Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/13S105_f2

Merging Approach to Support the Incremental Design of Ontology

Eman H. Alkhammash[*]

Department of Computer Science, College of Computers and IT, Taif University, Taif, Saudi Arabia

Received: 4 Dec. 2018, Revised: 23 Jan. 2019, Accepted: 30 Jan. 2019
Published online: 1 Aug. 2019

Abstract: Ontologies are used to formally describe the concepts, properties, and individuals within a domain to be automatically processed by computers. There are large number of ontologies on different domains available on the web. The existence of large number of ontologies imposed the need of merging tools and techniques to construct ontologies by integrating other ontologies. This paper proposes an ontology merging approach based on OntoGraf visualization to integrate the parts of interest of individual ontologies and allows to build ontology incrementally and fixing possible problems and inconsistency of the resulting ontology. The OntoGraf assists in understanding the concepts, properties, and individuals to be merged and assist in finding out possible conflicts before merging. The approach are applied to extend the development of Quran Corpus ontology by merging portion of Quran ontology that focuses on the living creatures and Allah's Names ontology. The major benefit of proposed approach is to strengthen the reuse of ontology knowledge and encourages modularity in the design and development of ontologies.

Keywords: OWL, Ontologies, Merging, Incremental design

1 Introduction

In recent years, research shows an increased interest in the need for the creation of ontologies to support the management of knowledge in different disciplines. Ontology design requires skills to organize the knowledge and categorize the information to be automatically processed by computers. Ontologies formally mean modeling the structure of the system, e.g. creating the related entities and the relations that emerge from the designers view about the system. Those relevant entities are going to be organized into concepts and relationships by an ontology engineer. The essential part of ontology consists of identifying the hierarchy of the concepts e.g. taxonomy, properties, and testing the whole system.

The increased number of existing knowledge sources and ontologies imposed the need of developing approaches to build new ontologies from existing sources and ontologies. Although there are many strategies to develop ontologies, however, there are limitations to the reuse of the ontology [19]. Thus, in this paper we propose an approach that aims to develop and extend OWL ontology incrementally based on the integration existing ontologies. The incremental process of the development of ontologies reduces complexity and allows to fix problems early. The proposed approach aims to integrate small portion of individual ontologies to develop new ontology or extend existing ontology incrementally. We adopt OntoGraf [10] to assist in the ontology design. OntoGraf is a visualization tool in Protégé that become a standard part of Protégé-OWL editor [3]. OntoGraf visualize classes, individuals, and different types of property relations to understand the ontology. The proposed approach suggests to integrate portion of interest of individual ontologies incrementally. To perform the integration, we break out the portion of interest in several features (classes, individuals, properties, axioms). Then we visualize a single feature in the individual ontologies and compare them and select the features to be merged and perform merging. The visualization helps to understand the features to be merged and also helps to understand the ontologies and resolve any possible conflicts. Figure 1 describes the steps of the proposed approach.

Finally we resolve any possible conflicts and evaluate the whole ontology. We apply the approach to extend the ontology of Quran corpus by merging features from Quran ontology that describe living creatures [17] and features from Allah′s names. ontology. We build Allah names ontology and details are found in Section 4. To the best of our knowledge there is no ontology for Allah′s

[*] Corresponding author e-mail: eman.kms@tu.edu.sa

names. Due to the great importance of names of Allah to Muslims, we build ontology based on Allahs names that clarifies and explains the meaning of each name in English and represents several properties related to it. There are several works that adopt ontologies to retrieve knowledge from religious texts such as Quran and Islamic resources. However, most of these ontologies are constructed from scratch and focus on particular topics. We aim to get benefit from merging techniques in developing and enriching Quran ontology.

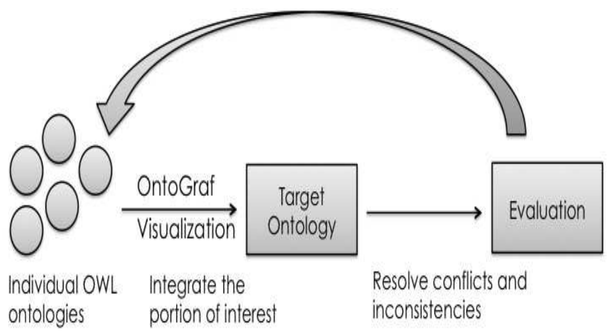

Fig. 1: The phases of the proposed approach

The rest of the paper is structured as follows: Sect 2 gives an overview about Semantic web, Ontology, OWL and OntoGraf. Sect 3 discusses some work of ontology design and integration and Quran ontology. Sect 4 focuses on building Allah's names ontology in OWL. Sect 5 describes the methodology adopted in this paper. Sect 6 describes the application of the proposed ontology to extend Quran corpus ontology. Sect 7 outlines discussion about the proposed approach. The conclusion is presented in Sect 8.

2 Preliminaries

2.1 Semantic Web

With the information age we currently experience, users can get information from different resources and often become overwhelmed with the amount of resources they went through everyday. Thus, many researchers and scientists try to find a viable way to represent knowledge and information and make them more accessible by users. Semantic web is one of the modern fields that emerges recently and offers a great revolution in knowledge representation. Semantic web aims to add semantics and better structure to the information available on the Web. In the semantic web, web resources are formulated in a way that is understandable by humans and by machines [4]. Semantic web converts the current web from being machine-readable to machine-understandable. Ontology is a fundamental technique that is used to provide a common, comprehensible foundation for resources on the semantic web [5].

