilakis, I. Daradimos, G. Lepouras, Y. Ioannidis, A. Dix, A. Poggi and T. Catarci, "http://alandix.com," 2008. [Online]. Available: http://alandix.com/academic/papers/PIM2008-personal-ontology/. [Accessed 2017].

[43] S. University, "http://protege.stanford.edu," [Online]. Available: http://protege.stanford.edu/community.php. [Accessed 2017].

[44] E. Alatrish, "Comparison Some of Ontology Editors," vol. 8, 2013.

[45] R. S. Khule, "The Study of Basics for a Query Formulation Language – MashQL," 2015.

[46] mbgnet.net, "mbgnet.net," 2009. [Online]. Available: http://www.mbgnet.net/bioplants/earth.html. [Accessed 2017].

[47] yeni_iglesias, "quizlet.com," [Online]. Available: https://quizlet.com/161033546/topic-9-flash-cards/. [Accessed 2017].

[48] "semanticweb.org," 2012. [Online]. Available: http://semanticweb.org/wiki/Ontology.html. [Accessed 2017].

[49] K. S. a. D. R. Umarani, "A Survey of Semantic based Solutions to Web Mining," 2012.

[50] M. Obitko, "obitko.com," 2007. [Online]. Available: https://www.obitko.com/tutorials/ontologies-semantic-web/web-ontology-language-owl.html. [Accessed 2017].

[51] V. Beal, "webopedia.com," [Online]. Available: http://www.webopedia.com/TERM/O/Ontology_Web_Language.html. [Accessed 2017].

[52] R. W. Group, "w3.org," 2014. [Online]. Available: https://www.w3.org/RDF/. [Accessed 2017].

[53] "ontotext.com," 2017. [Online]. Available: https://ontotext.com/knowledgehub/fundamentals/what-is-sparql/. [Accessed 2017].

[55] R. G. González, "A Semantic Web approach to," 2005.

[56] J. A. Bullinaria, "IAI : Knowledge Representation," 2005.

[57] H. S. a. P. S. Randall Davis, "What is a Knowledge Representation?," 1993.

[58] G. M. a. N. Cercone, "Approches to Knowledge Represenation," 1983.

[59] David Poole and Alan Mackworth, [Online]. Available: http://artint.info/html/ArtInt_8.html.

[60] H. S. a. P. S. R. Davis, "What is a Knowledge Representation?," 1993.

[61] "Knowledge_engineering," [Online]. Available: en.wikipedia.org/wiki/Knowledge_engineering.

[62] E. T. a. J. E. Aronson, Decision Support Systems and Intelligent Systems, Upper Saddle River, NJ: Prentice Hall, 2001.

[63] E. T. a. J. E. Aronson, "Decision Support Systems and Intelligent Systems," Upper Saddle River, NJ, 2001.

[64] L. Tari, "Link.Springer," Springer New York, 2013. [Online]. Available: http://link.springer.com/referenceworkentry/10.1007%2F978-1-4419-9863-7_166.

[66] N. L. G. a. F. D. Lewis, "A Rule-Based Inference Engine Which is Optimal and VLSI Implementable".

[67] "Validation and Verification of Knowledge Based Systems," in *Validation and Verification of Knowledge Based Systems*, 1999.

[68] A. Preece, "Evaluating Verification and Validation Methods in Knowledge Engineering".

[69] A. Preece, "Evaluating Verification and Validation Methods in Knowledge Engineering".

[70] M. K. a. S. R. Pascal Hitzler, "Foundations of Semantic Web," Taylor and Francis Group, LLC, 2010.

[72] "webopedia/Ontology_Web_Language," [Online]. Available: http://www.webopedia.com/TERM/O/Ontology_Web_Language.html. [Accessed 2016].

[73] G. a. S. W. H. Vavidu, "Ontology Mapping of Indian Medicinal Plants With Standardize Medical Terms," 2012.

[74] "revolvy.com," 2017. [Online]. Available: https://www.revolvy.com/topic/Plant&item_type=topic. [Accessed 2017].

APPENDICES

Appendix-A

Ontologies:

Ontologies are considered one of the pillars of the Semantic Web, although they do not have a universally accepted definition. A (Semantic Web) **vocabulary** can be considered as a special form of (usually light-weight) ontology, or sometimes also merely as a collection of URIs with an (usually informally) described meaning.

Ontologies on semanticweb.org.edu are usually assumed to be accompanied by some document in a formal ontology language, though some ontologies do not use standardized formats for that purpose. A list of documents that are considered ontologies in this wiki is given below [48].

In computer science and artificial intelligence, **ontology languages** are formal languages used to construct ontologies. They allow the encoding of knowledge about specific domains and often include reasoning rules that support the processing of that knowledge. Ontology languages are usually declarative languages, are almost always generalizations of frame languages, and are commonly based on either first-order logic or on description logic.

Markup ontology languages

These languages use a markup scheme to encode knowledge, most commonly with XML.

- DAML+OIL
- Ontology Inference Layer (OIL)
- Web Ontology Language (OWL)
- Resource Description Framework (RDF)
- RDF Schema (RDFS)
- SHOE [49]

Appendix-B
Web Ontology Language (OWL):

The Web Ontology Language OWL extends RDF and RDFS. Its primary aim is to bring the expressive and reasoning power of description logic to the semantic web [50].

OWL is a set of markup language which are designed for use by applications that need to process the content of information instead of just presenting information to humans.

The OWL ontology describe the hierarchical organization of ideas in a domain, in a way that can be parsed and understood by software. OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web.

