WEB LINKAGE VIA LEARNING INHERENT SEMANTICS AND EVOLUTION OF ONTOLOGIES

A dissertation submitted
to Kent State University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

by
Abdulelah Algosaibi
December 2015
Dissertation written by
Abdulelah Algosaibi
B.S., King Faisal University, Saudi Arabia, 2008
M.S., Kent State University, USA, 2011
Ph.D., Kent State University, USA, 2015

Approved by
Austin Melton, Professor, Ph.D., Computer Science, Doctoral Advisor
Johnnie Baker, Professor Emeritus, Ph.D., Computer Science
Angela Guercio, Associate Professor, Ph.D., Computer Science
Alan Brandyberry, Associate Professor, Ph.D., Management and Information Systems
Sarah Rilling, Associate Professor, Ph.D., English

Accepted by
Javed I. Khan, Professor, Ph.D., Chair, Department of Computer Science
James L. Blank, Professor, Ph.D., Dean, College of Arts and Sciences
TABLE OF CONTENTS

LIST OF FIGURES........................................................................................................ VII
LIST OF TABLES............................................................................................................... X
ACKNOWLEDGEMENTS .................................................................................................. XI

CHAPTER 1 SEMANTIC WEB INTRODUCTION ................................................................. 1
  1.1 WEB 1.0 .................................................................................................................. 1
  1.2 WEB 2.0 .................................................................................................................. 2
  1.3 WEB 3.0 .................................................................................................................. 4
    1.3.1 eXtensible Markup Language (XML) ................................................................. 7
    1.3.2 Resource Description Framework (RDF) ......................................................... 7
    1.3.3 RDF schema (RDF-S) ..................................................................................... 8
    1.3.4 Ontology .......................................................................................................... 9
    1.3.5 Proof and Trust ............................................................................................... 11
  1.4 WEB GENERATIONS’ EVOLUTION ..................................................................... 11
  1.5 GOALS AND CONTRIBUTIONS ......................................................................... 12
  1.6 ORGANIZATION OF THE DISSERTATION ......................................................... 14
  1.7 SUMMARY ............................................................................................................ 16

CHAPTER 2 BACKGROUND AND RELATED WORK ................................................. 17
  2.1 INTRODUCTION TO ONTOLOGY LEARNING .................................................... 17
    2.1.1 Type of Documents ....................................................................................... 18
    2.1.2 Unstructured Documents .............................................................................. 18
CHAPTER 4 USING THE SEMANTICS INHERENT IN SITEMAPS TO LEARN ONTOLOGIES

4.1 INTRODUCTION ................................................................................. 71

4.2 METHODOLOGY .............................................................................. 76

4.2.1 Taxonomic Phase ........................................................................ 77

4.2.1.1 INPUT PREPROCESSING .............................................................. 78

4.2.1.2 SEMANTIC SIMILARITY SYSTEM .............................................. 96

4.2.2 Enhancement Phase ..................................................................... 97

4.3 EXPERIMENT AND EVALUATION ............................................... 103

4.4 SUMMARY ....................................................................................... 112

CHAPTER 5 THREE DIMENSIONS ONTOLOGY MODIFICATION MATRIX ........113
5.1 BACKGROUND AND RELATED WORK ................................................................. 114

5.2 THREE DIMENSIONS ONTOLOGY MODIFICATION MATRIX ............................ 120

5.2.1 Three Dimensions Ontology Modification Matrix Empirical Study.................. 127

5.3 ADVANTAGES OF THREE DIMENSIONS ONTOLOGY MODIFICATION MATRIX .... 134

5.4 SUMMARY ........................................................................................................... 135

CHAPTER 6 CONCLUSION AND FUTURE WORK .................................................. 136

APPENDIX A ............................................................................................................. 141