2.2 Ontology definition

In Artificial Intelligence (AI), ontology is defined as knowledge representation that captures the concepts and the relationships relevant to a specific domain [5]. An ontology necessarily embodies some sort of world view to represent the given domain. The world view is often conceived as a set of concepts (e.g. entities, attributes, process), their definition and their inter-relationships [6]. A good ontology offers a huge number of benefits that are not available in rational database schema or other standard ways to structure information. One of the key advantages of ontology is that it can represent any form of information including unstructured information such as text and documents, semi-structured information like XML or web pages and structured one such as conventional databases′s data. Another benefit is that it allows concept matching, which means that we can match different concepts that have a semantic affinity with a target concept. For example,“$Al - Ghafor$” and “$Al - Ghafaar$” are both referring to the concept of the forgiveness. Ontologies also enable browsing and searching for domain-specific terminology and enable reuse of domain knowledge. Moreover, ontologies allow interoperability among different systems. The quality of an ontology can be ensured using ontology reasoners. An ontology reasoner is used to make the consequences of a certain ontology design explicit, and it thus allows evaluation of the ontology correctness without depending on concrete test cases [7]. The worldwide web consortium (W3C) created (OWL) that became a W3C recommendation in 2004. OWL is an important language in semantic web, which facilitates creating, modifying, linking and importing ontologies in different environments. OWL is derived from description logics that use formal semantics and vocabulary to allow machines to perform automatic reasoning [8,9].

2.3 OntoGraf

OntoGraf is a visualization tool in Protégé framework. OntoGraf enables quick visualization of the ontology or part of it [10]. OntoGraf uses nodes to visualize classes and individuals. The relationships between them are represented as edges. OntoGraf provides several layout to the user and allows the user to focus on particular classes, individuals, or properties through filtering edges and nodes to see the desired one and hide others [10]. Tooltips on OntoGraf show extensive details about other classes related to a particular class or individual such as equivalent classes, disjoint classes, etc.

3 Related work

This section outlines some work regarding ontology design and integration and also describes works in the development of ontologies in Quran and Islamic knoweledge.

3.1 Ontology Design and Integration

This subsection describes some work for designing ontologies from existing sources.

Authors in [11] design an ontology using syntactic matches among concepts in the individual ontolgies. They adopt dictionaries to determine synonymous and evaluate substring, prefix, and suffix. The limitation of [11] is that it is based on syntactic matches (i.e. it relies on grammatical rules and not semantic and the structure of the ontology).

Noy et al. [12] implements a tool called PROMPT. PROMPT uses an algorithm for semi-automatic to perform merging and alignment. It guides the user throughout the process of merging and alignment and it performs some operations automatically. PROMPT detects some errors and conflicts but it is not available in latest versions Protégé.

There is number of work based on converting UML into OWL Ontology. Gasevic et al. [13] proposed an approach that converts UML to OWL. The authors implement eXtensible Stylesheet Language Transformation (XSLT) that transforms the XML Metadata Interchange (XMI) representation of a UML Profile into Web Ontology Language (OWL).

Another study by Tun et al. [20] is based on developing MetaOntoModel as matching method that provides semantic matching method between two sorts by matching most closely corresponding properties instead of comparing all properties. As a limitation of this approach, the classes must be sortal and can enriched using the Meta-OntoModel. Another matching system is proposed by [21]. It aims to provide linked data instance matching on repositories with different schemata. It provides useful attributes pairs for comparing instance without prior knowledge about schemata. The approach applied to dataset contains 246 subsets with different schemata amd resuls show high accuracy compared to other systems.

Another work in the area integration of geospatial data is proposed by Zhang et al. [22]. The authors proposed a prototype that allows user to retrieve geospatial data. The algorithm adopted in the proposed approach uses translation function that saves the translating cost of several geospatial sources from different languages and integrate data via combining complementary properties from the linked records and eliminate data redundancy [22]. The result of the application of the linking method show high performance in generating matched candidates' record pairs.

The approach adopted in the paper focuses on constructing/extending OWL ontologies based on the use of OntoGraf. The OntoGraf describes the features graphically and helps the designer to select the features to be augmented in the target ontology and guides to resolve any possible conflicts.

3.2 Quran Ontologies

There is a number of ontologies that have been developed to represent the Holy Quran and the Islamic knowledge. However, to the best of our knowledge, there is no work related to Allah's names and his attributes. In this subsection we describe some work in the area of ontology development of the Holy Quran and Islamic knoweledge. Saidah and Naomi illustrate an ontology extraction for Islamic knowledge texts based on combining several algorithms together [14]. The authors proposed the design of the ontology using Fuzzy Formal Concept Analysis (FFCA). They firstly extracted the key phrases based on Islamic knowledge to generate meaningful concepts. The proposed ontology is based on general subjects that are found in Quran and Hadith such as organizing financial relations, human and social relations, Al-Jihad, etc. After that, the authors have successfully applied ontological components using previous researches such as Natural Language Ontology (NLO), Islamic Knowledge Gold Standard, Domain Ontology and Ontology Instances. The authors had found that the combination of lexico-syntactic and statistical learning methods would improve the accuracy and the computational efficiency of the ontology discovery process.

Khan et al. studied sample domain ontology based on the living creatures in the Holy Quran has been developed in Protégé editor tool [15]. These creatures include animals and birds. An extensive work has been carried out to capture the knowledge, which involves survey of the Holy Quran to enumerate all the terms that cover animals and birds. The authors discussed some of the main important concepts such as upper ontology, ontology merging, ontology integration and ontology mapping. SPARQL queries of various types that have successfully been run. Saad et al. presented a simple methodology has been proposed to extract concepts based on the Holy Quran [16]. The terminologies or phrases are extracted based on one of the general subjects such as faith, general and political relations, Jihad, etc. The presented approach is stimulated based on the combination of natural language processing techniques, Information Extraction (IE) and Text Mining. From the authors' perspective, the created ontology has a simple format and it can be upgraded using complex NLP techniques. The authors of [17] proposed an ontology that describes named entities in the Holy Quran including concepts such as Person, Prophet, Animal, Angel, Place, Plant,etc. It also used Hadith to populate the ontology and extract additional named entities. The work in [17] applies learning techniques and statistical models to classify ontology instances.

Although, we focus in this paper on developing Quran ontology incrementally using merging approach to enable ontologies to communicate and exchange information. We integrate ontology for Allah names and a previous study extended to extend Quran corpus ontology [17].

4 Development of Names of Allah OWL ontology

In this section, we explain in details of the necessary phases to build Allah's Names ontology. Starting with the data acquisition phase, followed by ontology capture, and then ontology coding.

