Sudan University of Science and Technology

Collage of Graduate Studies

PHD in Computer Science

Building a Model for Holy Quran Tajweed Knowledge using Ontology

بناء أنموذج معرفي لتجويد القران الكريم بإستخدام الأنطولوجيا

A thesis submitted in fulfillment of the requirements of the award of the degree of Doctor of Philosophy in Computer Science

Prepared by:

Eiman Alsiddig Eltayeb Ibrahim

Supervised by:

Associate Prof. Eric Atwell

August 2017

بسم الله الرحمن الرحيم

يَا أَيُّهَا الْمُزَّمِّلُ{١} قُمِ اللَّيْلَ إِلَّا قَلِيلاً{٢} نِصْفَهُ أَوِ انقُصْ مِنْهُ
قَلِيلاً{٣} أَوْ زِدْ عَلَيْهِ وَرَتِّلِ الْقُرْآنَ تَرْتِيلاً{٤} إِنَّا سَنُلْقِي عَلَيْكَ قَوْلاً
ثَقِيلاً{٥}

سورة المزمل (١ -٥).

DEDICATIONS

To the spirit of my mother,,

My idol, my dear father,,

My Soul mate, my husband,,

The source of my happiness, my Lovely Sons,,

Honorary source, My Brothers and Sisters,,

Inexhaustible Knowledge source, my Venerable supervisor,,

Those who did not hesitate to give a helping hand, my Colleagues,,

All science students,,

I dedicate my modest research. I ask God Almighty to benefit Islam and Muslims.

ACKNOWLEDGEMENTS

Thanks God first and foremost to complete this research. Then I would like to express my gratitude to my Supervisor Associate Prof. Eric Atwell, who gave me a golden advices and eliminated all difficulties and odds even completed this search.

I am also grateful to my family, they always ask Almighty Allah to guide me. My special thanks to my dear husband, Assistant prof. Mohammed Awad Mohammed, without his help I wouldn't be able to finish this work. I am a grateful to each one who gave a helping hand to reach goals of this research.

ABSTRACT

The Holy Quran is the word of Allah Almighty revealed by His Messenger Muhammad (peace be upon him). It has a set of rules governing the way in which the words of the Quran should be pronounced during its recitation (Tajweed).

The objective of this research is to design an ontology for Holy Quran Tajweed (Articulations Points of the Letters, Un Vowel Noon and Tanween) to support the learning of this part of Tajweed and to facilitate the sharing of knowledge with the other Holy Quran applications. In order to achieve this goal, the researcher adopted the applied approach using protégé framework with OWL to build the Ontology.

This research listed the steps of ontology development process. Enumerated the terms that can possibly found in the domain, and then addressed the complex issues of defining class hierarchies, organized them in a hierarchy depending on which class subsumes another. And then defined properties for each class and the relationships linked these classes. Also restrictions were put on some properties according to what the rules of Tajweed stipulate. Last step in designing the ontology was added instances to each class to be able to make queries, and ensure that all the classes are consistent with each other by using the Reasoner. Finally, some Queries were applied to the ontology using both DL Query and SPARQL Query.

The designed ontology can retrieves information about Articulation Points of the Letters, Un Vowel Noon and Tanween. Furthermore, it can be shared and reused in applications related to the Holy Quran.

Key words: Quran, Tajweed, Articulations Points of the Letters, Ontology, Un Vowel Noon, Tanween.

المستخـــلص

القرآن الكريم هو كلام الله سبحانه وتعالى المنزل على رسوله محمد (صل الله عليه وسلم). وله مجموعة من القواعد التي تحكم الطريقة التي يجب أن تنطق بها كلمات القرآن خلال تلاوته (التجويد).

والهدف من هذا البحث هو تصميم أنطولوجيا لتجويد القرآن الكريم (مخارج الحروف وأحكام النون الساكنة والتنوين) لدعم تعلم هذا الجزء من التجويد وتسهيل تبادل المعرفة مع تطبيقات القرآن الكريم الأخرى. ولتحقيق هذا الهدف، اعتمد الباحث المنهج التطبيقي باستخدام Protégé Framework مع OWL Language من أجل بناء الأنطولوجيا.

ذُكرت في طيات هذا البحث الخطوات المتبعة في بناء الانطولوجيا كما تم توضيح القضايا المعقدة في التصنيف الهرمي لفئات النظام موضوع الدراسة إبتداءاً من تحديد المجال ورصد كافة المفاهيم والخصائص والعلاقات، ثم بعد ذلك ترتيب المفاهيم (الفصائل) في شكل هرمي يوضح الفئات والفئات الفرعية بناءا على علاقة تضمين هذه الفئات. ومن ثم تم تعريف خصائص الفئات والعلاقات التي تربط بينها، كما فرضت قيود على خصائص بعض الفئات وفقا لما تنص عليه قواعد التجويد وأخيرا تم إنشاء نماذج أو أمثلة لفئات أونطولوجيا التجويد حتى يتسنى إجراء الاستعلامات المرغوب فيها. وبعد الانتهاء من تصميم الانطولوجيا، استخدم المفكر (Reasoner) المضمن في أداة الدراسة لاختبار مدى انسجام فئات أنطولوجيا التجويد، ولحساب علاقة تضمين هذه الفئات.

وتتمثل أهم نتائج تصميم انطولوجيا لمخارج الحروف وأحكام النون الساكنة والتنوين في تمثيل واضح لمخارج الحروف وأحكام النون الساكنة والتنوين وخصائصها والعلاقات بينها؛ كما

يمكن أيضاً من خلالها إجراء الاستعلامات لاسترجاع أية معلومات عن مخارج الحروف وأحكام النون الساكنة والتنوين وخصائصها. بالإضافة إلى أنه يمكن مشاركة هذه الانطولوجيا وإعادة استخدامها في التطبيقات ذات الصلة بمجال القرآن الكريم.

TABLE OF CONTENTS

CHAPTER ONE : INTRODUCTION

CHAPTER TWO: LITERATURE REVIEW

CHAPTER THREE: ONTOLOGY BUILDING TOOLS

CHAPTER FOUR: BUILDING TAGWEED ONTOLOGY

LIST OF FIGURES

LIST OF TABLES

TABLE OF ABBREVIATIONS

Abbreviation	Full Name
XML	eXtensible Markup Language
OWL	Web Ontology Language
RDF	Resource Description Framework
HTML	Hyper Text Markup Language
KIF	Knowledge Interchange Format
W3C	World Wide Web Consortium
DAML	DARPA Agent Markup Language
OIL	Ontology Inference Layer
DL	Description Logics
SPARQL	SPARQL Protocol and RDF Query Language
CLIR	Cross Language Information Retrieval
ZVTM	Zoomable Visual Transformation Machine
API	Application Programming Interface
OKBC	Open Knowledge Base Connectivity protocol

CHAPTER ONE

This chapter narrates the historical background of ontology and gives brief introduction about Quran Tajweed, addition to the research methodology.

1. 1 Introduction

1.1.1 Ontology

Historically, ontologies arise out of the branch of philosophy known as metaphysics, which deals with the nature of reality – of what exists. This fundamental branch is concerned with analyzing various types or modes of existence, often with special attention to the relations between particulars and universals, between intrinsic and extrinsic properties, and between essence and existence. The traditional goal of ontological inquiry in particular is to divide the world "at its joints" to discover those fundamental categories or kinds into which the world’s objects naturally fall [1].

Since the mid-1970s, researchers in the field of artificial intelligence (AI) have recognized that capturing knowledge is the key to building large and powerful AI systems. AI researchers argued that they could create new ontologies as computational models that enable certain kinds of automated reasoning. In the 1980s, the AI community began to use the term ontology to refer to both a theory of a modeled world and a component of knowledge systems [1].

In the context of computer and information sciences, ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application.

In the context of database systems, ontology can be viewed as a level of abstraction of data models, analogous to hierarchical and relational models, but intended for modeling knowledge about individuals, their attributes, and their relationships to other individuals.

Ontologies are typically specified in languages that allow abstraction away from data structures and implementation strategies; in practice, the languages of ontologies are closer in expressive power to first-order logic. For this reason, ontologies are said to be at the "semantic" level. Due to their independence from lower level data models, ontologies are used for integrating heterogeneous databases, enabling interoperability among disparate systems, and specifying interfaces to independent, knowledge-based services. In the technology stack of the Semantic Web standards, ontologies are called out as an explicit layer. There are now standard languages and a variety of commercial and open source tools for creating and working with ontologies [1].

In recent years, there has been a global growing demand for Islamic knowledge by both Muslims and non-Muslims, especially knowledge relating to the Quran, as it is the main source of knowledge, law and wisdom in the Islamic religion.

1.1.2 Holy Quran

The Holy Quran is the word of Allah Almighty revealed by His Messenger Muhammad (peace be upon him). It is the primary source of universal laws and regulations for all Muslims and non-Muslims. The Holy Quran is a well-structured source of information. It consists of 30 divisions (Juz), 114 chapters (Surah) and 6236 verses (Ayah) covering many themes and concepts that make up divine knowledge and law. Muslims need to understand it as well as Allah in order to properly worship him.

Quran Tajweed is the set of rules governing the way in which the words of the Quran should be pronounced during its recitation.

1.2 Problem Statement:

The Holy Quran's ontology uses knowledge representation to define the key concepts in the Holy Quran in order to facilitate its understanding for Muslims and Non-Muslims. Although several Quran search applications have been built to facilitate the retrieval of knowledge from the Quran, such as Qurany, Quran Explorer, Tanzil, and Quranic Arabic Corpus, there is still no ontology for Quran Tajweed available to help the users in getting their desired knowledge.

1.3 Problem Significance

Developing an integrated ontology model for the Holy Quran Tajweed will facilitate understanding the Holy Quran Tajweed. Additionally, it will enhance the ability to search for desired knowledge within the model.

1.4 Research Question/Hypothesis/Philosophy

1.4.1 Research Question

The main question that addressed in this research was how to construct an integrated ontology model for the Holy Quran Tajweed. There are also some additional sub-questions:

1. What are the suitable mechanisms to help build such a model?
2. What is the format of the existing ontologies?
3. What are the search techniques being used in the existing ontologies?

1.4.2 Research Hypothesis

The most important factor seen in all different types of searches is the need to find exactly what you were looking for. This will be achievable if we use an ontology-based approach.

1.4.3 Research Philosophy

The philosophy behind building an integrated model for the Holy Quran Tajweed is based on the want to use appropriate methods to construct ontologies whilst using techniques for the goal of achieving the optimal semantic search.

1.5 Research Objective

The main objective of this research is to investigate techniques that will:

1. Design an integrated model for the Holy Quran Tajweed knowledge using ontology.

2. Save the Designed Model in a format that allows combining it with other ontologies.

3. Evaluate the models accuracy under several differing circumstances.

1.6 Open Issues:

From the literature review section, it is clear that Quran ontologies remain an open problem, and there are some issues as follow:

- There are still some open research issues that need to be addressed like Tajweed ontology development.
- How can building an integrated ontology model for Holy Quran related knowledge?
- How can developing a complete application that allows searching and learning with a semantic based method?

1.7 Research scope

- Investigating a mechanism for developing ontology for part of Quran Tajweed.
- Achieving semantic searches in the Holy Quran Tajweed.

1.8 Proposed Solution

- Use one of Ontology building Tools to construct ontology for Holy Quran Tajweed.
- Save the generated ontology in appropriate formats.
- Use ontology-based approaches for enhancing the search features to become accurate and specific.

1.9 Research Methodology

To begin with, we investigated and assessed the more state of the art models in this area by reviewing the ontologies and techniques related to the Quran and Information Extraction. Also, different Quran ontologies were reviewed to determine the discussion surrounding current ontologies. Based on this research a new Integrated Model for Quran Tajweed was suggested, because the researcher found no ontologies on this aspect. This new model should take into consideration the different ontology formats, as they will be combined to produce the most powerful model possible for semantic searches in the Holy Quran Tajweed in the future.

CHAPTER TWO

LITERATURE REVIEW

This chapter discusses the issues of ontology and ontology engineering in extra level of details. Also shows some ontology studies in Holy Quran domain.

2.1 Ontology

2.1.1 Ontology Definition

There are many interpretations about what ontology is. In fact, hot discussions are often done in many meetings on ontology. However, the ontology community has come to an agreement on giving up its definition. The following are some of the definitions of ontology.

- **In philosophy, it means**

Theory of existence, it tries to explain what is being and how the world is configured by introducing a system of critical categories to account things and their intrinsic relations.

- **From AI point of view**

Ontology is defined as "explicit specification of conceptualization" [1] which is widely accepted in AI community. "Conceptualization" here should be interpreted as "intentional" rather than "extensional".

- **From knowledge-based systems point of view**

"A theory (system) of concepts/vocabulary used as building blocks of an information processing system". In a context of problem solving, ontologies are divided into two types: Task ontology for problem solving process and domain ontology for the domain where the task is performed.

- **Another definition given by Gruber**

Ontologies are agreements about shared conceptualizations. Shared conceptualizations include conceptual frameworks for modeling domain knowledge; content-specific protocols for communication among inter-operating agents; and agreements about the representation of particular domain theories. In the knowledge sharing context, ontologies are specified in the form of definitions of representational vocabulary. A very simple case would be a type hierarchy, specifying classes and their subsumption relationships. Relational database schemata also serve as ontologies by specifying the relations that can exist in some shared database and the integrity constraints that must hold for them.

- **A compositional definition is given as follows**

An ontology consists of concepts, hierarchical (is-a) organization of them, relations among them (in addition to is-a and part-of), axioms to formalize the definitions and relations [2] (Figure2.1).

2.2. Ontology of Ontologies

Depending on their intend use, ontologies come in variety of types, varying in their formality and their specificity.

2.2.1 Informal Ontologies

- **Controlled vocabulary** perhaps the simplest kind of ontology is controlled vocabulary. In a controlled vocabulary, we make use of a few predefined keywords to classify entities: no perhaps, properties, or axioms. In some domains, this is sufficient.

- **Terms/glossary** here, we have a list of terms, as with a controlled vocabulary, but some attempt is made (typically with natural language, e.g. an English explanation) to define the meaning of these terms. However, the computational power of such ontology is roughly that of a controlled vocabulary, since we cannot in general compute with the natural language explanation. The value of the explanation is therefore usually for the ontology designer.

- **Thesaurus** a thesaurus defines synonyms: terms that have the same meaning. Thus, if you want to find a web service that provides weather forecasts, it might be useful to know that 'meteorological forecast' means the same thing.

- **Informal 'is-a' taxonomies** here we think of controlled vocabularies organized into an informal hierarchy. We find such hierarchies on web sites such as Amazon.com, for example, where goods for sale are organized into loose hierarchies.

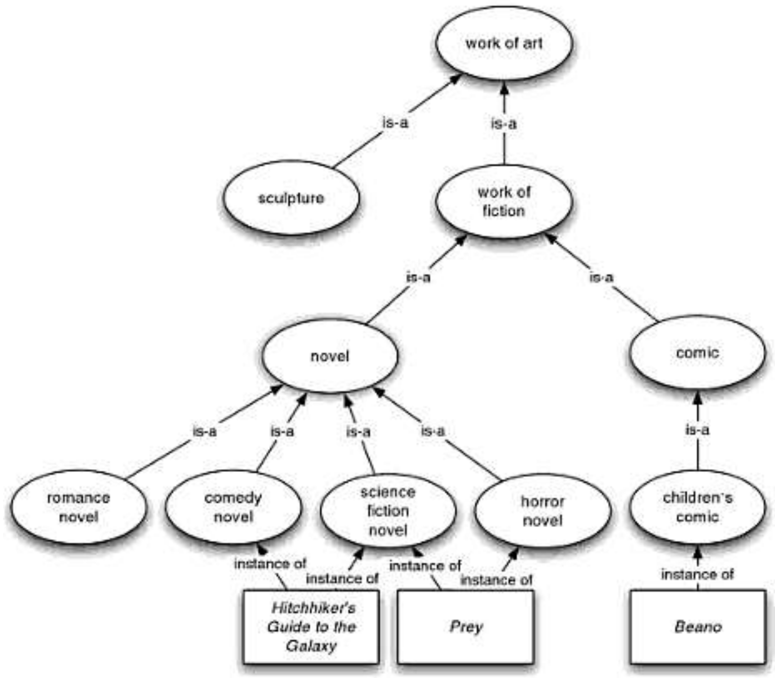

Figure 2.1: Example of Work of Art ontology [3].