OWL is part of the W3C recommendations related to the Semantic Web [51]. The figure below which is taken from the slideshare.net website shows where the OWL language lies in the Semantic Web stack or architecture:

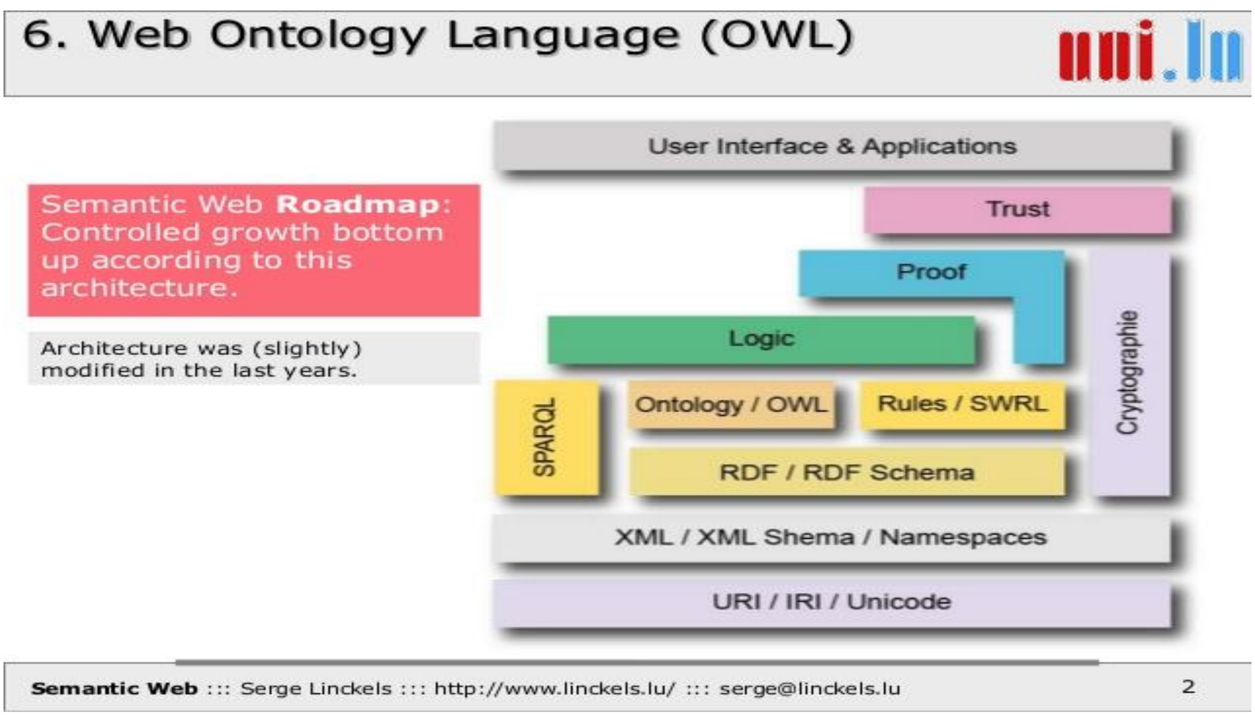


Figure 24: Semantic Web Stack

Appendix-C

Resource Description Framework (RDF)

RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed.

RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a “triple”). Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications.

This linking structure forms a directed, labeled graph, where the edges represent the named link between two resources, represented by the graph nodes. This graph view is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations [52].

Appendix-D

SPARQL Query

SPARQL, pronounced ‘sparkle’, is the standard query language and protocol for Linked Open Data on the web or in a semantic graph database (also called RDF triplestore).

SPARQL, short for “SPARQL Protocol and RDF Query Language”, enables users to query information from databases or any data source that can be mapped to RDF.

The SPARQL standard is designed and endorsed by the W3C and helps users and developers focus on what they would like to know instead of how a database is organized.

Just like SQL allows the user to retrieve and modify data in a relational database, SPARQL provides the same functionality for NoSQL graph databases like GraphDB.

In addition, a SPARQL query can also be executed on any database that can be viewed as RDF via middleware. For example, a relational database can be queried with SPARQL by using a Relational Database to RDF (RDB2RDF) mapping software.

This is what makes SPARQL such a powerful language for computation, filtering, aggregation and sub-query functionality.

In contrast to SQL, SPARQL queries are not constrained to work within one database: Federated queries can access multiple data stores (endpoints). Consequently, SPARQL overcomes the constraints posed by local search.

The power of SPARQL together with the flexibility of RDF can lead to lower development costs as merging results from multiple data sources is easier.

These design choices – enabling queries over distributed sources on non-uniform data, are not accidental. SPARQL is designed to enable Linked Data for the Semantic Web. Its goal is to assist people to enrich their data by linking it to other global semantic resources, thus sharing, merging, and reusing data in a more meaningful way.

The biggest strength of SPARQL is navigating relations in RDF graph data through graph pattern matching, where simple patterns can be combined into more complex ones that explore more elaborate relations in the data.

Such relations can be explored by using basic patterns, pattern joins, unions, by adding optional patterns that may extend the information about the found solutions, etc. Furthermore, property paths allow sequential composition (sequencing), parallel composition (alternatives), iterations (Kleene star), inversion, etc.

A basic graph pattern consists of a triple in which each element (the subject, predicate and object) can be a variable (wildcard).

For example, the pattern *'John' (a subject)->'has son' (a predicate)->X (a wildcard object)* will have as a solution each triple in the RDF graph that matches the subject, matches the predicate, and has any object.

So if John has two sons – Bob and Michael, the triples '*John'->'has son'->'Bob'* and *'John'->'has son'->'Michael'* will be the solutions to the SPARQL query [53].