4.1 Data Acquisition

Data acquisition is defined as a process of gathering and collecting information before processing it using a dedicated computer and processing algorithm. This process can be done using several methods such as surveys, interviews, focus groups and so on. In this phase, we gather information about names and attributes of Allah using reliable resources. Those resources are essentially well-known books trusted by Muslims. For example [18] is a book written by Robert Charles Stade, which includes a translation of the major portion of Al-Ghazali's AlMaqsad Al-Asnabook. As it is known to Muslims, Al-Ghazali is one of the greatest scientists in Islamic culture. In his books, he always attempts to connect minds to logics which makes his books much more understandable and intriguing to learners. Some of the main information we′ve collected includes the meaning of each name, its origin and its root. Also, we collect some properties related to each name such as the specific meaning of each name, the number of times it is mentioned in the Quran, the other names related to it, the opposite names related to it and the names that are similar to it.

4.2 Ontology Capture

After collecting data, we need to formalize them in an organized way. This means that we identify the key concepts and the relationships in the domain and generate accurate definitions for such concepts and relationships. Therefore, in this phase, we firstly produce all potential phrases and terms, which represent concepts and give them clear and unambiguous definitions. For example, $AL-Rahman$ and $AL-Rhaim$ are both derived from Al-Rahma (mercy) but each one of them has a distinct and precise definition. Thus, we look for the accurate definition for each of them. Secondly, we group similar and synonyms terms together and form sub groups. Groups are useful to find relationship between concepts that are related together and share some properties. For instance, $AL-Khaliq$ and $AL-Bari$ have different roots but they have one specific meaning which is "The creator". Finally, we capture the relationships between terms and concepts. In real word, every entity is related to other one. Thus the relationships between entities should lead us to a comprehensive knowledge and enable deeper understanding of entities. After gathering the necessary information from authoritative books and defining the main concepts and their definitions, we suggest several relationships between the concepts until we′ve reached our appropriate view about the domain. We have four main concepts, which are $NamesOfAllah$, $Meaning$, $Source$, and $Root$. Each name has its origin (i.e. the source of the name: *Quran* or *Sunnah*). The *Meaning* class provides all meanings related to each name, they can be a general meaning that can refer to several names and also a particular meaning that are special to each name. The *Root* class represents the primary lexical unit of a word.

4.3 Coding

Coding means representing the concepts and the relationships captured from the previous phase and convert them into a shape that computer can understand using a programming language. We use Protégé platform to create classes, sub-classes, object′s properties, data properties, and individuals related to the domain. The class hierarchy of the developed ontology is displayed in Figure 2.

Fig. 2: The class hierarchy of the proposed ontology

Classes describe the domain concepts. A class can have subclasses that represent concepts that are more specific than the super class. The names of Allah are individuals belonging to *Names Of Allah* class. *Meaning* is a class that includes individuals that describe the general meaning of Allahs names such as *Creator, King, etc*. *Source* is a class, which includes two individuals: *Quran* and *Sunnah*, where each name of *Allah* is originated from *Quran* or *Sunnah* or both. Finally the *Root* class describes the primary lexical unit of each name; for instance the root of *Al-Bari* name is *Bara*. In Protégé, an ontology can be stored in various formats, e.g. in RDF(S), XML, OWL. Figure 3 describes some of OWL code that describes *AL-Rahman* and some properties related to it. It shows that $Al-Rahman$ is individual, which has a root "Rahem and general meaning "mercy and specific meaning"general mercy for all of creation. It also shows that $Al-Rahman$ is mentioned 169 times in the *Quran* and also mentioned in *Sunnah*. Furthermore, $Al-Rahman$ has synonym name which is $Al-Rahem$.

```
<ObjectPropertyAssertion>
<ObjectProperty IRI="#Synonyms"/>
<NamedIndividual IRI="#Al-Rahman"/>
<NamedIndividual IRI="#AlRahim"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#hasSource"/>
<NamedIndividual IRI="#Al-Rahman"/>
<NamedIndividual IRI="#Quran"/></ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#hasSource"/>
<NamedIndividual IRI="#Al-Rahman"/>
<NamedIndividual IRI="#Sunnah"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#isRootOf"/>
<NamedIndividual IRI="#Rahm"/>
<NamedIndividual IRI="#Al-Rahman"/>
</ObjectPropertyAssertion>
<DataPropertyAssertion>
<DataProperty IRI="#SpecificMeaning"/>
<NamedIndividual IRI="#Al-Rahman"/><Literal datatypeIRI="&xsd;string">General mercy for all
creation</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
<DataProperty IRI="#TimesMentionedInQuran"/><NamedIndividual IRI="#Al-Rahman"/><Literal
datatypeIRI="&xsd;integer">169</Literal>
</DataPropertyAssertion>
```

Fig. 3: Example of OWL code that describe Al_Rahman name

5 Methodology

Merging creates single coherent ontology from different resource (ontologies). One problem that often occurs in full merging of ontologies is representing the same information and introducing multiple inheritance for some concepts. Such problems may be avoided by asymmetric merge approaches [2] that preserve only the concepts and relationships of one input and integrate only non-redundant concepts of the other ontology. In this paper we propose an approach that focuses on asymmetric merge to preserve only the concepts and relationships of one input and integrate only non-redundant concepts of target ontology. The idea is to use OntoGraf visualization to visualize the classes, properties, individuals of the input ontologies so they can be merged incrementally in the resulting ontology. OntoGraf gives details about each class and other classes related to it. These details help to draw the designer attention to the desired classes, properties, and individuals that need to be merged to the target ontology and not get bothered with unrelated other classes, properties or individuals. This also helps to grasp possible conflicts and inconsistencies and guides the designer to the changes that needs to be done in the target ontology. The approach is incremental and helps the designers to focus and integrating small portion of ontology every time.