Typically the hierarchies are not formal hierarchies, because related goods (e.g. cameras and camera bag) are collected together in the same place, without any formal definition of how or why the goods are clustered in this way [3].

2.2.2 Formal Ontologies

Where some attempts is made to give the terms in the ontology some formal semantics.

- **Formal 'is-a' taxonomies** here, we explicitly define subsumption relationships between classes as shown in (figure 2.2).

- **Properties** here, we now allow classes to have properties, and together with the subsumption relation, this permits us to draw conclusions about the properties of classes.

- **Value restrictions** value restrictions give additional information about relationships; for example, a typical restriction might say that 'every person has exactly one birth mother'.

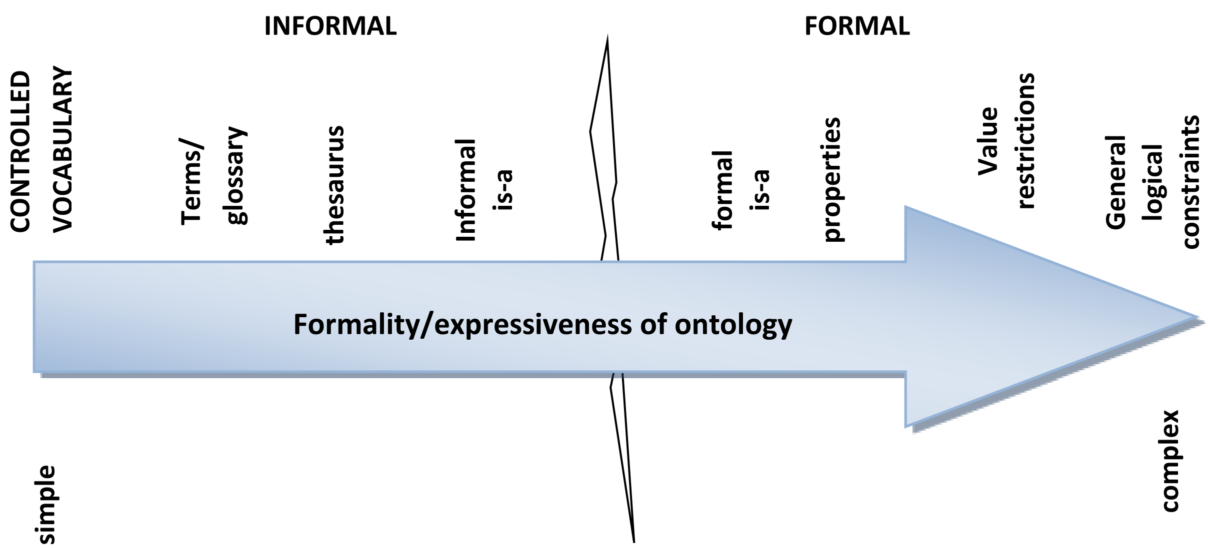

Figure 2.2: The ontology spectrum from informal up to formal expressive [3].

- **Arbitrary logical constrains** finally, we might have ontologies with arbitrary logical constrains. Such constrains go beyond value restriction, taxonomical hierarchies, etc. In general, such constrains allow us a great degree of precision when defining an ontology. The drawback with such constrains is that, in general allowing arbitrary logical expressions leads to very high computational complexity (and even undecidability) with respect to reasoning [3].

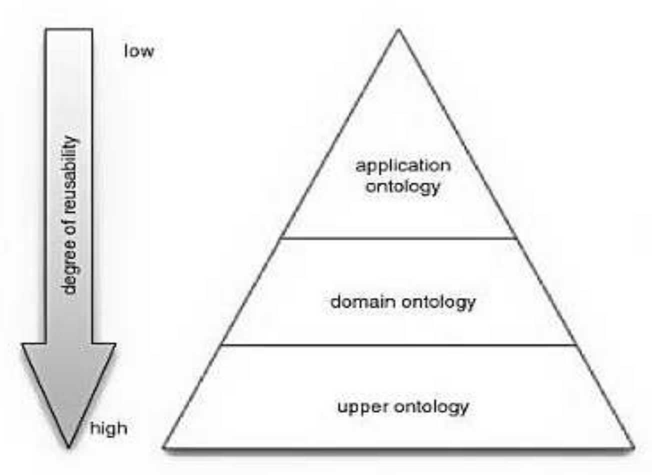

Figure 2.3: The ontology hierarchy: Degree of reusability [3].

As well as distinguishing ontologies based on their formality and the types of information (Figure 2.2), can also usefully distinguish ontologies based on their role in an application (Figure 2.3). At the bottom, the most general kind of ontology: the so-called '*upper ontology* ' (even though it appears at the bottom in (Figure 2.3) such an ontology might start out by defining the most general class and then define classes that specialize this.

Domain ontology defines concepts appropriate for a specific application domain, this research is domain ontology. For example, it might define concepts relating to Learning terminology, and be used by a number of applications in the area of Open Learning. Note that domain ontology will typically build upon and make use of concepts from an upper ontology: this idea of *reuse* of ontologies is very important, as the more applications use a particular ontology, the more agreement there will be on terms.

Finally, ***application ontology*** defines concepts used by a specific application. Again, it will typically build upon a domain ontology and it turn upon some upper ontology. Concepts from application ontology will not usually be reusable: they will typically be of relevance only within the application for which they were defined [3].

2.3 Ontology languages

2.3.1 XML – ad hoc ontologies

Arguably the simplest ontologies to create and use are *ad hoc* ontologies: create with little effort, for a specific purpose, usually with a short expected period of use. Often such ontologies take the form of a controlled vocabulary, and XML is usually the language of choice of such ontologies.

The extensible markup language (XML) is not an ontology language, although it can be directly used to define simple ontology in an informal way. It is best thought of as a kind of extension to HTML, which in a nutshell allows us to define our own tags and document structures [3].

(a) plain HTML

```
<ul>
    <li> <em> agent book</em>,
        <b>micheal wooldridge</b>,
        USD50<br> <p>
    <li> <em> software engineering book</em>,
        <b>ian summerville</b>,
        USD40<br><p>
</ul>
```

(b) XML

```
<catalogue>
      <product type=”book”>
          <title> agent</title>
          <author> micheal wooldridge </author>
          <price currencey=”USD”> 50 </price>
     </product>
    <product type=”book”>
          <title> software engineering </title>
          <author ian summerville </author>
          <price currencey=”USD”> 40 </price>
   </product>
</catalogue>
```

Figure 2.4: plain HTTL versus XML

It is clear that computer program would have a much easier time *understanding the meaning* of (Figure 2.4,b) than it would with (Figure 2.4,a), there is nothing to help a program understand which part of the document refers to the price of the product, which refers to the author of the product, and so on. We can think of the tags in (Figure 2.4, b) as being a controlled vocabulary, providing a useful but relatively informal ontology.

2.3.2 Ontolingua

Ontolingua is originally an Interlingua for ontology representation and sharing developed by KSL (Knowledge Systems Lab) at Stanford University [9]. It is designed by adding frame-like representation and translation functionalities to KIF (Knowledge Interchange Format) [KIF] which is a logic-based interlingua for knowledge representation. It can translate from and to some description logics languages such as Loom, Epikit, etc. Ontolingua itself does not have inference functionality. It has currently developed into a development environment which Provides a set of ontology development functions (browse, create, edit, modify and use ontologies) and a library of modular and reusable ontologies. Although it had been a key language for ontology representation for years since its development, it is not active recently because of the advent of XML family languages.

2.3.3 RDF(S)

Resource Description Framework (RDF) is a framework for metadata description developed by W3C (WWW Consortium). It employs the triplet model <object, attribute, value>, well-known in AI community, in which object is called resource representing a web page. A triplet itself can be an object and a value. Value can take a string or resource. Object and value are considered as a node and attribute as a link between nodes. Thus, an RDF model forms a semantic network. RDF has an XML-based syntax (called serialization) which makes it resembles a common XML-based markup language. But, RDF is different from such a language in that it is a data representation model rather than a language and that the XML's data model is the nesting structure of information and the frame-like model with slots.

Although RDF has been designed for metadata representation model, it can be used as a general-purpose knowledge representation, which might be apparent from the fact that it is a kind of semantic network model.

2.3.4 Web Ontology Language (OWL)

Web Ontology Language (OWL) is also a language developed by W3C. OWL is designed to make it a common language for ontology representation and is based on DAML+OIL. OWL is an extension of RDF Schema and also employs the triple model. Its design principle includes developing a standard language for ontology representation to enable semantic web, and hence extensibility, modifiability and interoperability are given the highest priority. At the same time, it tries to achieve a good trade-off between scalability and expressive power.

OWL is a collection of several XML-based ontology frameworks, within which ontologies in these various frameworks can be expressed. Specifically, there are three main level of OWL, as follows:

- **OWL Lite.** This is simplest (least expressive) variant of OWL, which supports only basic ontology features. In particular, OWL Lite places a number of restrictions on the type of axioms one can write. The point about these restrictions is that they result in a language that is computationally more tractable (and is also somewhat easier for human to use and understand) than more expressive OWL variants.

- **OWL DL.** This language extends the properties of OWL Lite. The features of OWL DL were carefully chosen so that the language corresponds exactly to a particular formalism know as description logic.

- **OWL Full.** This is very expressive framework, providing many features for defining ontologies; however, in its full glory, the framework is so rich that many reasoning problems with OWL full (such as consistency checking) are understandable [3].

Because OWL is the most recent development in standard ontology languages and providing many features for defining ontologies, it will be adopted to be the language used to designing the ontology for Holy Quran Tajweed.

2.4 Ontology components

Contemporary ontologies share many structural similarities, regardless of the language in which they are expressed. As mentioned above, most ontology describes individuals (instances), classes (concepts), attributes, and relations. In this section each of these components is discussed in turn.

Common components of ontologies include:

- **Individuals:** instances or objects (the basic or "ground level" objects).
- **Classes:** sets, collections, concepts, classes in programming, types of objects, or kinds of things.
- **Attributes:** aspects, properties, features, characteristics, or parameters that objects (and classes) can have.
- **Relations:** ways in which classes and individuals can be related to one another.
- **Function terms:** complex structures formed from certain relations that can be used in place of an individual term in a statement.

- **Restrictions:** formally stated descriptions of what must be true in order for some assertion to be accepted as input.
- **Rules:** statements in the form of an if-then (antecedent-consequent) sentence that describe the logical inferences that can be drawn from an assertion in a particular form.
- **Axioms:** assertions (including rules) in a logical form that together comprise the overall theory that the ontology describes in its domain of application. This definition differs from that of "axioms" in generative grammar and formal logic. In those disciplines, axioms include only statements asserted as a priori knowledge. As used here, "axioms" also include the theory derived from axiomatic statements.
- **Events:** the changing of attributes or relations [4].

2.5 Ontology Engineering

Ontology engineering (or ontology building) is a subfield of knowledge engineering that studies the methods and methodologies for building ontologies. It studies the ontology development process, the ontology life cycle, the methods and methodologies for building ontologies, and the tool suites and languages that support them.

Ontology engineering aims to make explicit the knowledge contained within software applications, and within enterprises and business procedures for a particular domain. Ontology engineering offers a direction towards solving the interoperability problems brought about by semantic obstacles, such as the obstacles related to the definitions of business terms and software classes.

There is no one correct methodology for developing ontologies. It is usually an iterative process. We can start with a rough first pass at the ontology and then revise and refine the evolving ontology. Ontology is a model of a real domain in the world and the concepts in the ontology must reflect this reality. After defining an initial version of the ontology, we can evaluate and debug it by using it in applications or problem-solving methods or by discussing it with experts in the field. As a result, we will almost certainly need to revise the initial ontology. This process of iterative design will likely continue through the entire lifecycle of the ontology.

Developing Ontology may include:

- Selection of Domain and Scope.
- Consider Reuse.
- Find out Important Terms.
- Defining Classes and Class Hierarchy.
- Defining Properties of Classes and Constraints.
- Create Instances of classes [5].

Figure 2.5 show the main steps in the methodology, which will be illustrate as follow:

Step 1: Determine domain and scope

Development of an ontology starting by defining its domain and scope. That is, answer several basic questions:

- What is the domain that the ontology will cover?
- For what we are going to use the ontology?
- Who will use and maintain the ontology?

The answers of these questions may change during the ontology-design process, but at any given time they help limit the scope of the model [3].

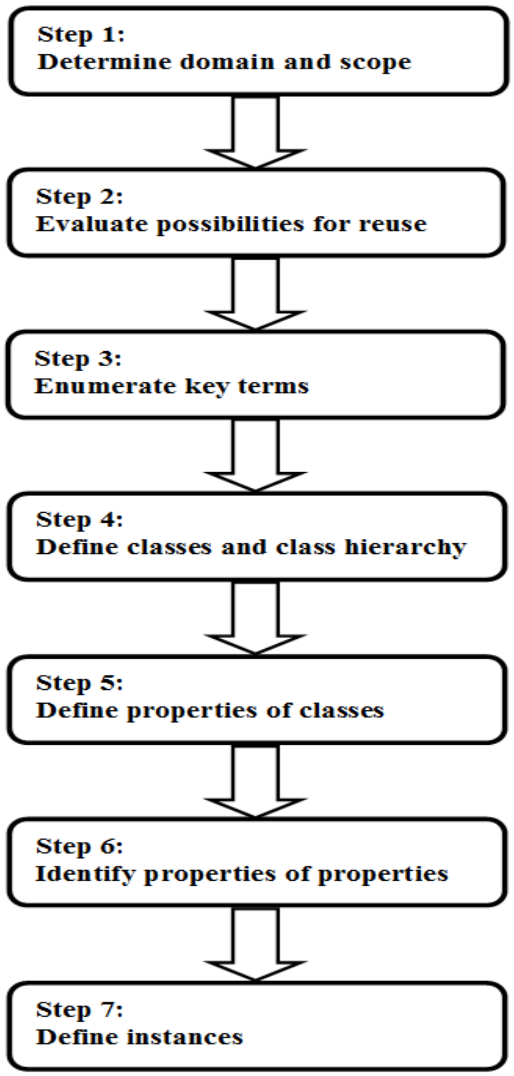

Figure 2.5: Ontology Development [3].

Step 2: Consider reuse

Reuse is extremely important in ontology development, for the simple reason that ontology is most useful if everybody uses it: if we all define our own ontology for every application, then we are defeating the object of sharing meaning [3].

Step 3: Enumerate all the relevant terms

This step simply involves knowledge acquisitions about the terms associated with our domain literally, list all the domain-specific words and concepts that appear in the requirements [3].

This work was done by reference to authentic Tajweed books as well as dialogue with experts in the Tajweed field.

Step 4: Define classes and class hierarchy

This step organize the understanding of the domain by identify classes and organize them. Refining classes into hierarchies can be done either *top down* (identifying the most general classes, then next most general, and so on), or else *bottom up* (clustering similar terms into progressively more general classes) [3].

Step 5: Define properties

This step involves identifying, for each class, the properties that are associated with that class. These properties attached to the most general class that has them. Properties come in several different types:

- **Intrinsic properties** are those that relate to the nature of an object.

- **Extrinsic properties** are abstract properties such as ‘name’, which are attached to an object.
- **Component of an object** if the object is structured in some way, then it might be useful to identify its component parts.
- **Relationship** linking the objects with one another [3].

Step 6: Define properties of properties

This step is to identify the properties that each property has. Properties might be:

- Cardinality constraints.
- Type constraints.
- Range constraints.
- Domain constraints [3].

Step 7: Create instances

At this stage the ontology populate with instances of classes [3].

2.6 Literature Review:

2.6.1 Quran Ontology

Ontologies can be seen as formal representations of knowledge. They have been investigated in many artificial intelligence studies including semantic web, software engineering, and information retrieval. The aim of ontology is to develop knowledge representations that can be shared and reused [6].

Muslims believe that the Qur'an is God's word and the most widely read book in the world since its revelation; every Muslim can memorize and recite some parts of the Quran at least 17 times every day when praying. Its recitation and reading have not stopped one day since its revelation. The Qur'an includes a range of knowledge in different subjects such as science, art, stories and history, agriculture and industry, human and social relations, and organization of finance, education and health. For some Muslims who do not speak Arabic, and for non-Muslims, the Qur'an is difficult to understand, although it has been translated into over 100 different languages [7].

By reviewing subsequent and more recent literature it has been found that most of the studies are still in infancy and much more work needs to be done to complete the ontology of the Holy Quran. Moreover, most of the Holy Quran's ontology has been developed based on English and Arabic, and so does not have a complete application that allows searching and learning with a semantic based method.

2.6.2 Related Work:

Several initial studies have been undertaken on the topic of Qur'an ontology. Most of these studies have been developed in order to improve the efficiency of information retrieval for the Qur'an. These have facilitated the process of accessing Qur'an knowledge. However, they vary from each other in different aspects such as coverage of the Qur'an, discourse level, language of the text used; original Arabic text or other translation, domain focused on, number of concepts and types covered, concept extraction method, relation types they provide, development process they followed during construction, technology used in ontology construction, availability, and verification method [7].

Work in [8] investigates the possibility of linking Holy Quran verses with web multimedia resources (audio, video and written materials) related to the Holy Quran, using ontology tagging. Also, it suggests a framework for using Holy Quran ontology terms in annotating and linking related web multimedia resources, and demonstrates a prototype of the proposed web multimedia enabled Holy Quran browser.

Firstly it discusses the research efforts that have been conducted for building Holy Quran ontology tagging generally. Then it specifies the details of the proposed system and its components. Finally it discusses a sample run of the web multimedia enabled Holy Quran browser. The main contribution of this paper is linking Holy Quran verses with web multimedia resources.

The main limitations to this paper are the misleading descriptions of multimedia resources and the replication of the same content with different descriptions that occurs on the web.

Work in [9] proposes a model which would help Muslims and non-Muslims to semantically retrieve verses in relation to their queries relating to the Holy Quran. It proposed an ontology assisted semantic search system in the Qur'ans domain. The system would be composed of a Quran concepts and semantic search model. The ontology would be used to link the concepts found in the Holy Quran with various relationships that existed between these concepts. The system would use the existing ontology from the University of Leeds, UK, which is graphically represented in a network of 300 linked concepts with 350 relations. It would use protégée software to store this

Quran ontology. The tool suggested here would provide a reasoning capability to make inference over their linked data.

The main contribution of this paper is improving the search capability of knowledge within the Holy Quran, where the system retrieves concepts and their various linked relationships with a 95% level of accuracy.

Work in [10] talks about using an ontology approach to represent Al-Quran concepts that can be classified and organized according to specific themes. It proposed and developed an ontology model according to the themes of Al-Quran knowledge as describe in Symmil Al-Quran Miracle the Reference. The ontology approach here headed the use of conventional methods such as taxonomy, hierarchy or tree structure. The main contribution of this paper is to facilitate the understanding of Al-Quran knowledge by presenting said Al-Quran knowledge in a way that is theme-based. However, the main limitation in my point of view is that the paper does not mention how to use the ontology to categorize Al-Quran knowledge (muhkamat _semantic sentence is clear_ and mutashabihat _ semantic sentence is not clear) which requires a high level of complex explanations.

Work in [11] is concerned with the use of association rules to extract the Quran ontology where the manual acquisition of ontologies from Quran verses could be very costly. For this an intelligent system designed for Quran ontology construction using pattern based schemes is needed, and associations rules to discover Quran concepts and semantics relations from Quran verses is required. The proposed system is based on the use of a combination of statistics and linguistics methods which would be used to extract concepts and conceptual relations from the Quran. In particular, a linguistic pattern-based approach would be exploited to extract specific concepts from the Quran, with the conceptual relations being found more via the association rules technique. The main contribution of this paper is building ontologies for the Quran's corpus, providing a powerful representation of Quran knowledge and showing the relations between all classes of connected concepts in the Quran ontology. However the main limitation is that it's more complicated than other methods.

Work in [12] proposed a simple methodology for automatic extraction of a concept based on the Qur'an in order to build ontology. This paper used a method based on extracting the verses which contain a word of prayer in it as well as the previous and next verse. This method relies on a format of one English translation of the Qur'an that included some aspects such as Uppercase Letter. An uppercase letter is used to identify the concepts such as the Book. Another feature called compound noun is used to identify the relationship of hyponym or "Part-OF" between the concepts. A copula is used to identify the syntactic relationship between subject and adjective or noun. The ontology is based on the information obtained from domain experts.

The development process is adopted from [13]. However, the authors have focused on the subject of Prayer or "Salat" particularly in daily prayer, thus this ontology does not cover all subjects in the Qur'an. In addition, there is no mention about underlying format or ontology technologies used in this paper.

Saad et al, continued their work to develop a framework for automated generation of Islamic knowledge concrete concepts that exist in the holy Qur'an as presented in [14].

The framework takes into account some situations form the sciences of the Qur'an, such as the cause of revelation (Asbab Al Nuzul), and verses overridden by related verses that were revealed later (Nasikh Mansukh).

The methodology of ontology development was also adopted from [12], and the method to obtain the concepts is applying a grammar and extraction rules to the English translation of the Qur'an. The 374 extracted instances only cover verses that have the keyword salah or pray and this does not cover the entire Qur'an. These instances were mapped to six abstract concepts. This paper differs from the previous in synonym relations.

Work in [15] proposed methods for designing an ontology based on translated texts of the Qur'an. Information used in developing the ontology was collected by the domain experts. Their ontology also only covers the subject of "Salat" (pray). Another ongoing research project on a prototype of a framework called SemQ is presented in [16].

SemQ identifies opposition relationships between Quranic concepts. The idea is SemQ receives a verse as input and produces a list of words that are opposed to each other with the degree of the opposition. The coverage is in the domain of "Women" in the Qur'an. Ontology development makes use of the Buckwalter morphology POS annotation and focuses on nouns and verbs that are related to the semantic field of Time. This paper used OWL and UPON technologies in order to represent the concepts and relations. The ontology consists of seven abstract concepts and eleven concrete concepts. This ontology is sharable and can be downloaded. This study was limited to word level which includes only nouns and verbs of the "Women" domain. However, there are no evaluable results provided by the authors or any validation attempts for their proposed framework.

In [17], a theme-based approach is proposed to represent and classify the knowledge of the Qur'an using ontology. Their ontology was developed according to themes described in Syammil Al-Quran Miracle the Reference, and using protégé-OWL and Malay language as medium of concepts, and was validated by the domain experts. It only covers two themes: "Iman" which means faith and "Akhlaq" which means deed.

This was an Ontology-based approach to represent and classify Quranic concepts according to specific semantic fields. The structure of the ontology was verified by Qur'an domain experts. The ontology was developed using Protégé-OWL and using Malay Language as the medium language. The authors proposed a representation approach which differs from traditional representation which consists of Juz, Chapter and Verse. There is no explanation of what language was used for this ontology and what source the concepts were based on. They implemented the ontology using protégé. There are no details of results or validation of the ontology, although the paper states that the process of creating the ontology was reviewed by seven Qur'an domain experts.

Work in [18] developed a simple ontology for the Qur'an that includes the animals that are mentioned in the Qur'an in order to provide Qur'anic semantic search. The ontology was built using protégé editor, and SPARQL query language was used to retrieve the answers to a query. The English translation of the Qur'an by Pickthall is used in this ontology. The ontology provides 167 direct or indirect references to animals in the Qur'an obtained based on information mentioned in a book entitled "Hewanat-E-Qurani". The relationship type is a taxonomy relation. The paper concludes that the existing Arabic WordNet does not help for retrieving this type of document information.

Work in [19] proposed a model for defining the important Quranic concepts by knowledge representation and presented the relationships between them using Description Logic (Predicate logic). They reused the Quranic Arabic Corpus ontology by [20]. This ongoing research attempts to reuse and improve an existing ontology developed in Leeds by adding more relations. Protégé is used in ontology construction. A top-down ontology development process was followed. It has 15 abstract concepts.

Work in [21] has proposed ontology-based semantic search for the Qur'an using protégée editor and Manchester OWL query language. The ontology was built by reusing the existing Quranic Arabic Corpus ontology developed by (Dukes 2013), and adding more than 650 relationships depending on the Qur'an, Hadith, and Islamic websites. This ontology was constructed manually.

Work in [22] proposed a semantic search for the Qur'an based on Cross Language Information Retrieval (CLIR). They created a bilingual ontology for the Quran composed of concepts based on existing Quranic Arabic Corpus ontology by (Dukes 2013), and found 5695 documents belonging to a main concept, where 541 documents are not assigned to any concepts in an English translation. In Malay, there are 5999 documents assigned to main concepts, where 237 documents do not belong to any concept.

In [23], the authors did experiments on retrieving verses of the Quran using a semantic query approach exploiting Cross Language Information Retrieval (CLIR).

Work in [24] developed a tool for searching for the Quranic concrete and abstract concepts. She exploited an existing Qur'an topics index from a scholarly source: Tafsir of Ibn Kathir. This ontology covered the whole Qur'an.

In [20], in his PhD thesis defines 300 concepts in the Qur'an, and extracts the interrelationships using Predicate logic. The number of relations is 350. The type of relation between concepts is Part-of or IS-A. The ontology is also based on the Tafsir by Ibn Kathir.

In [25], developed an ontology covering the whole Qur'an in terms of pronoun tagging. Each pronoun is linked to its antecedent or previous subject.

In [26], the authors have created a dataset called QurSim which consists of 7600 pairs of related verses for evaluating the relatedness of short texts.

An automatic knowledge extraction method based on rules and natural language patterns is described in [27]. Their methods rely on the English translation of the Qur'an and have identified a new pattern language named Qpattern which is suitable for extraction of taxonomy part-of relations. This research also identified that it is difficult to extract information from text that includes co-reference like the Qur'an.

CHAPTER THREE

ONTOLOGY BUILDING TOOLS

This chapter introduces some of ontology building tools, and Protégé-OWL in more details, Protégé components and its platforms that modeling the ontologies.

3.1 ISAVIZ:

IsaViz is a visual environment for browsing and authoring RDF models as graphs. This tool is offered by W3C Consortium. IsaViz [28] was developed by Emmanuel Pietriga. The first version was developed in collaboration with Xerox Research Centre Europe which also contributed with XVTM, the ancestor of ZVTM (Zoomable Visual Transformation Machine) upon which IsaViz is built. As of October 2004, further developments are handled by INRIA Futurs project In Situ. IsaViz also includes software developed by HP Labs (Jena 2 Semantic Web Toolkit), the Apache Software Foundation (Xerces Java 2), and makes use of the GraphViz library developed by AT&T Research. IsaViz does not follow or include any methodology for building ontology. IsaViz imports RDF/XML and N-Triples, and exports RDF/XML [13], N-Triples, Portable Network Graphics (PNG) and Scalable Vector Graphics (SVG). Therefore, it is possible to import ontologies to other editors, for instance, Protégé or OilEd. The IsaViz environment is composed of four main windows: the IsaViz RDF Editor window, the Graph window, the Definition window and the Attribute window [28].

3.2 APOLLO

Is a user-friendly knowledge modeling application. The modeling is based around the basic primitives, such as classes, instances, functions, relations etc. Internal model is built as a frame system according to the internal model of the OKBC protocol.

Apollo's class system is modeled according to the OKBC. The knowledge base consists of ontology's that are hierarchically organized. Ontology can inherit other ontology's and then use classes of inherited ontology's as its own. Every ontology inherits at least one ontology − a default ontology, which contains all primitive classes: Boolean, integer, float, string, list etc. Class contains slots of two types: non template and template slots.

Apollo currently does not support non template class slots. For each class is possible to create a number of instances. An instance inherits all slots of the class. Each slot has a set of facets [29].

3.3 SWOOP

Is a Web-based OWL ontology editor and browser. SWOOP contains OWL validation and offers various OWL presentation syntax views. It has reasoning support and provides a multiple ontology environment. Ontologies can be compared, edited and merged. Different ontologies can be compared against their Description Logic-based definitions, associated properties and instances.

SWOOP's interface has hyperlinked capabilities so that navigation can be simple and easy. SWOOP does not follow a methodology for ontology construction. Users can reuse external ontological data. This is possible either by purely linking to the external entity, or importing the entire external ontology. It is not possible to do partial imports of OWL. There are several ways to achieve this, such as a brute-force syntactic scheme to copy/paste relevant parts (axioms) of the external ontology, or a more elegant solution that involves partitioning the external ontology while preserving its semantics and then reusing (importing) only the specific partition as desired.

It is possible to search concepts across multiple ontologies. SWOOP makes use of an ontology search algorithm, that combines keywords with DL-based in order to find related concepts. This search is made along all the ontologies stored in the SWOOP knowledge base [30].

3.4 Protégé

Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats [31].

Protégé can be customized to provide domain-friendly support for creating knowledge models and entering data. Further, Protégé can be extended by way of a plug-in architecture and a Java-based Application

Programming Interface (API) for building knowledge-based tools and applications [31].

Ontology describes the concepts and relationships that are important in a particular domain _as mentioned in previous chapter, providing a vocabulary for that domain as well as a computerized specification of the meaning of terms used in the vocabulary. Ontologies range from taxonomies and classifications, database schemas, to fully axiomatized theories.

The Protégé platform supports two main ways of modeling ontologies:

- **The Protégé-Frames editor** enables users to build and populate ontologies that are frame-based, in accordance with the Open Knowledge Base Connectivity protocol (OKBC). In this model, ontology consists of a set of classes organized in a subsumption hierarchy to represent a domain's salient concepts, a set of slots associated to classes to describe their properties and relationships, and a set of instances of those classes - individual exemplars of the concepts that hold specific values for their properties.
- **The Protégé-OWL editor** enables users to build ontologies for the Semantic Web, in particular in the W3C's Web Ontology Language (OWL). "OWL ontology may include descriptions of classes, properties and their instances. Given such ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. [31].

3.5 Protégé-OWL

The Protégé-OWL is an extension of Protégé that supports the Web Ontology Language (OWL). OWL is the most recent development in standard ontology languages, endorsed by the World Wide Web Consortium (W3C) to promote the Semantic Web vision. "OWL ontology may include descriptions of classes, properties and their instances. Given such ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms.

The Protégé-OWL Editor enables users to:

- Load and save OWL and RDF ontologies.
- Edit and visualize classes and properties.
- Define logical class characteristics as OWL expressions.
- Execute Reasoner such as description logic classifiers.
- Edit OWL individuals for Semantic Web markup. Protégé-OWL's flexible architecture makes it easy to configure and extend the tool [31].

3.6 OWL Ontologies

Ontologies are used to capture knowledge about some domain of interest. Ontology describes the concepts in the domain and also the relationships that hold between those concepts. Different ontology languages provide different facilities. The most recent development in standard ontology languages is OWL from the World Wide Web Consortium (W3C). Like Protégé, OWL makes it possible to describe concepts but it also provides new facilities. It has a richer set of operators e.g. intersection, union and negation. It is based on a different logical model which makes it possible for concepts to be defined as well as described. Complex concepts can therefore be built up in definitions out of simpler concepts [31].