The steps needed to achieve consistent and coherent merged ontology are as follows:

– Break out the features of the individual ontologies into manageable features.
– Use OntoGraf to compare the parts of interest in the individual ontologies and identify inconsistencies and potential problems.
– Copy classes of interest that are different in the individual ontologies to the merged in the target ontology
– Merge classes that are semantically same and have same name
– Rename classes that have different names and semantically same in the merged ontology
– Copy the properties of interest that are different in the individual ontologies to the merged ontology
– Merge properties that are similar semantically and syntactically in individual ontologies in the merged ontology
– Fix possible inconsistencies in the merged (target) ontology

We′ve mentioned earlier that OntoGraf can help to investigate possible conflicts and inconsistency. In this section, we outline some examples that describe some possible conflicts. One source of difficulty that often occurs in the merged ontology is making an incorrect class hierarchy by classifying a concept as a subclass of a concept that is not really belong to. This can happen in case of property subsumption violation or constraint dissatisfaction [1]. Investigating this type of semantic inconsistencies during merging is often a difficult task. We believe that the use of OntoGraf can better help to identify this type of error visually before merging and identify classes, properties, and individuals of interest in the individual ontologies. Thus, the designer can check only those axioms and suggest the correct hierarchy of the merged ontology. The major reasons the lead to inconsistency exist in Ref. [1], Firstly, the concept with 'weaker domain' is placed under a concept with 'stronger domain' in the merged ontology. domain. Secondly, a subclass in the merged ontology could have some features that violate features in the super class or partially meet the properties of super class and violate the others. Thirdly, merging ontology has placed a concept as a subclass of a concept that occupies a disjoint domain.

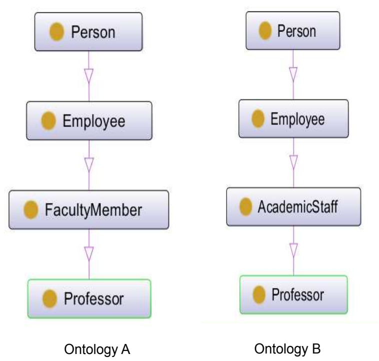

Fig. 4: Concepts that have different names in individual ontologies

Figure 4 shows a case where there are two parents one in ontology *A* which is *AcademicStaff* and the second *Faculty Member* in ontology *B*. It is clear from the diagrams that merging classes cause circle after merging as shown in Figure 5 . Therefore, we need to remove one of the parents because it makes circle.

This can be achieved by renaming the class *Faculty Member* to *Academic Staff* after the merge. The effect of this is that all properties and axioms that are related to *Faculty Member* are updated accordingly with *Academic Staff* in the merged ontology. The properties and axioms need to be checked for any possible inconsistency before merging.

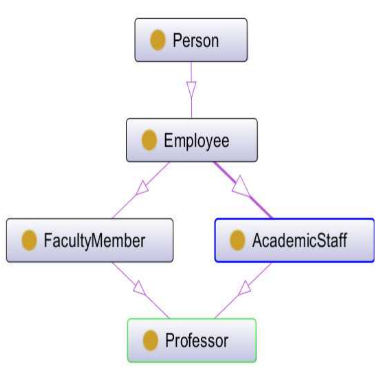

Fig. 5: Result of merging concepts in Figure 4

If the designer wants to copy some properties from individual ontology then OntoGraf can be used to guide in deciding which properties need to be copied. Figure 6 shows five arrows between class *University* and *Person*. The designer can click on each arrow and check the properties that can be of interest.

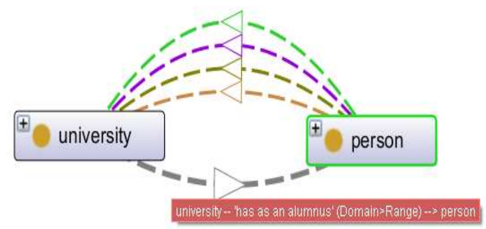

Fig. 6: Object properties between *University* and *Person*

OntoGraf can also help to avoid redundancy in class hierarchies after merging and allows the designer to organize the hierarchy in the merged ontology and suggests the possible solutions to avoid redundancy. For instance, Figure 7 shows redundancy between concepts in the hierarchy of Autombile in ontology *A* and *B*.

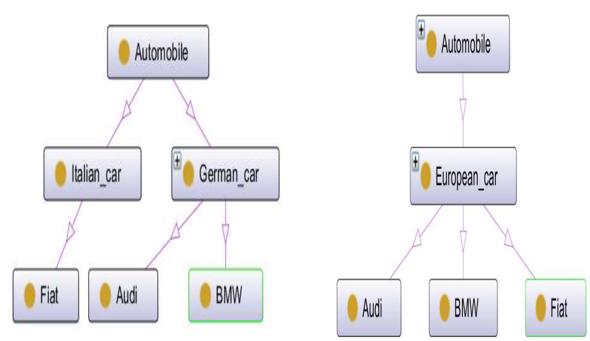

Fig. 7: Two different hierarchy of automobile in ontology A and B

The visualization of OntoGraf helps the designer to suggest the desired hierarchy in the merged ontology as shown in Figure 8. The designer selects to omit two classes *Italian_car* and *German_car* and make the several classes that represent cars as subclasses of *European_car*.

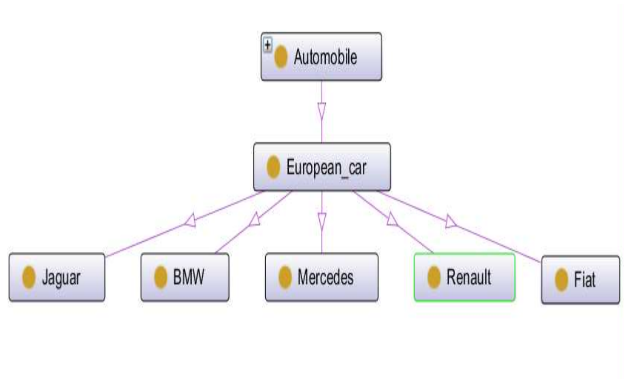

Fig. 8: The hierarchy of automobile in the merged ontology

Moreover, OntoGraf helps the designer to check if a subclass concept should not belong to disjoint domain or weaken or breach the properties and features of any superclass concepts.