Furthermore, the logical model allows the use of a Reasoner which can check whether or not all of the statements and definitions in the ontology are mutually consistent and can also recognize which concepts fit under which definitions. The Reasoner can therefore help to maintain the hierarchy correctly. This is particularly useful when dealing with cases where classes can have more than one parent [32]

3.7 Components of OWL Ontologies

OWL ontologies have similar components to Protégé frame based ontologies. However, the terminology used to describe these components is slightly different from that used in Protégé. OWL ontology consists of Individuals, Properties, and Classes, which roughly correspond to Protégé frames Instances, Slots and Classes [4].

Individuals: represent objects in the domain. Individuals are also known as instances (Figure 3.1). Individuals can be referred to as being 'instances of classes'.

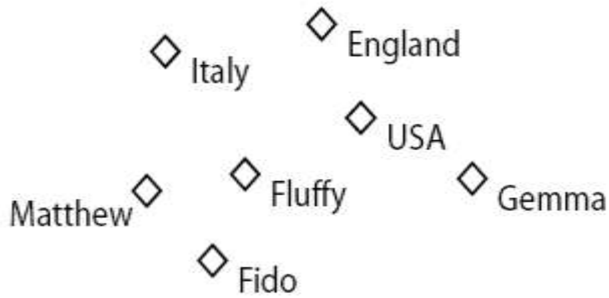

Figure 3.1: Representation of Individuals [4].

Properties: are binary relations on individuals that link two individuals together (Figure 3.2). Properties are roughly equivalent to slots in Protégé. They are also known as roles in description logics and relations in UML and other object oriented notions. In GRAIL and some other formalism they are called attributes [8].

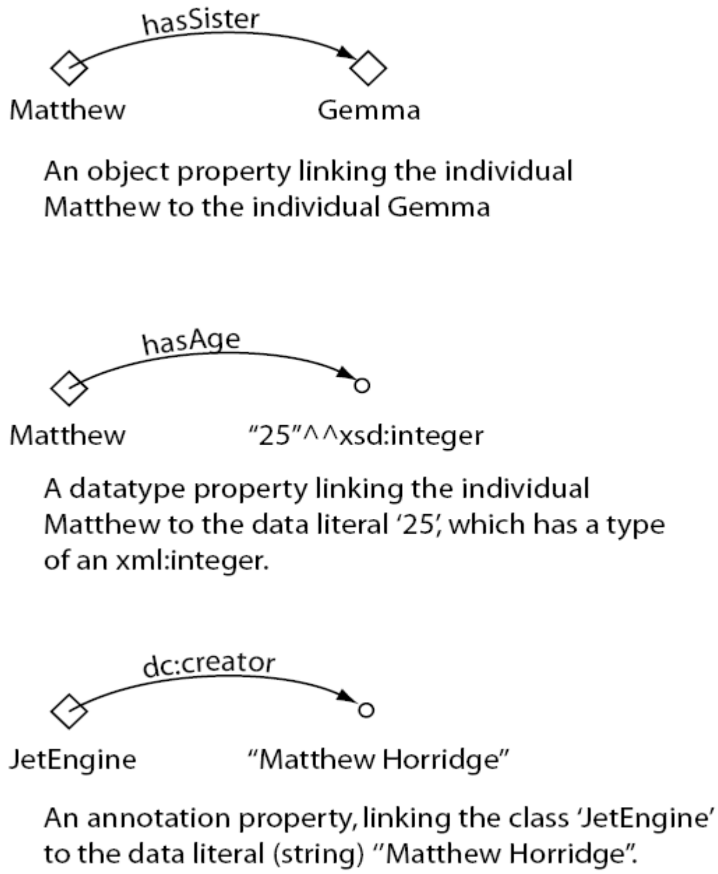

Figure3.2: Representation of Properties [8].

OWL classes: are interpreted as sets that contain individuals (Figure 3.3). They are described using formal (mathematical) descriptions that state precisely the requirements for membership of the class. The word concept is sometimes used in place of class. Classes are a concrete representation of concepts [8].

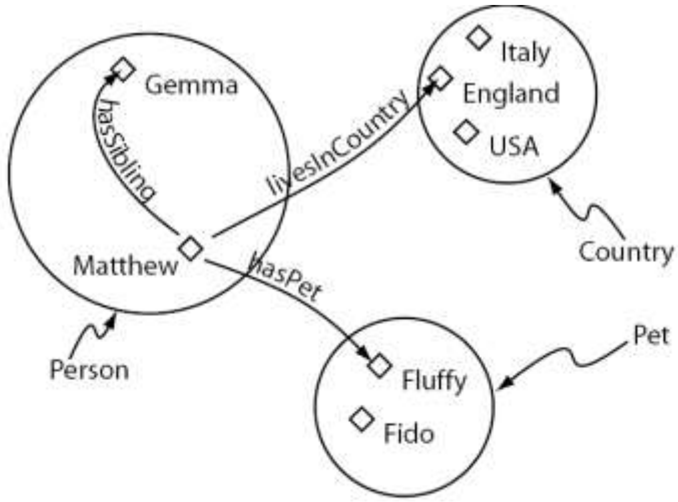

Figure 3.3: Representation Of Classes (Containing Individuals) [4].

CHAPTER FOUR

ONTOLOGY BUILDING

4.1 Introduction

The Holy Quran has rules that the reader must learn so that he can read the Qur'an correctly (Tajweed). In this chapter we design ontology for some of the provisions of Tajweed UN Vowel Noon and Tanween (أحكام النون الساكنة والتنوين) and Articulations Points of the Letters.

Un Vowel Noon (Noon sakinah) is a noon free from any vowel (حركة). It remains unchanged in its written form and as well as in pronunciation when continuing to read after it and when stopping on it. Noon sakinah occurs in nouns and verbs in the middle of the word and at the end of the word, and occurs in prepositions and particles (حروف) only at the end of the word. The noon sakinah can have a (sukoon) on it, as inمِنْهُمْ ; or can be written with no vowel on it, as inمِّن [33].

Un Vowel Noon has four Rules:

- The Clearness (الإظهار).
- The Merge (الإدغام).
- The Hiding (الإخفاء).
- The Change (الإقلاب).

The Tanween is a term for an extra noon saakinah not used for emphasis, found at the end of nouns when continuing the reading, but absent from the noun in the written form (the noon of the Tanween is pronounced but not written),and abandoned in pronunciation when stopping. The Tanween can be accompany adouble kasrah, or double dhammah, or double fathah, as in (ٍ ٌ ً) [33].

Articulation Point of the Letters is the place of emitting the letter when pronouncing it, which differentiates that specific letter from other letters.

- The empty space in the mouth and throat: has one articulation point for the three, lengthened letters: الألف، الواو، الياء.
- The Throat: has two articulations points for six different letters which are pronounced from:
  - Deepest part of the Throat: الهمزة، الهاء.
  - Middle part of the Throat: العين، الحاء.
  - Closest part of the Throat:الغين، الخاء.
- The Tongue: it has ten articulations points for eighteen letters: القاف، الكاف، الجيم، الشين،الياء غير المدية، الضاد، اللام، النون، الراء، الطاء، الدال، التاء، الصاد، الزاى، السين، الظاء، الدال، الثاء .
- The Two Lips: it has two articulations points for four letters: الفاء، الواو غير المدية، الميم، الباء.
- The Nose: from the hole of nose towards the inside of the mouth, here is one articulation point is that for Ghunnah [33].

4.2 Ontology engineering

Ontology engineering (or ontology building) is a subfield of knowledge engineering that studies the methods and methodologies for constructing ontologies. It studies the ontology development process, the ontology life cycle, the methods and methodologies for building ontologies, and the tool suites and languages that support them. Here we use Protégé which is a free, open source ontology editor and OWL which is the most recent development in standard ontology languages to build our ontology.

Ontology Engineering is a set of tasks related to the development of the ontology of a specific domain [1]. As in (Figure4.2), the main steps in the methodology are as follows:

4.2.1 Determine the domain and scope of the ontology

We start the development of ontology by defining its domain and scope. That is, answer several basic questions:

- What is the domain that the ontology will cover?
- For what we are going to use the ontology?
- For what types of questions the information in the ontology should provide answers?
- Who will use and maintain the ontology?

The answers to these questions may change during the ontology-design process, but at any given time they help limit the scope of the model [6].

We did our ontology on part of Quran Tajweed, specifically the provisions of the Un Vowel Noon and Tanween. One of the ways to determine the scope of the ontology is to write down a competency questions, which are a list of questions that the designed ontology should be able to answer them in the context of the ontology, such as:

- What is UN Vowel Noon?
- What is Tanween?
- What are the letters of Ghunnah?
- What are the UN Vowel Noon and Tanween provisions?
- What are the articulations of the letters?
- Give an example of throat's Letters?
- Give an example of Clearness?
- What is the letters of Change?
- How many letters come out from tongue?

Judging from this list of questions, the ontology will include the information that the user need to retrieve about UN Vowel Noon and Tanween provisions, articulations of the letters.

4.2.2 Consider reusing existing ontologies

Reusing existing ontologies may be a requirement if a system needs to interact with other applications that have already committed to particular ontologies or controlled vocabularies [4].

This ontology was constructing from scratch.

4.2.3 Enumerate important terms in the ontology

At this stage we write down all terms that can be in the domain. In our ontology terms can be as follow:

The Articulation Point of the Letter, The Empty Space of the Mouth and Throat, The Nose, The Two Lips, The Tongue, Deepest Part of the Tongue, Edge of the Tongue, Middle of the Tongue, Tip of the Tongue, The Throat, The Closest Part of the Throat, Deepest Part of the Throat, The Closest Part of the Throat, The Un Vowel Noon and Tanween Rules, Un Vowel Noon, The Change, The Clearness, The Hiding, The Merging, With Ghunnah, Without Ghunnah, Tanween, get out from, is part of, subsumes, has applied definition, has linguistic definition, has wards, has example, and so on [33].

Initially, it is important to get a comprehensive list of terms without worrying properties that the concepts may have, or whether the concepts are classes or slots.

4.2.4 Define the Classes and the Class Hierarchy

At this stage the concepts of the domain will be arranged hierarchically through the Class and subclass relationship as in (Figure4.3). Class hierarchy can be developed throw several adopted approaches: Top-Down and Button-up.

The ontology has been built using first approach with two main classes as follow:

- **The Articulation Point of the Letter** with its sub classes**:**

- ***The Empty Space of the Mouth and Throat.***
- ***The Nose.***
- ***The Two Lips.***
- ***The Tongue,*** with its sub classes**:**
  - *Deepest Part of the Tongue.*
  - *Edge of the Tongue.*
  - *Middle of the Tongue.*
  - *Tip of the Tongue.*
- ***The Throat,*** with its sub classes**:**
  - *The Closest Part of the Throat.*
  - *Deepest Part of the Throat.*
  - *The Closest Part of the Throat.*

- **The Un Vowel Noon and Tanween Rules,** with its sub classes:
  - ***Un Vowel Noon:***
    - *The Change.*
    - *The Clearness.*
    - *The Hiding.*
    - *The Merging,* with its sub classes:
      - With Ghunnah, with its sub classes**:**
        - Complete.
        - Incomplete.
      - Without Ghunnah.
  - ***Tanween.***

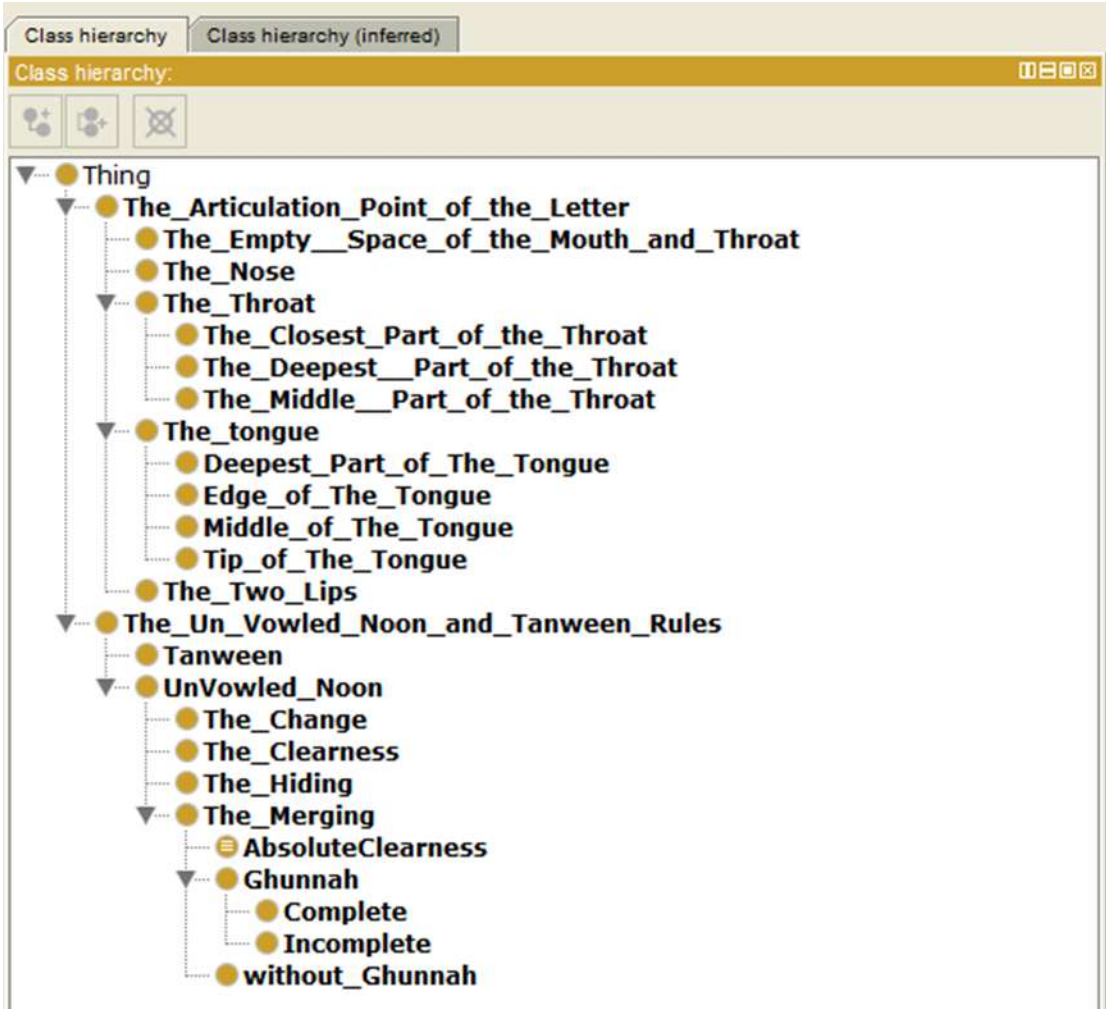

Figure4.1: The Main Classes of Tajweed ontology.

4.2.5 Define the properties of classes

We have already selected classes from the list of terms we created in Step 3, but the classes alone will not provide enough information to answer the competency questions from Step 1. Once we have defined some of the classes, we must describe the internal structure of concepts [6].

OWL Properties represent relationships. There are two main types of properties, Object properties and Datatype properties. Object properties are relationships between two individuals, Datatype properties are between individuals and Data type. OWL also has a third type of property, Annotation properties. Annotation properties can be used to add information (metadata — data about data) to classes, individuals and object/Datatype properties [4].

According to provisions of UN Vowel Noon and Tajweed ontology, the properties will be like:

- **Object Properties:**
  - ❖ get out from:
  - ❖ is part of
  - ❖ subsumes
  - ❖ has one word.
  - ❖ has two words.
  - ❖ has type.

  For example get out from: links an individual of class **Clearness** to an individual of class **Throat**.

- **Data Properties:**
  - ❖ has applied definition.
  - ❖ has linguistic definition.
  - ❖ has wards.
  - ❖ has example.

For example has linguistic definition, Links an individual of class **Clearness** to **String** Data Type.

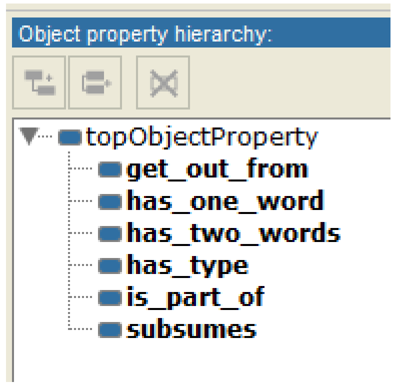

Figure4.2: Object Properties Hierarchy.

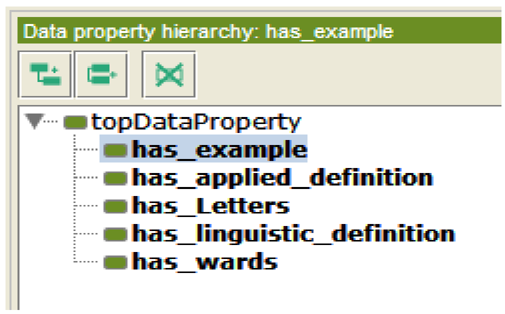

Figure4.3: Data Properties Hierarchy.

4.2.6 Define properties of properties

Here we will define properties for each property. Properties might be:

- **Cardinality constraints:** are used to talk about the number of relationships that an individual may participate in for a given property. Cardinality restrictions come in three flavors: Minimum cardinality restrictions, Maximum cardinality restrictions and exact cardinality restrictions.

For example in our ontology **Tanween** should be in two words [33], as in (Figure4.4). This constrain written use cardinality constrain as shown in (Figure4.5).

Also, when one of the letters of The **Merge** comes after the UN Vowel Noon, and it was in **one word** there is no **Ghunnah**, but called **Absolute** as in (Figure4.6), it comes only in **four words** in the Holy Quran

{صنوان، قنوان، الدنيا، بنيان} .[3]

. Such class called **Enumerated Class**

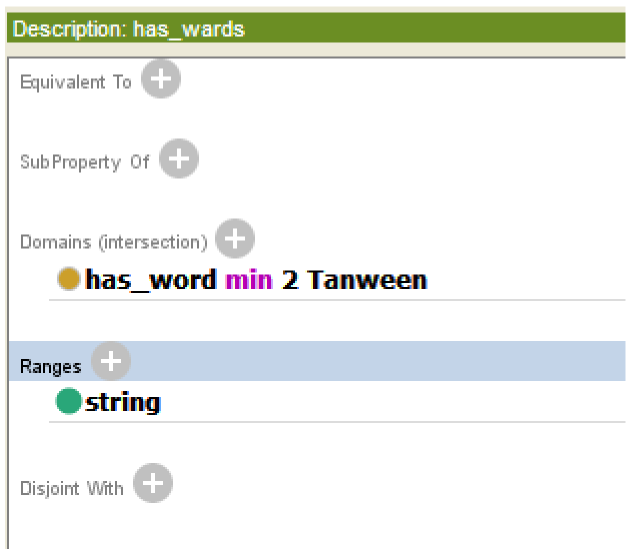

Figure4.4: has_word Property Description.

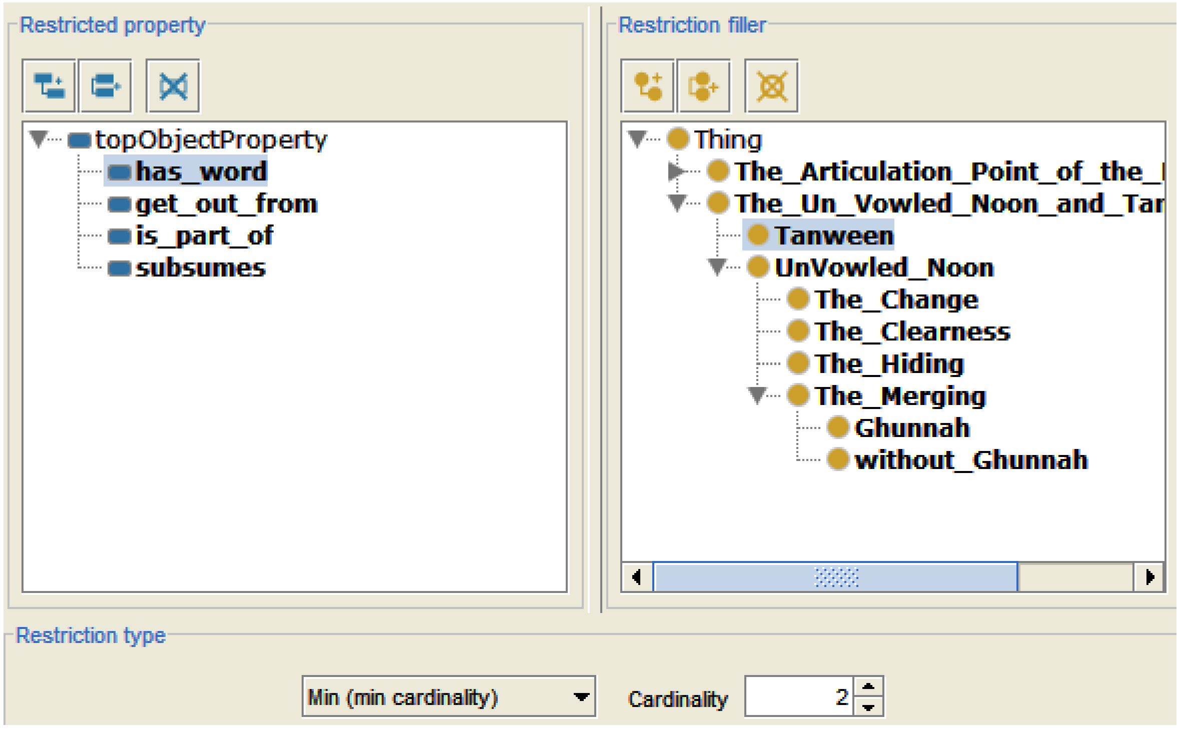

Figure4.5: Cardinality Constraint for has_word Property.

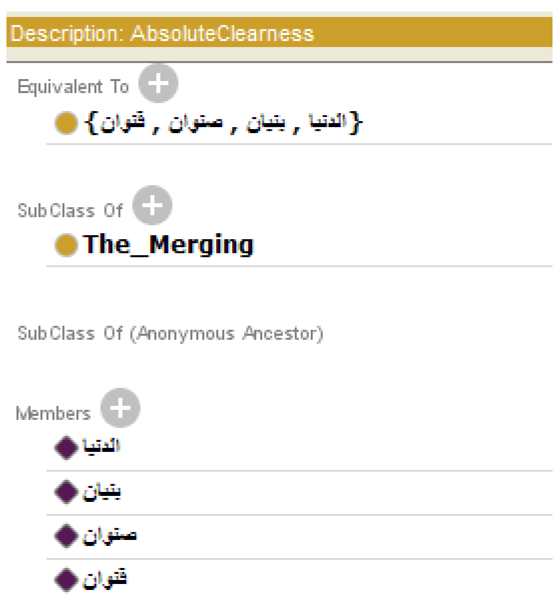

Figure4.6: Enumerate class Absolute with its members.

- **Data Type constraints**: Data type properties describe relationships between individuals and data values.

- **Range constraints and Domain constraints:** Properties may have a domain and a range specified. Properties link individuals from the domain to individuals from the range. For example, in this ontology, the property **get_out_from** would probably link individuals belonging to the class **The_Clearness** to individuals belonging to the class of **The_Throat**. In this case the domain of the **get_out_from** property is **The_Clearness** and the range is **The_Throat** this is depicted in (Figure4.7).

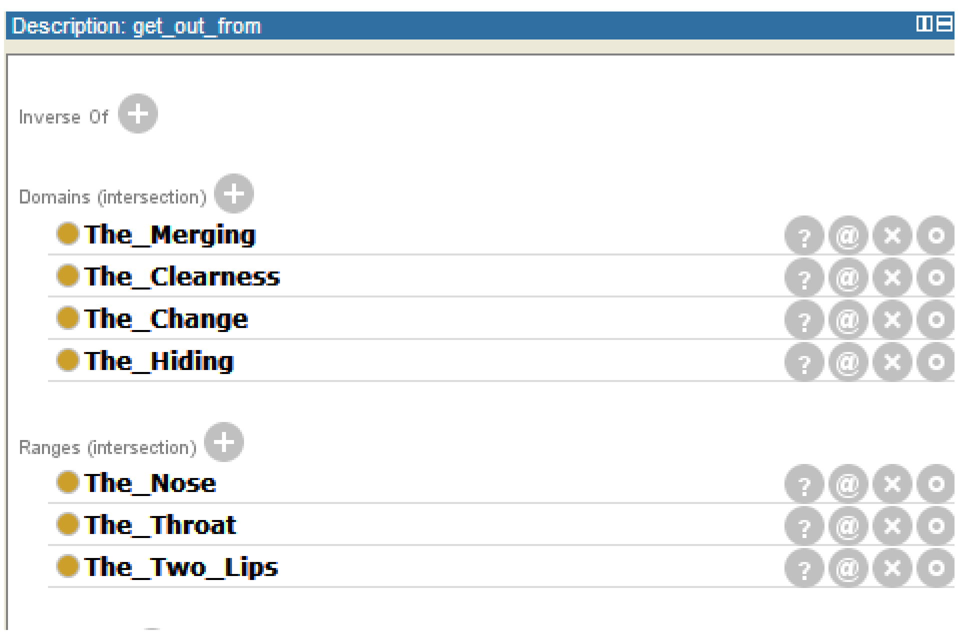

Figure4.7: Domain and Range of get_out_from Properties.

4.2.7 Create instances

At this level the ontology populate with instances of all classes. For example, class **Ghunnah** has instances such as **النون، الميم، الياء، الواو غير المدية** as shown in (Figure4.8). Also, it may have more details with verses from Quran as in (Figure4.9). And such process of populating the instances did for the rest of the ontology classes.

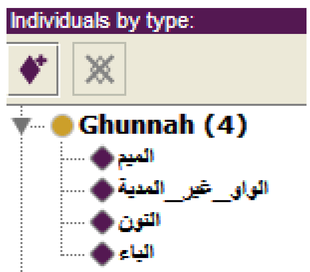

Figure4.8: Ghunnah Class Individuals.

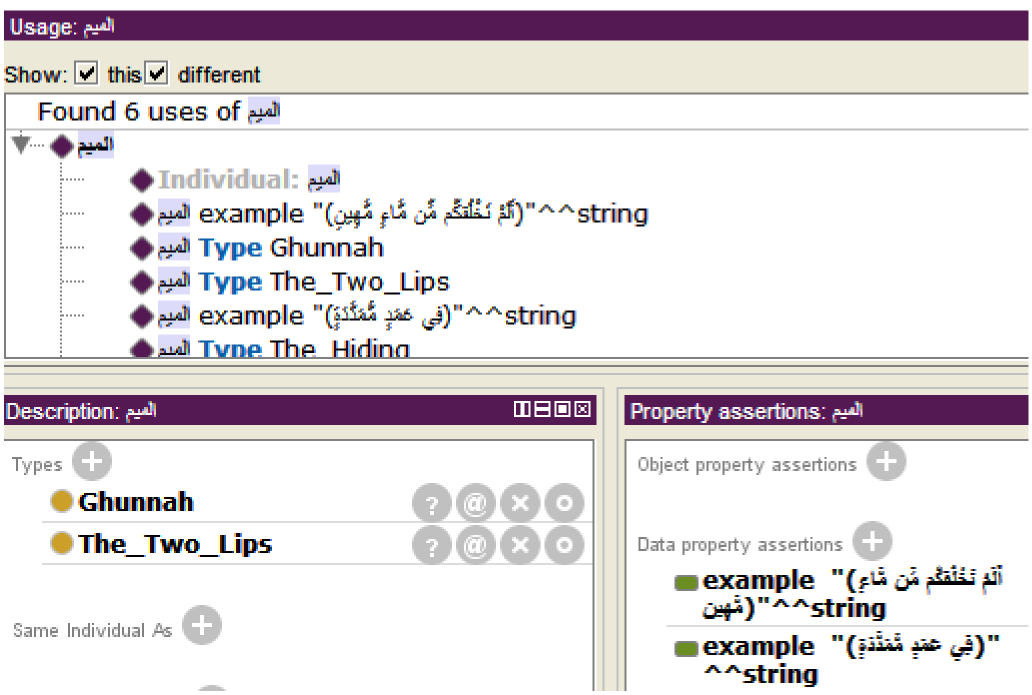

Figure4.9: Instance with more Information.الميم

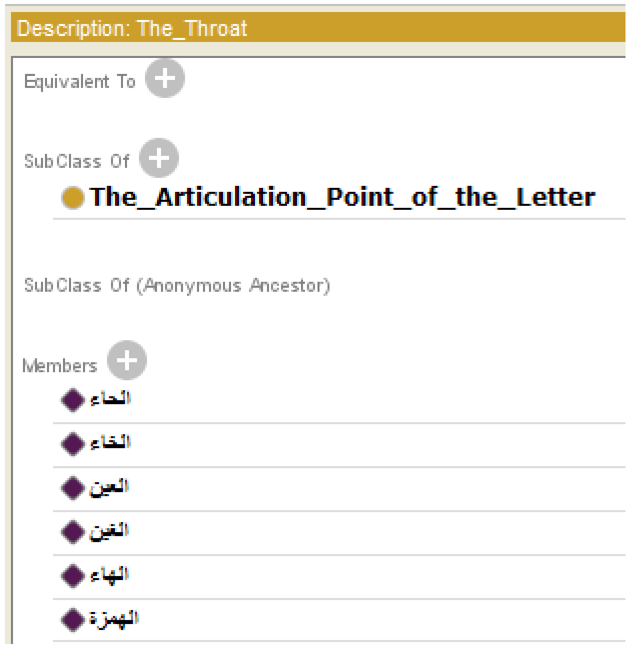

Figure4.10: The_Throat Class Individuals

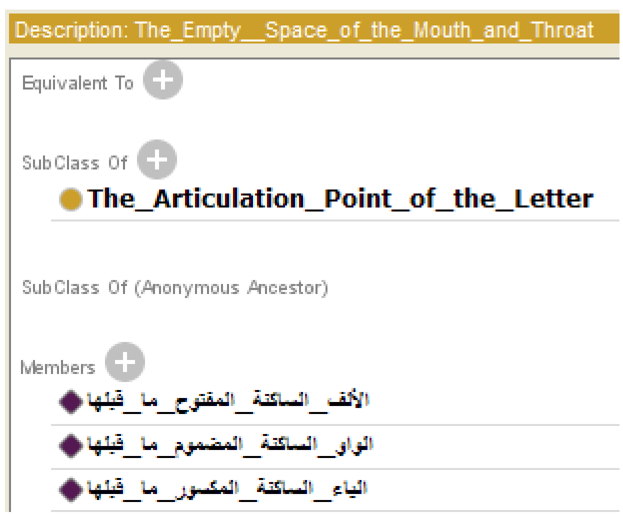

Figure4.11: Class Instances.

Visual representation of the ontology after its construction appears in (Fiure4.12). Now the ontology is ready and can make any query to retrieve knowledge about Articulation points of letters, UN vowel noon and Tanween.