Figure 9 shows three concepts, a superclass *PizzaTopping* and subclasses *VegetableTopping* and *MeatTopping*. *MeatTopping* is disjoint with other classes such as *VegetableTopping*, *CheeseTopping*, *etc*.

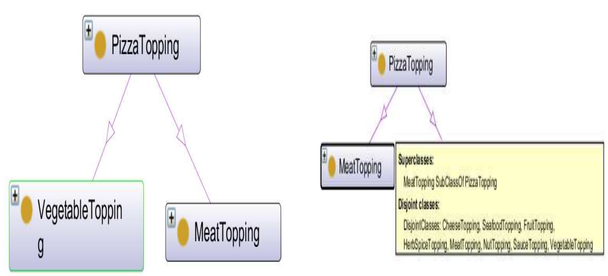

Fig. 9: *MeatTopping* superclass and disjoint classes

Figure 10 describes *MeatyVegetableTopping* in another ontology and shows that it belongs to two overlapped classes which are *MeatTopping* and *VegetableTopping*.

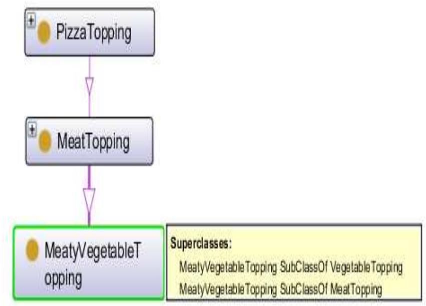

Fig. 10: *MeatyVegetableTopping* superclasses

If the designer wants to merge *MeatyVegetableTopping* in a target ontology. Then *MeatyVegetableTopping* requires to be a subclass of *MeatTopping* and *VegetableTopping* and both classes should be overlapped and not disjoint classes order to get successful merge as shown in Figure 11.

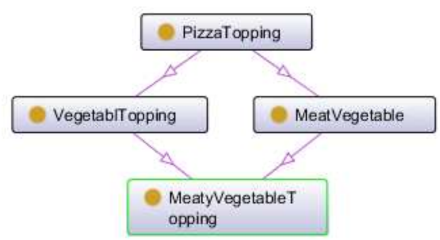

Fig. 11: The result of merging MeatyVegetableTopping

6 The application of the proposed ontology to extend Quran corpus ontology

In this section we extend Quran corpus ontology presented in [17] by merging some classes, properties, and individuals from the Name's of Allah ontology in section 4 and Quran ontology of living creatures in [15]. We consider Quran Corpus ontology as the target ontology and we extend the classes, properties, and individuals by classes, properties, and individuals existing in Quran ontology of living creatures and Name's of Allah ontology. We break out the individual ontologies into *Bird* hierarchy, *People* hierarchy, *Prophet* hierarchy, and *NamesOfAllah* hierarchy.

Figure 12 shows *Bird* concepts and properties related to it on Quran ontology of living creatures whereas *Bird* class in Quran corpus ontology is shown in Figure 13.

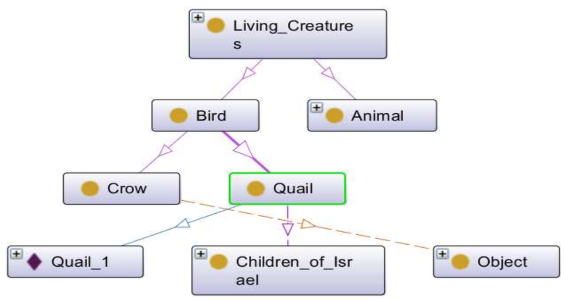

Fig. 12: Bird hierarchy in Quran ontology of living creature

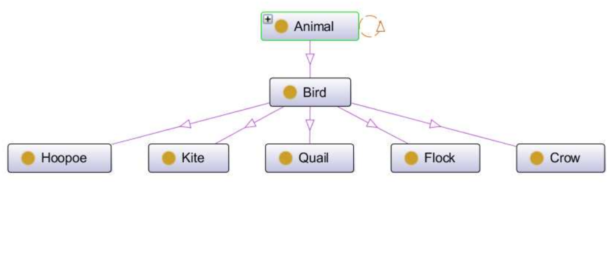

Fig. 13: Bird hierarchy in Quran corpus ontology

In order to merge *Bird* class we organize the hierarchy by adding *Living_Creature* concept as a super class of *Animal* and make *Bird* superclass of *Bird* class and then merge class *Bird*, *Quail*, and *Crow* with *Bird*, *Quail*, *Crow* in the Quran corpus ontology. The result is shown in Figure 14 .

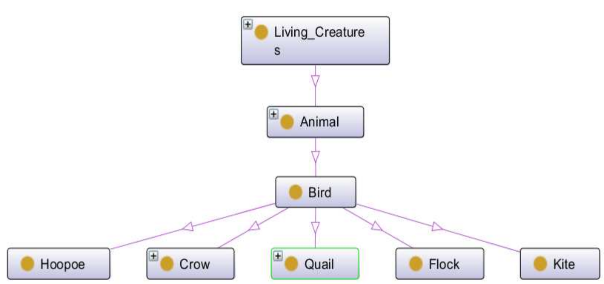

Fig. 14: Merging concepts from Figure 12 and Figure 13

Another example of merging is to merge some concepts about people mentioned in Quran. The hierarchy of people mentioned in Quran corpus ontology as shown in Figure 15 is different from the hierarchy of people mentioned in Quran ontology of living creatures as shown in Figure 16.