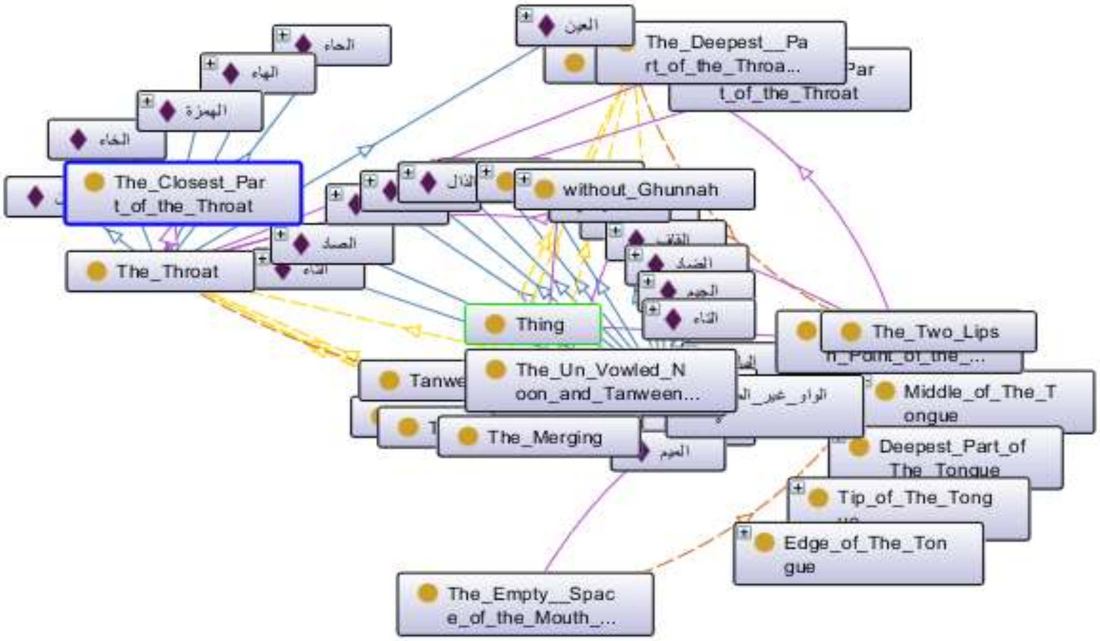

Figure4.12: Visualization of UN Vowel Noon and Tanween Classes.

4.3 Invoke the Reasoner

The Reasoner has been invoked - via the ‘start Reasoner’ button in the Reasoner drop down menu- to automatically compute the classification hierarchy, and also to check the logical consistency of the ontology. The inferred hierarchy has been computed, and an inferred hierarchy window popped on top the existing asserted hierarchy window. (Figure 4:13) and (Figure 4:14) display the difference before and after invoking the Reasoner.

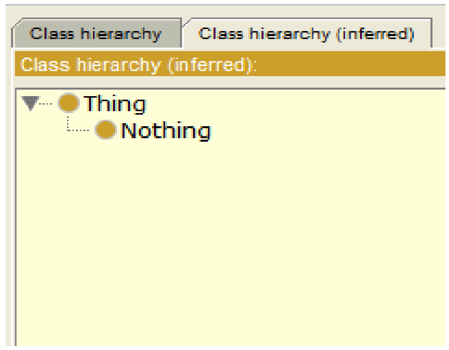

Figure4.13: Inferred class Hierarchy before invoke the Reasoner

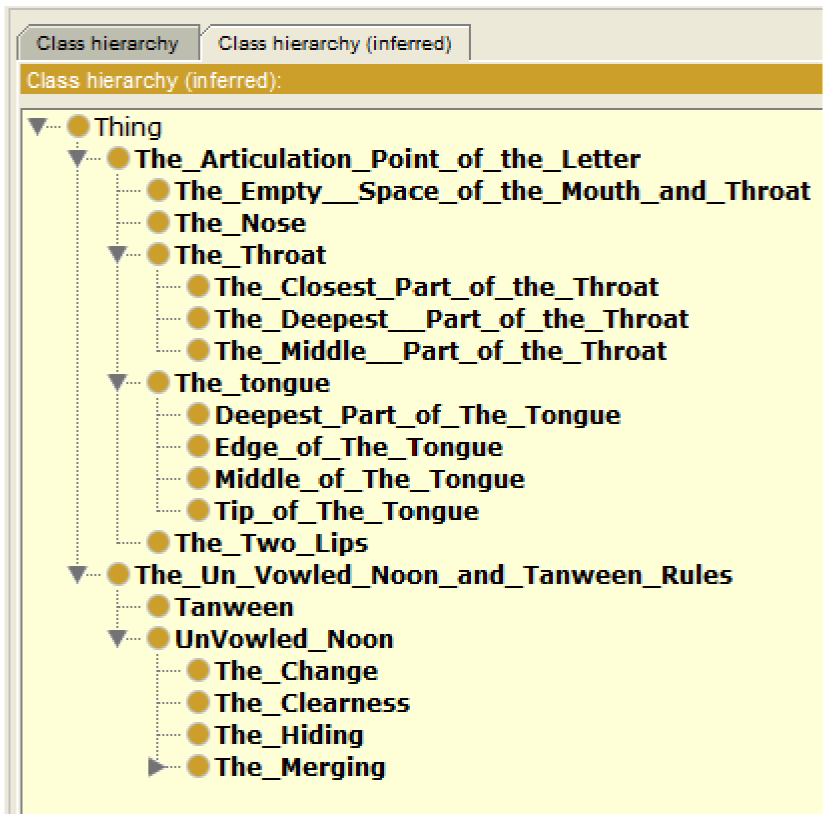

Figure4.14 Inferred class Hierarchy after invoke the Reasoner.

Protégé Owl allows storing data in variety of format, such as OWL.RDF, XML, etc. as shown in (Figure4.15).

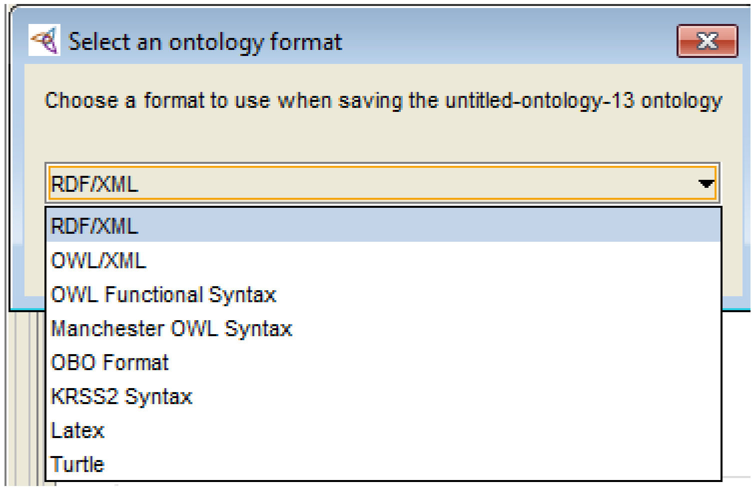

Figure4.15: Protégé File Format.

As mentioned in chapter two (Figure 2.5), to develop an ontology there are several steps to be follow. In this chapter these steps will be illustrated in more details with respect to Tajweed Ontology (and Articulations Points of the Letters, UN Vowel Noon and Tanween).

CHAPTER FIVE

5.1 Examples of the Tajweed Ontology in Protégé

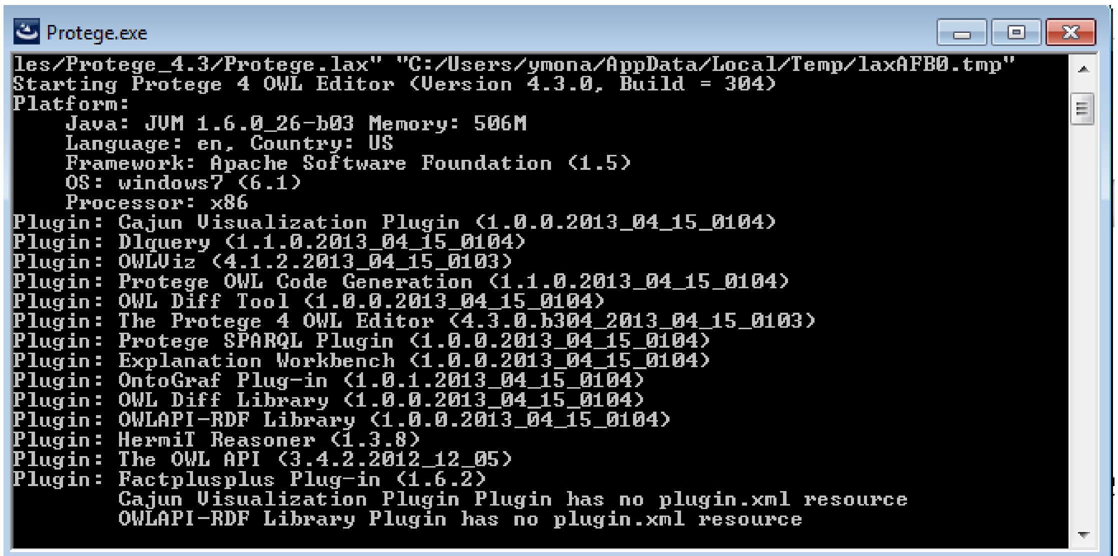

Figure5.1: Protégé Stat up Window

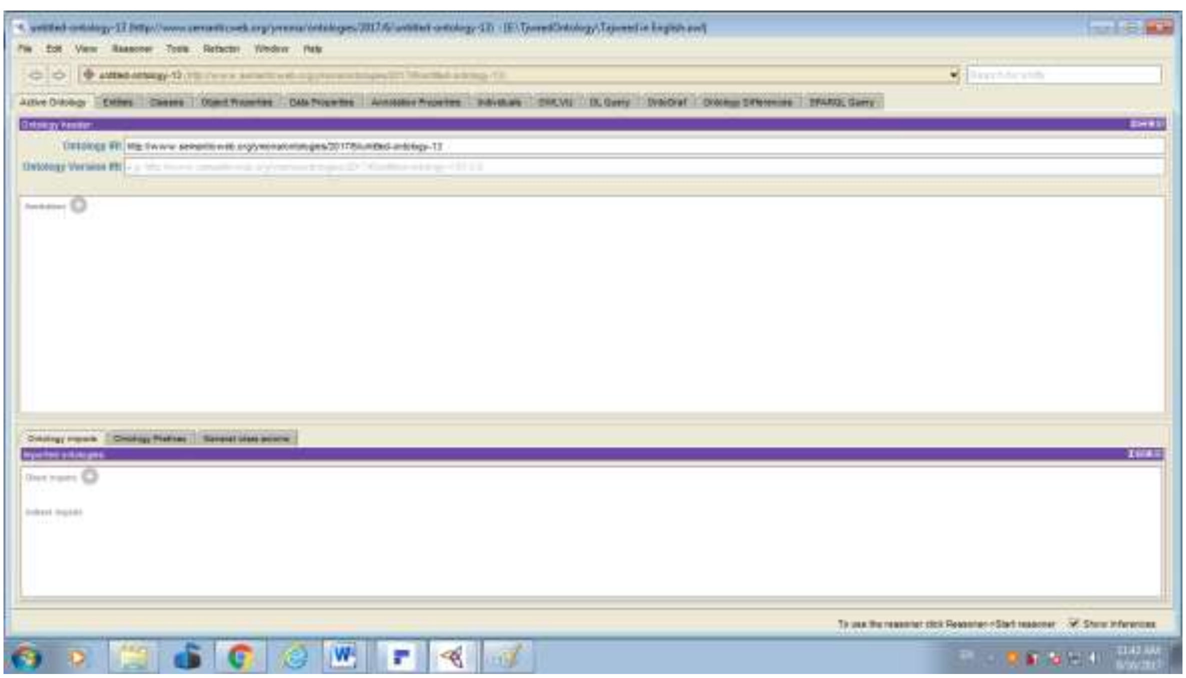

Figure5.2: Protégé main window.

Figure5.3: The_Empty_Space_of_the_Mouth_and_Throat Class

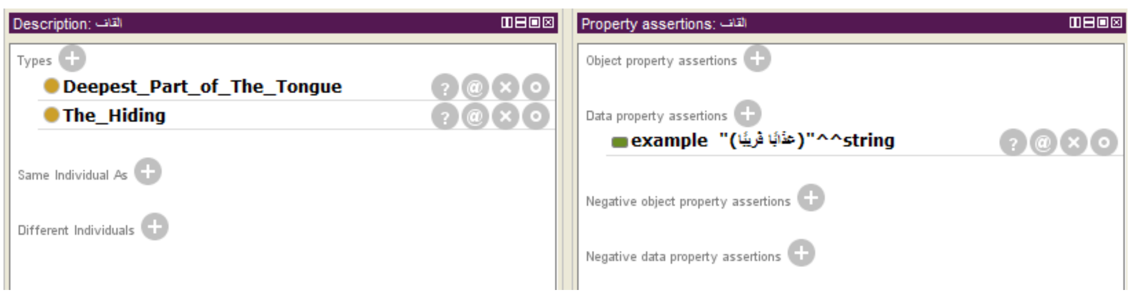

Figure5.4: Individual القاف

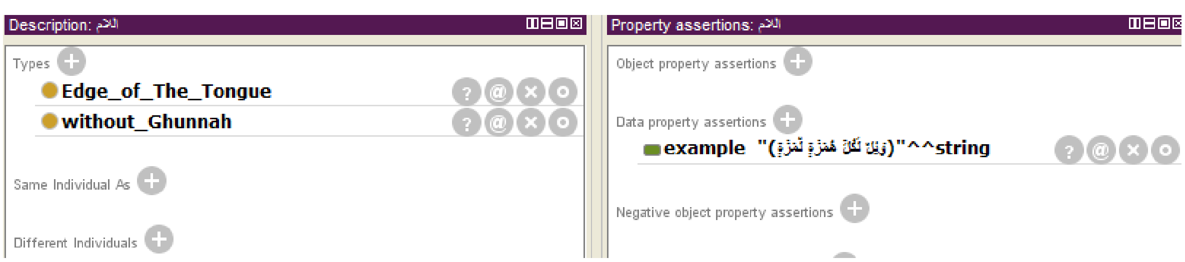

Figure5.6: Individual اللام

5.2 Results

After completing the design of the ontology, and ascertaining that all the classes are consistent with its own definitions via using the Reasoner. As a result, we have explicit representation and full definitions for the ontology objects, properties and their relationships, also can make any query to retrieve information. We used DL Query and SPARQL Query to inquire about Articulation Points of Letters, Provisions of UN Vowel Noon and Tanween.

For Example

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?subject ?object

WHERE { ?subject rdfs:subClassOf ?object }

This SPQRQL Query will retains back all subjects and objects of the Tajweed ontology as in (Figure5.7).

Also we can use DL Query to make Queries, (Figure5.8) show the result of inquired about The Merging (**الإدغام**), which is all information about Merging.