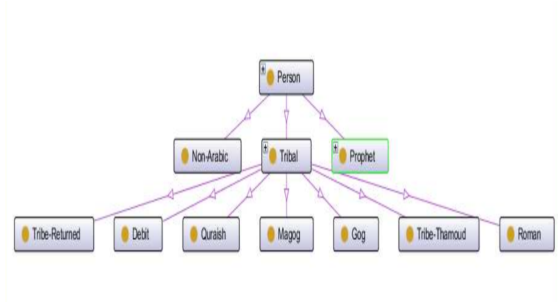

Fig. 15: The hierarchy of people mentioned in Quran corpus ontology

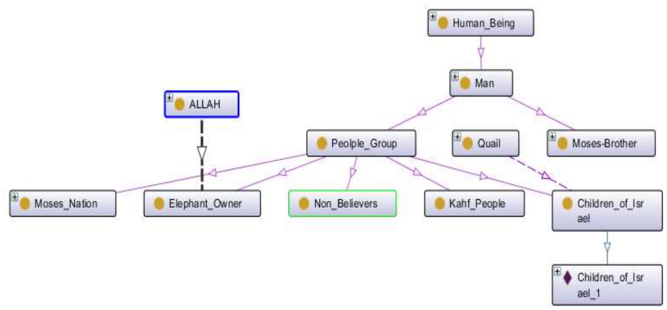

Fig. 16: The hierarchy of people mentioned in Quran ontology of living creatures

The result of the merge is shown in Figure 17. We remove *Tribal* class and make all sub classes of *Tribal* as subclasses of *People_Group*.

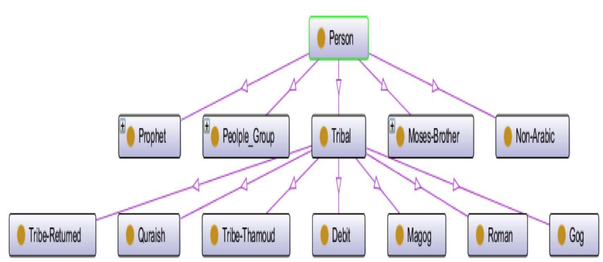

Fig. 17: Merging concepts from Figure 15 and Figure16

In the context of people mentioned in *Quran*, we also would like to focus on *Prophet* class that mentioned in Figure 18 and Figure 19 and perform merging of individuals.

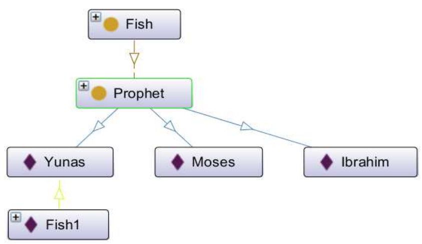

Fig. 18: Prophet concept in Quran corpus ontology

The result of merging the individuals from Figure 18 and 19 is shown in Figure 20.

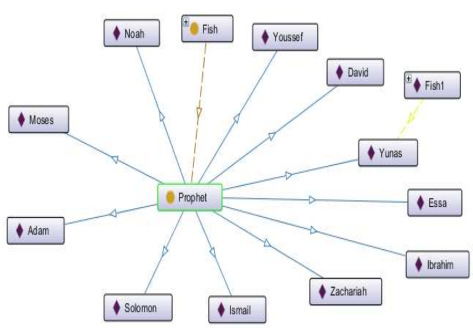

Fig. 19: Prophet concept in Quran ontology of living creatures.

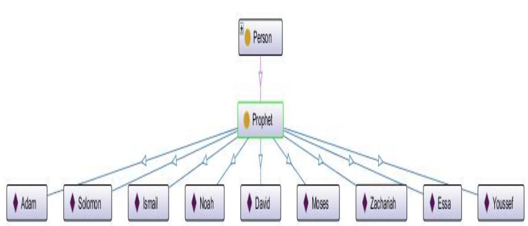

Fig. 20: The result of merging individuals in *Prophet* concepts

Allah's names ontology describes several names of Allah. Figure 21 shows some names which are represented as individuals.

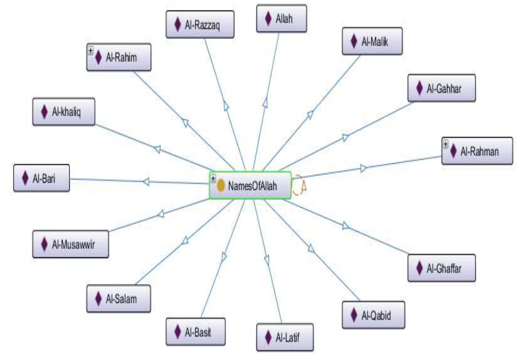

Fig. 21: Some individuals in NamesOfAllah concept.

The hierarchy of Allah's Names are described in Figure 22.

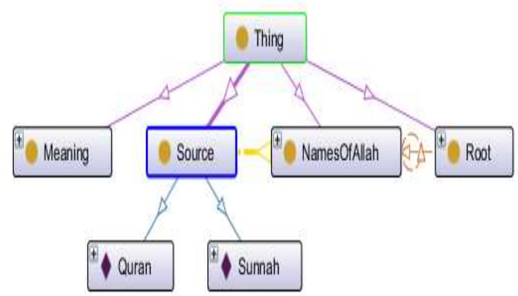

Fig. 22: The hierarchy of *NamesOfAllah*

As can be seen in Figure 22, *Source* class has two individuals *Quran* and *Sunnah*. The object property *hasSource* links between *Source* and *NamesOfAllah*. On the other hand Quran corpus ontology describes *HadithBook* and *QuranBook* as concepts.

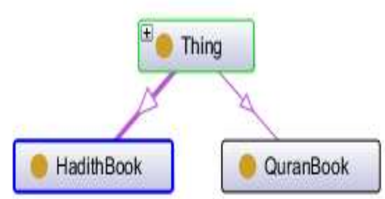

Fig. 23: *QuranBook* and *HadithBook* concepts in Quran corpus ontology.

We can merge Allah's Names ontology to Quran Corpus ontology by copying all concepts and make *HadithBook* and *QuranBook* concepts as subclasses of *Source* concept as shown in Figure 24.

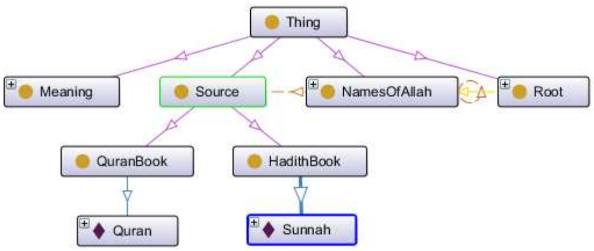

Fig. 24: Result of merging Allah's names ontology and Quran Corpus ontology.