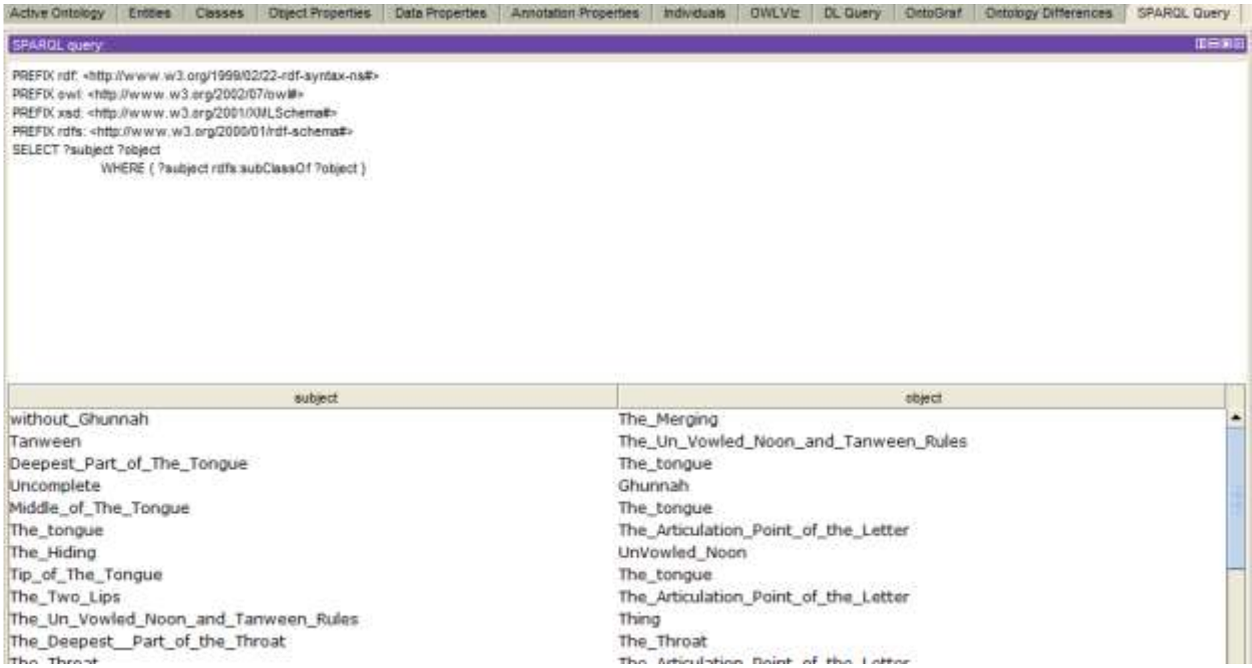

Figure5.7: Query of Whole ontology Elements.

More queries done over the ontology and the out puts as in (Figure5.9), (Figure5.10) and (Figure5.11) which are retrieve information about AbsoluteClearness class, The_Two_Lips class and Clearness class with their individuals respectively.

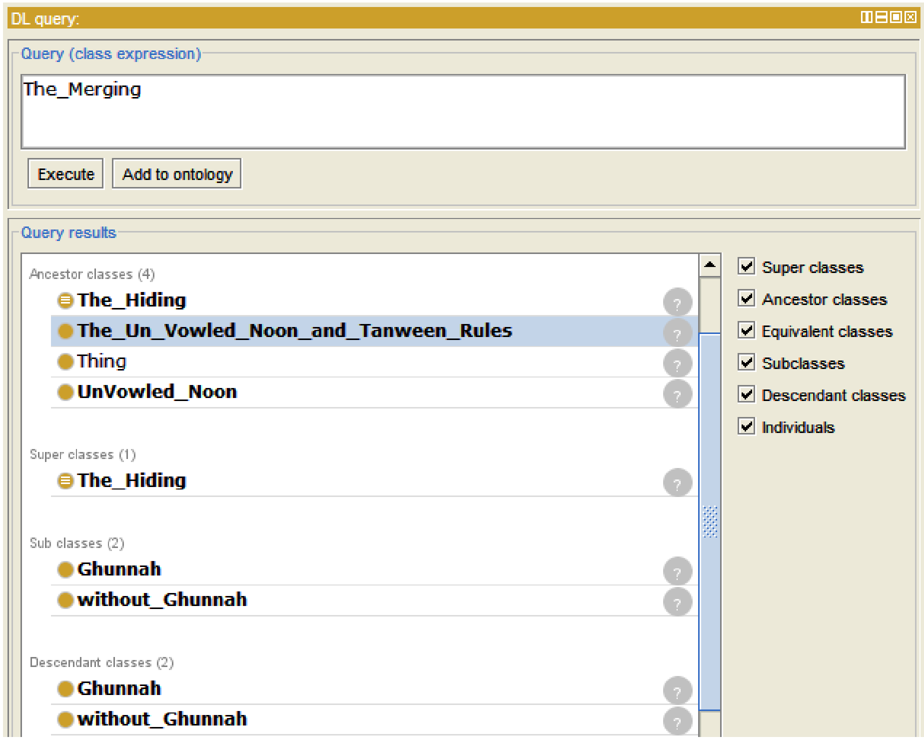

Figure5.8: The_Merging Query Result.

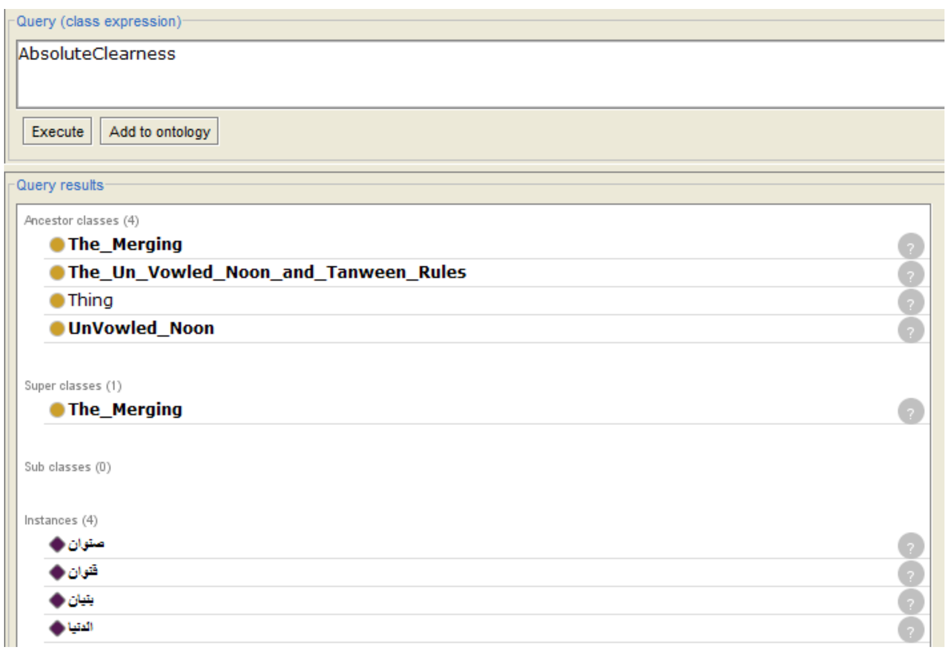

Figure5.9: Query about Absolute Class

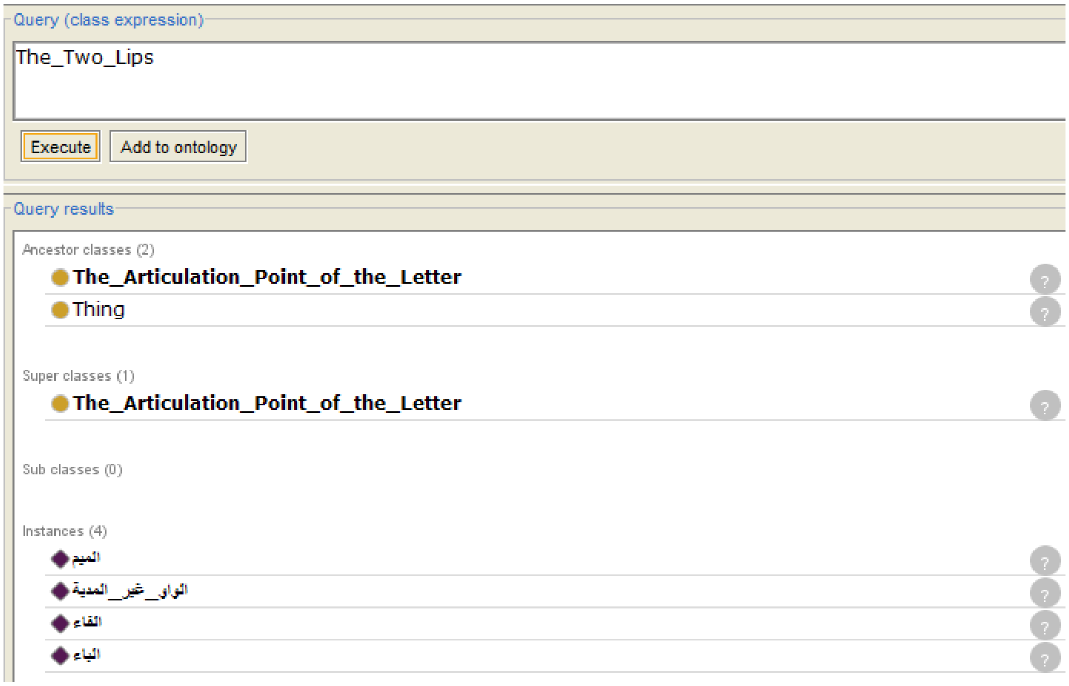

Figure5.10: Query about Absolute Class

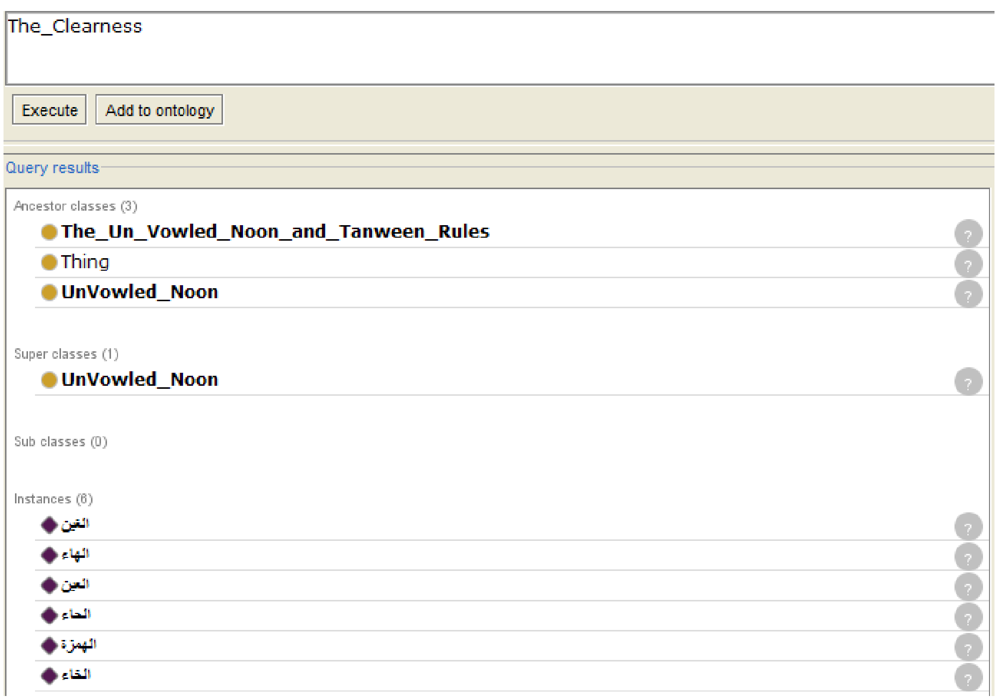

Figure5.11: Query about Clearness Class

5.3 Results Discussion

The user can specify what part of the query he wants to be appeared through the selection buttons on the right of the DL Query window, such as Super classes, Ancestor classes, Subclasses, Descendent classes, Individuals. (Figure5.9)

Furthermore, user also can do so when using SPARQL Query by selects what concept he wants to look for -Class, subclass or Object- through the below Query statement.

SELECT ?subject ?object

WHERE { ?subject rdfs:subClassOf ?object }

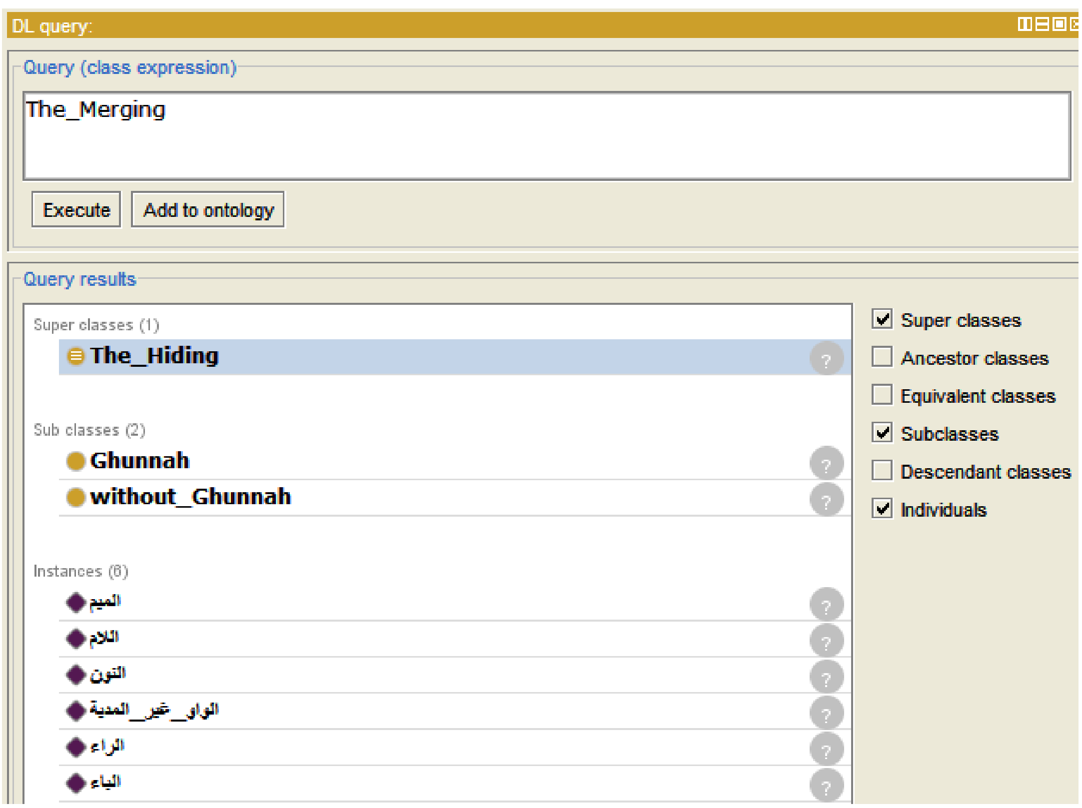

Figure5.9: Selective Member.

5.3 Research Evaluation

The Holy Quran Tajweed ontology has been evaluated by scholarly sources (Tajweed Books) and domain experts at Sudan University of Science and Technology, Institute for Science & Islamic Research, Department of Islamic Studies & Research.

The model tested 20 times and gets absolute correct answers.

Table 5.1: Result of Queries on Tajweed Ontology

No of Queries	**Result**			**Response Time**
	Correct	Fault	Accuracy	0.49 sec
20	20	0	100%	

This gives us information about quality of the work that has been conducted.

CHAPTER SIX

6.1 Conclusion

The Holy Quran is the main book of Islam, and Muslims believe that the Holy Quran is word of God, as spoken to the Prophet Mohammed (PBUOH), and it is thus his Miracle. The Holy Quran's ontology uses knowledge representation to define the key concepts in the Holy Quran in order to facilitate its understanding for Muslims and Non-Muslims.

In this research Protégé Framework and OWL were used to design the ontology for Provisions of Quran Tajweed, enumerated the terms that can possibly found in the domain, then organized them in a hierarchy depending on which class subsumes another, and defined properties for each class. After that the steps went deeply to define constraints on the properties according to what the rules of Tajweed stipulate, such as: cardinality constraints, domain and range constraints. Final step in designing the ontology was defining some instances to be able to make queries. After that, the Reasoner used to check the consistency of the classes and to generate inferred class hierarchy. The designed ontology makes any query to retrieve information about the provisions of UN Vowel Noon and Tanween. Furthermore, it can be shared and reused in applications related to Holy Quran Tajweed.

6.2 Research Contributions

The research highlighted a significant aspect of the Muslim's life, namely the recitation of the Holy Quran, beside:

- Enriching the library of Islamic Knowledge with a new ontology (Tajweed ontology).
- Building Ontology for Holy Quran Tajweed, thus, can be shared and reused in different Islamic Applications.
- Published Papers in Holy Quran Tajweed.