HadithBook class in Quran corpus ontology has *Hadith* as individual which is semantically similar to *Sunnah* in Quran Ontology. In the merged ontology we merge the two individuals after renaming both as *Sunnah*.

Now we need to focus on the object properties of Allah's names such as *Synonyms*, *hasSource*, and data properties *TimeMentionedInQuran* and *SpecificMeaning* that are shown in Figure 25.

Fig. 25: Representation of object properties and data properties of Allah's names.

There are quite similar object properties in Quran corpus ontology but syntactically different such as *hasSynonym*, *hasSource*, and *means* as shown in Figure 26. In fact *means* is considered as object property in Allah's names but is is considered as data property in Quran corpus ontology. Therefore we decide to consider it as object property in the merged ontology. We need also to fix the domain and range of the property. Since *Animals* and *NamesOfAllah* are disjoint concepts. Then we can select the domain as *Thing* and the range as *Thing*. Also for other object properties like *hasSource* and *hasSynonym* we need to fix the domain and the range.

Table 1: Object properties merged in the resulting ontology

Object property	Domain	Range
Sent_down_to	Quail	Children_of_Israel
Scratch	Crow	Object
Deal_with	NamesOfAllah	Elephant_Owner
Turn_into	NamesOfAllah	Moses_Nation
worship	Moses_Nation	Calf_Moses
Not_forbade_to_eat	NamesOfAllah	Swine
forbade_to_eat	NamesOfAllah	Swine
hasOwner	Animal	Person
Give_Example	NamesOfAllah	Gnat
Swallow	Fish	Prophet
hasSource	Thing	Source
hasOwner	Thing	Person

Table 2: Data properties merged in the resulting ontology

Data property	Domain	Range
hasColor	Calf_Moses	Yellow
hasSynonym	Thing	String
means	Thing	String
Is_brought_to	Calf_Ibrahim	Angle

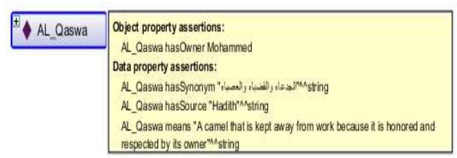

Fig. 26: Representation of object properties and data properties of individual in animal class.

The object properties and data properties that are merged in the resulting ontology are shown on Table 1 and 2.

We can merge Allah's Names ontology to Quran Corpus ontology by copying all concepts and make *HadithBook* and *QuranBook* concepts as subclasses of *Source* concept as shown in Figure 24.

7 Discussion

The approach adopted in this paper facilitates the development of ontologies through the reuse external sources. It allows the designer to use OntoGraf diagrams to understand different features of ontologies in stepwise manner and focus on fixing errors and inconsistencies during the development. The development of each increment is simpler than developing and integration of the whole ontology and would enable a continuous assessment of the ontology status.

Each increment is evaluated first using OntoGraf visualization, and after that it can be integrated with the target ontology with the amendments required to resolve problems and inconsistencies. At the end evaluate the resulting ontology.

The use of graphical notation in OntoGraf facilitates the understanding and representation of ontology designs.

Currently, we have extended Quran Corpus ontology by merging different portion from two ontologies: Quran ontology of living creatures and Allah's names ontology.

However, possible limitations may include the need for further checking to the domain and range of data properties and the type of object properties (i.e. functional, inverse, symmetric, reflexive, etc) since these details do not appear in OntoGraf diagram. However, OntoGraf can draw the focus of the designer to the desired properties that need to be checked and not get bothered with other properties.

8 Conclusions

Merging techniques are crucial for modelling complex domain. This paper presents an approach for incremental development of ontology using graphical merging approach. OntoGraf allows for understanding and inspection of possible errors and conflicts in concepts, properties, and individual of existing ontologies before integration. We have applied the proposed approach to extend Quran Corpus ontology by integrating Quran ontology and Allah's names ontology. Firstly, we break out the portion of interest in the input ontologies. Then, we use OntoGraf to visualize the portion of interest in the individual ontologies. Then we check the hierarchy, properties, and individuals that are represented in OntoGraf to discover any possible conflicts or problem. Then we integrate these portions and fix problems and finally, we evaluate the whole ontology. We believe that this approach can be useful to understand the ontology and develop ontology incrementally based on reuse.

We plan to apply this approach to design large volume of ontologies and investigating graphical patterns that aid the designer through decision-making during the development of ontology.

References

[1] Fahad, M. and Moalla, N. and Bouras, A., "Detection and resolution of semantic inconsistency and redundancy in an automatic ontology merging system," Journal of Intelligent Information Systems, Bangkok, 39(2), 2012.

[2] Mahfoudh, M.and Forestier, G. and Hassenforder, M., "benchmark for ontologies merging assessment.," In International Conference on Knowledge Science, Engineering and Management KSEM 2016, Passau, Germany, October 5-7, 2016, Proceedings, 39(2), 2016, pp. 555-566.

[3] Knublauch, H. and Fergerson, R.W. and Noy, N.F. and Musen, M.A., "The Protg OWL plugin: An open development environment for semantic web applications," Springer, 39(2), 2004, pp. 229–243.

[4] Taye, M.M., "Understanding semantic web and ontologies: Theory and applications," arXiv preprint arXiv:1006.4567, 2010.

[5] Guarino, N. and Oberle, D. and Staab, S., "What is an Ontology?," Springer International Publishing, Handbook on ontologies, Springer, 2009, pp. 1-17.

[6] Uschold, M. and Gruninger, M., "Ontologies: Principles, methods and applications," KNOWLEDGE ENGINEERING REVIEW, 11, 1996, pp. 93-136.

[7] Sirin, E. and Parsia, B. and Grau, B.C. and Kalyanpur, A. and Katz, Y., "Pellet: A Practical OWL-DL Reasoner," Web Semant., 5(2), 2007, pp. 51-53.

[8] Heflin, J. and Words, K., "AN INTRODUCTION TO THE OWL WEB ONTOLOGY LANGUAGE," 2007.

[9] Horridge, M., "A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3," The University Of Manchester, 2011.

[10] Falconer, S., "OntoGraf," http://protegewiki.stanford.edu/wiki/OntoGraf, 2010.

[11] Chalupsky , H. and Hovy, E. and Russ, T. "Progress on an Automatic Ontology Alignment Methodology," 1997.

[12] Noy, N.F. and Musen, M.A., "Algorithm and tool for automated ontology merging and alignment," Web Semant., 5(2), 2000.

[13] Gasevic, D. and Djuric, D. and Devedzic, V. and Damjanovi, V., "Converting UML to OWL ontologies," Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, ACM, 2004, pp. 488-489.

[14] Saad, S. and Salim, N. and Zainal, H., "Islamic knowledge ontology creation," Proceedings of the 4th International Conference for Internet Technology and Secured Transactions, ICITST 2009, London, UK.

[15] Khan, H.U. and Saqlain, S.M. and Shoaib, M. and Sher, M., "Ontology based semantic search in Holy Quran," International Journal of Future Computer and Communication, 2(6), 2013, pp. 570.

[16] Saad, S. and Salim, N., "Methodology of Ontology Extraction for Islamic Knowledge Text," Postgraduate Annual Research Seminar, 2008.

[17] Alkhmmash, E. and Ben Abdessalem, W. "A Holy Quran Ontology Construction with Semi-automatic population," The international Conference on Cummunication, Management and Information Technology (ICCMIT 2017), University of Warsaw, Poland, 2017.

[18] Stade, R.C., and PRESS, D., "Ninety-nine names of God in Islam," The international Conference on Cummunication," Management and Information Technology (ICCMIT 2017), University of Warsaw, Poland, 1970.

[19] Cristani, M. and Cuel, R. "A Survey on Ontology Creation Methodologies," Semantic Web-Based Information Systems: State-of-the-Art Applications. IGI Global, 2007, pp. 98-122.

[20] Tun, N.N. and Tojo, S., "Semantic enrichment in ontologies for matching," International Journal on Semantic Web and Information Systems (IJSWIS), IGI Global, 2(4), 2006, pp. 33–67.

[21] Nguyen, K. and Ichise, R., "Automatic schema-independent linked data instance matching system," Information Retrieval and Management: Concepts, Methodologies, Tools, and Applications, IGI Global, 2018, pp. 1446–1469.

[22] Zhang, Y., and Li, C. and Chen, N. and Liu, S. and Du, L. and Wang, Z. and Ma, M., "Semantic web and geospatial unique features based geospatial data integration,"

International Journal on Semantic Web and Information Systems (IJSWIS), IGI Global, 12(1), 2016, pp. 1–22.
[23] Horridge, M., “OWLViz,” http://protegewiki.stanford.edu/wiki/OWLViz, 2010.
[24] Harith, A., “TGVizTab: An ontology visualisation extension for Protégé,” Knowledge Capture (K-Cap’03), Workshop on Visualization Information in Knowledge Engineering, 2003, http://oro.open.ac.uk/20054/.
[25] Sintek, M., “OntoViz tab: Visualizing Protégé Ontologies,” 2003.
[26] Castaneda, V. and Ballejos, L. and Caliusco, L. and Galli, M. R., “Knowledge Capture (K-Cap’03), The Use of Ontologies in Requirements Engineering,” Global Journal of Researches in Engineering specification, 10(6), 2010, pp. 2-8.
[27] Avdeenko, T. and Pustovalova, N., “The ontology-based approach to support the completeness and consistency of the requirements specification,” Control and Communications (SIBCON), 2015 International Siberian Conference on The ontology-based approach to support the completeness and consistency of the requirements specification, 2015, pp. 1-4, 10.1109/SIBCON.2015.7147184.
[28] Thayer, R.H. and Bailin, S.C. and Dorfman, M., “Software Requirements Engineerings, 2Nd Edition,” IEEE Computer Society Press, isbn:0818677384, 2nd, Los Alamitos, CA, USA, 1997.
[29] Belgueliel, Y. and Bourahla, M. and Brik, M., “Towards an Ontology for UML State Machines,” Lecture Notes on Software Engineering, 2(1), 2014, pp. 116–120.
[30] Kroha, P. and Janetzko, R. and Labra, J. E., “Ontologies in Checking for Inconsistency of Requirements Specification,” Proceedings of the 2009 Third International Conference on Advances in Semantic Processing, SEMAPRO ’09, isbn:978-0-7695-3833-4, IEEE Computer Society, Washington, DC, USA, 2009, pp. 32–37, http://dx.doi.org/10.1109/SEMAPRO.2009.11.
[31] Dermeval, D. and Vilela, J. and Bittencourt, I.I. and Castro, J. and Isotani, S. and Brito, P. and Silva A., “A Systematic Review on the Use of Ontologies in Requirements Engineering,” Software Engineering (SBES), 2014 Brazilian Symposium on A Systematic Review on the Use of Ontologies in Requirements Engineering, 2014, pp. 1–10, doi:10.1109/SBES.2014.13.
[32] Fuchs, N.E. and Schwertel, U., “Reasoning in Attempto Controlled English,” Principles and Practice of Semantic Web Reasoning, International Workshop PPSWR 2003, isbn:978-3-540-20582-1, 2003, pp. 174-188, doi:10.1007/b94439.
[33] Horridge, M. and Knublauch, H. and Rector, A. and Stevens, R. and Wroe, C., “A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools Edition 1.0,” The University Of Manchester, 2004.

Eman Alkhammash works as assistant professor of Computer Science at the University of Taif in Saudi Arabia. She holds MSc and PhD (Computer Science) from University of Southampton. Eman does research in Formal Methods and Software Engineering. She was awarded a Fellow of the Higher Education Academy (FHEA) in September 2018.