6.3 Published Papers

Bellow, the papers that researcher was published:

1. **Design Ontology for Open Learning Domain**:

Mohammed Awad Mohammed AtaElfadiel, Eiman Alsiddig Altayeb Ibrahim, International Journal of Science and Research (IJSR), Volume 5 Issue 5, May 2016.

Available at: https://www.ijsr.net/archive/v5i5/NOV161981.pdf.

2.Provisions of Quran Tajweed Ontology (Articulations Points of the Letters, UN Vowel Noon and Tanween),

Eiman Alsiddig Altayeb Ibrahim, Mohammed Awad Mohammed Ataelfadiel and Eric Steven Atwell, International Journal of Science and

Research (IJSR), Volume 6 Issue 8, August 2017.

Available at: **https://www.ijsr.net/archive/v6i8/v6i8.php.**

1. **Ontology Life Cycle: A survey on the Ontology and its Development Steps**:

   Eiman Alsiddig Altayeb Ibrahim, Mohammed Awad Mohammed Ataelfadiel, International Journal of Emerging Technology and Advanced Engineering, Volume 7, Issue 8, August 2017.

6.4 Recommendations and Future Works

After careful study of the ontological issues and to make maximum use of this technology, the researcher recommends:

1. Publish the Articulation Points of letters, UN Vowel Noon and Tanween ontology to be available online.
2. Do more ontology researches on Provisions of Quran Tajweed.
3. Merge All Tajweed Ontologies Together to build a great Holy Quran Tajweed ontology.
4. Apply the Tajweed Ontology in other Artificial Intelligence research in Quran recitation, such as [34], [35], [36].

REFERENCES

[1] Thomas R. Gruber, “Toward Principles for the Design of Ontologies Used for Knowledge Sharing”, Stanford Knowledge Systems Laboratory, 2008.

[2] Floriano Zini , Leon Sterling, “Designing Ontologies for Agents”, Department of Computer Science Software Engineering, University of Melbourne, Parkville 30502,Australia,1999.

[3] Michael Wooldridge,” An Introduction to Multiagent Systems”, University of Liverpool, 2008.

[4] Matthew Horridge,” A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools”, Edition 1.3.

[5] Bhaskar Kapoor, Savita Sharma, “A Comparative Study Ontology Building Tools for Semantic Web Applications”, International journal of Web & Semantic Technology (IJWesT), Vol.1, Num.3, July 2010

[6] Natalya F. Noy, Deborah L. McGuinness,” Ontology Development 101: A Guide to Creating Your First Ontology”, Stanford University, Stanford, CA, 94305.

[7] Sameer M. Alrehaili, Eric Atwell,” Computational ontologies for semantic tagging of the Quran: A survey of past approaches”, School of Computing, University of Leeds, Leeds LS2 9JT, UK.

[8] Y. Abdelhamid, M. Mahmoud, T. M. El-Sakka,“Using Ontology for Associating web multimedia resources with the Holy Quran,” in Taibah university international Conference on Advances in information technology for the Holy Quran and its sciences, proceedings volume:1, pp. 266-271, 2013.

[9] Aliyu Rufai Yauri, Rabiah Abdul Kadir, Azreen Azman and Masrah Azmi Murad, "Quranic Verse Extraction base on Concepts using OWL-DL Ontology," Research Journal of Applied Sciences, Engineering and Technology 6(23): 4492-4498, 2013.

[10] Azman Ta'a, Syuhada Zainal Abidin, Mohd Syazwan Abdullah, Abdul Bashah B Mat Ali, and Muhammad Ahmad "Al-Quran themes classification using Ontology," Proceedings of the 4th International Conference on Computing and Informatics, ICOCI, 2013.

[11] Fouzi Harrag, Abdullah Al-Nasser, Abdullah Al-Musnad, Rayan Al-Shaya, Abdulmalik Salman Al-Salman, "Using association rules for ontology extraction from a Quran corpus", College of computer & Information Science,King Saud University, Riyadh, 11543.

[12] Saad, S. & Salim, N., "Methodology of Ontology Extraction for Islamic Knowledge Text", In Postgraduate Annual Research Seminar, UTM, 2008.

[13] Saad, S., Salim, N. & Zainal, H., "Pattern extraction for Islamic concept", Electrical Engineering, August, 2009, pp.333–337.

[14] Saad, S. et al., "A framework for Islamic knowledge via ontology representation", International Conference on Information Retrieval & Knowledge Management (CAMP), 2010, pp.310–314.

[15] Saad, S., Salim, N. & Zainuddin, S., "An early stage of knowledge acquisition based on Quranic text", Semantic Technology June,2011, pp.130–136

[16] Al-khalifa, H. & Al-yahya, M.M., "SemQ: A Proposed Framework for Representing Semantic Opposition in the Holy Quran using", International Conference on the Current Trends in Information Technology (CTIT), 2009, pp.0–3.

[17] Ali, B. & Ahmad, M., "AL-QURAN THEMES CLASSIFICATION USING ONTOLOGY", Sarawak, Malaysia, International Conference on Computing and Informatics, ICOCI, August, 2013, pp.383–389.

[18] Ullah Khan, H. et al., "Ontology Based Semantic Search in Holy Quran", International Journal of Future Computer and Communication, 2013, pp.570–575.

[19] Yauri, A.R. et al., "Quranic-based concepts: Verse relations extraction using Manchester OWL syntax", International Conference on Information Retrieval & Knowledge Management, 2012, pp.317–321.

[20] Dukes, K., "Statistical Parsing by Machine Learning from a Classical Arabic Treebank", PhD Thesis, School of Computing, University of Leeds, 2013.

[21] Yauri, A.R. et al., "Ontology Semantic Approach to Extraction of knowledge from Holy Quran", Computer Science and Information Technology (CSIT), 2013 5thInternational Conference, pp.19–23.

[22] Yahya, Z. et al., "Query Translation Using Concepts Similarity Based on Quran Ontology for Cross-Language Information Retrieval. Journal of Computer Science, 2013, pp.889–897.

[23] Yunus, M. a, Zainuddin, R. & Abdullah, N., "Semantic query for Quran documents results", IEEE Conference on Open Systems (ICOS 2010, pp. 5–7.

[24] Abbas, N., "Quran search for a Concept Tool and Website". MSc Thesis, School of Computing, University of Leeds, 2009.

[25] Muhammad, A.B., "Annotation of Conceptual Co-reference and Text Mining the Qur'an", PhD Thesis, School of Computing, University of Leeds, 2012.

[26] Sharaf, A.-B. & Atwell, E., “QurSim: A corpus for evaluation of relatedness in short texts”, In N. C. (Conference Chair) et al., eds. Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12). Istanbul, Turkey: European Language Resources Association (ELRA), 2012.

[27] Saad, S., Salim, N. & Zainal, H., “Rules and Natural Language Pattern in Extracting Quranic Knowledge”, In the proceedings of the Tayba University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences (NOORIC 2013). Medina, Saudi Arabia: IT Research Center for the Holy Quran and Its Sciences (NOOR) 2013.

[28] http://www.w3.org/2001/11/IsaViz/Overview.html

[29] http://apollo.open.ac.uk/index.html.

[30] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, Bernardo Cuenca Grau, James A. Hendler, “Swoop: A Web Ontology Editing Browser”, 2006, pp. 144-153.

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

[32] E. Ibrahim,” Design ontology for Open Learning Domain”, Sudan University for Science and Technology, 2013.

[33] Ayman Rushdi Swayd, Kareema carol Czerepinski, “Tajweed rules of Quran”, Part one.

APPENDESS

APPENDEX A

Part of owl file:

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-
ns#" >
]>


<rdf:RDF
xmlns="http://www.semanticweb.org/ymona/ontologies/2017/6/unt
itled-ontology-13#"

xml:base="http://www.semanticweb.org/ymona/ontologies/2017/6/
untitled-ontology-13"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <owl:Ontology
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13"/>



    <!--

/////////////////////////////////////////////////////////////
//////////////////////////
    //
    // Object Properties
    //

/////////////////////////////////////////////////////////////
//////////////////////////
     -->
```

```
    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#get_out_from -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#get_out_from">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Change"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Clearness"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Hiding"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Merging"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Nose"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Throat"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Two_Lips"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#has_one_word -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#has_one_word">
        <rdfs:range
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#AbsoluteClearness"/>
```

```
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#Ghunnah"/>
        <rdfs:domain>
            <owl:Restriction>
                <owl:onProperty
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#has_one_word"/>
                <owl:onClass
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#AbsoluteClearness"/>
                <owl:qualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">4</owl:qualifiedCardin
ality>
            </owl:Restriction>
        </rdfs:domain>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#has_two_words -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#has_two_words"/>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#has_type -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#has_type"/>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#is_part_of -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#is_part_of">
```

```
        <rdf:type
rdf:resource="&owl;InverseFunctionalProperty"/>
        <rdfs:range
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Articulation_Point_of_the_Letter"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Empty__Space_of_the_Mouth_and_Throat"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Nose"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Throat"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Two_Lips"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_tongue"/>
        <owl:inverseOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#subsumes"/>
    </owl:ObjectProperty>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#subsumes -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#subsumes">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    </owl:ObjectProperty>



    <!--

/////////////////////////////////////////////////////////////
//////////////////////////
    //
    // Data properties
```

```
    //

/////////////////////////////////////////////////////////////
//////////////////////////
     -->




    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#example -->

    <owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#example">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Un_Vowled_Noon_and_Tanween_Rules"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#has_Letters -->

    <owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#has_Letters">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Articulation_Point_of_the_Letter"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Un_Vowled_Noon_and_Tanween_Rules"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>
```

```
    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#has_applied_definition -->

    <owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#has_applied_definition">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Un_Vowled_Noon_and_Tanween_Rules"/>
        &lt;rdfs:range rdf:resource="&xsd;string"/&gt;
    &lt;/owl:DatatypeProperty&gt;



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#has_linguistic_definition -->

    <owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#has_linguistic_definition">
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Un_Vowled_Noon_and_Tanween_Rules"/>
        &lt;rdfs:range rdf:resource="&xsd;string"/&gt;
    &lt;/owl:DatatypeProperty&gt;



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#has_wards -->

    <owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#has_wards">
        &lt;rdfs:range rdf:resource="&xsd;string"/&gt;
        &lt;rdfs:domain&gt;
            &lt;owl:Restriction&gt;
                <owl:onProperty
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#has_two_words"/>
```

```
                <owl:onClass
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#Tanween"/>
                <owl:minQualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">2</owl:minQualifiedCar
dinality>
            </owl:Restriction>
        </rdfs:domain>
    </owl:DatatypeProperty>



    <!--
///////////////////////////////////////////////////////////////
//////////////////////////
    //
    // Classes
    //
///////////////////////////////////////////////////////////////
//////////////////////////
     -->




    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#AbsoluteClearness -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#AbsoluteClearness">
        <owl:equivalentClass>
            <owl:Class>
                <owl:oneOf rdf:parseType="Collection">
                    <rdf:Description
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#ØµÙ†Ù^Ø§Ù†"/>
                    <rdf:Description
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#Ù‚Ù†Ù^Ø§Ù†"/>
                    <rdf:Description
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#Ø¨Ù†ÙŠØ§Ù†"/>
```

```
                    <rdf:Description
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#Ø§Ù„Ø¯Ù†ÙŠØ§"/>
                </owl:oneOf>
            </owl:Class>
        </owl:equivalentClass>
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Merging"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#Complete -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#Complete">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#Ghunnah"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#Deepest_Part_of_The_Tongue -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#Deepest_Part_of_The_Tongue">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_tongue"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#Edge_of_The_Tongue -->
```

```
    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#Edge_of_The_Tongue">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_tongue"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#Ghunnah -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#Ghunnah">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Merging"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#Incomplete -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#Incomplete">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#Ghunnah"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#Middle_of_The_Tongue -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#Middle_of_The_Tongue">
```

```
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_tongue"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#Tanween -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#Tanween">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Un_Vowled_Noon_and_Tanween_Rules"/>
        <rdfs:comment rdf:datatype="&xsd;string">The Tanween
is a term for an extra noon saakinah not used
for emphasis, found at the end of nouns when continuing the
reading, but absent from
the noun in the written form (the noon of the tanween is
pronounced but not written),
and abandoned in pronunciation when stopping.
</rdfs:comment>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#The_Articulation_Point_of_the_Letter -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_Articulation_Point_of_the_Letter">
        <rdfs:subClassOf rdf:resource="&owl;Thing"/>
        <rdfs:comment>The Articulation point is the place of
emitting the letter when pronouncing it that differentiates
that specifc letter from other letter.</rdfs:comment>
    </owl:Class>
```

```
    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#The_Change -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_Change">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#UnVowled_Noon"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#The_Clearness -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_Clearness">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#UnVowled_Noon"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#The_Closest_Part_of_the_Throat -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_Closest_Part_of_the_Throat">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Throat"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#The_Deepest__Part_of_the_Throat -->
```

```
    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_Deepest__Part_of_the_Throat">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Throat"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#The_Empty__Space_of_the_Mouth_and_Throat -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-
13#The_Empty__Space_of_the_Mouth_and_Throat">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Articulation_Point_of_the_Letter"/>
        <rdfs:comment rdf:datatype="&xsd;string"> Has in it
one articulation point for the three lengthened letters,
which are wow with no vowel with a dhammah before it, yaâ€™
with no vowel with a kasrah before it, and an alif with a
fath before it.  </rdfs:comment>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#The_Hiding -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_Hiding">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#UnVowled_Noon"/>
    </owl:Class>
```

```
    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#The_Merging -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_Merging">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#UnVowled_Noon"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#The_Middle__Part_of_the_Throat -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_Middle__Part_of_the_Throat">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Throat"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#The_Nose -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_Nose">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Articulation_Point_of_the_Letter"/>
        <rdfs:comment rdf:datatype="&xsd;string">There is one
articulation point, that of the ghunnah</rdfs:comment>
    </owl:Class>
```

```
    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-ontology-13#The_Throat -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_Throat">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Articulation_Point_of_the_Letter"/>
        <rdfs:comment rdf:datatype="&xsd;string"> It has
three articulation points for six different letters which are
pronounced from the deepest, middle, and closest part of the
throat.</rdfs:comment>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-ontology-13#The_Two_Lips -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_Two_Lips">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Articulation_Point_of_the_Letter"/>
        <rdfs:comment rdf:datatype="&xsd;string">The lips
have two articulation points for four letters.
</rdfs:comment>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-ontology-13#The_Un_Vowled_Noon_and_Tanween_Rules -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_Un_Vowled_Noon_and_Tanween_Rules">
        <rdfs:subClassOf rdf:resource="&owl;Thing"/>
    </owl:Class>
```

```
    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#The_tongue -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#The_tongue">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Articulation_Point_of_the_Letter"/>
        <rdfs:comment rdf:datatype="&xsd;string">It has ten
articulation points for eighteen letters.</rdfs:comment>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#Tip_of_The_Tongue -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#Tip_of_The_Tongue">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_tongue"/>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#UnVowled_Noon -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#UnVowled_Noon">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-
13#The_Un_Vowled_Noon_and_Tanween_Rules"/>
        <rdfs:comment rdf:datatype="&xsd;string">Un Vowel
Noon (Noon sakinah) is a noon free from any vowel (ØØ±ÙƒØ©).
```

```
It remains unchanged in its written form and as well as in
pronunciation when continuing to read after it and when
stopping on it.
</rdfs:comment>
    </owl:Class>



    <!--
http://www.semanticweb.org/ymona/ontologies/2017/6/untitled-
ontology-13#without_Ghunnah -->

    <owl:Class
rdf:about="http://www.semanticweb.org/ymona/ontologies/2017/6
/untitled-ontology-13#without_Ghunnah">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ymona/ontologies/201
7/6/untitled-ontology-13#The_Merging"/>
    </owl:Class>
```