REFERENCES .......................................................................................................... 178
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Web generation time line</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>Semantic Web stack layer</td>
<td>6</td>
</tr>
<tr>
<td>1.3</td>
<td>Visualization of RDFS</td>
<td>8</td>
</tr>
<tr>
<td>2.1</td>
<td>Ontology learning steps by [12]</td>
<td>23</td>
</tr>
<tr>
<td>2.2</td>
<td>Main steps in ontology creation by [13]</td>
<td>25</td>
</tr>
<tr>
<td>2.3</td>
<td>Ontology development stages by [19]</td>
<td>26</td>
</tr>
<tr>
<td>2.4</td>
<td>Extracting instances segments</td>
<td>29</td>
</tr>
<tr>
<td>2.5</td>
<td>Studing of the HTML tags</td>
<td>30</td>
</tr>
<tr>
<td>2.6</td>
<td>DOM tree of New York Times front page snapshot from [29]</td>
<td>38</td>
</tr>
<tr>
<td>2.7</td>
<td>A snapshot from [30] to show query interface and the result of a query</td>
<td>40</td>
</tr>
<tr>
<td>3.1</td>
<td>HTML output page</td>
<td>53</td>
</tr>
<tr>
<td>3.2</td>
<td>HTML source page</td>
<td>53</td>
</tr>
<tr>
<td>3.3</td>
<td>Number of results for single search</td>
<td>56</td>
</tr>
<tr>
<td>3.4</td>
<td>Web directory shows category and subcategory for some domains</td>
<td>56</td>
</tr>
<tr>
<td>3.5</td>
<td>Google referring the user to HTML sitemap</td>
<td>58</td>
</tr>
<tr>
<td>3.6</td>
<td>XML sitemap example</td>
<td>60</td>
</tr>
<tr>
<td>3.7</td>
<td>Example of HTML sitemap captured from (<a href="http://www.tigerdirect.com">www.tigerdirect.com</a>)</td>
<td>61</td>
</tr>
<tr>
<td>4.1</td>
<td>Example of a HTML sitemap</td>
<td>73</td>
</tr>
<tr>
<td>4.2</td>
<td>Revised HTML sitemap from website#1</td>
<td>74</td>
</tr>
<tr>
<td>4.3</td>
<td>Revised HTML sitemap from website#2</td>
<td>75</td>
</tr>
<tr>
<td>Figure 4.4: Revised HTML sitemap from website#3</td>
<td>.......................................................... 75</td>
<td></td>
</tr>
<tr>
<td>Figure 4.5: The approach is divided into two phases</td>
<td>.......................................................... 77</td>
<td></td>
</tr>
<tr>
<td>Figure 4.6: Proposed system interface</td>
<td>.......................................................... 79</td>
<td></td>
</tr>
<tr>
<td>Figure 4.7: Output of advanced mode</td>
<td>......................................................................... 80</td>
<td></td>
</tr>
<tr>
<td>Figure 4.8: Example to calculate Seu and SiU</td>
<td>.......................................................... 82</td>
<td></td>
</tr>
<tr>
<td>Figure 4.9: Segmentation of HTML sitemap from its root</td>
<td>.......................................................... 85</td>
<td></td>
</tr>
<tr>
<td>Figure 4.10: Match proposed-concepts of different generations</td>
<td>.......................................................... 89</td>
<td></td>
</tr>
<tr>
<td>Figure 4.11: Visualize an ontology use the Protégé API</td>
<td>.......................................................... 101</td>
<td></td>
</tr>
<tr>
<td>Figure 4.12: Tracing proposed relationships using Protégé API</td>
<td>.......................................................... 102</td>
<td></td>
</tr>
<tr>
<td>Figure 4.13: Editing proposed relationships using Protégé API</td>
<td>.......................................................... 103</td>
<td></td>
</tr>
<tr>
<td>Figure 4.14: Sample of output ontology at apparel domain</td>
<td>.......................................................... 105</td>
<td></td>
</tr>
<tr>
<td>Figure 4.15: Sample of auto parts domain ontology result</td>
<td>.......................................................... 106</td>
<td></td>
</tr>
<tr>
<td>Figure 4.16: Threshold and average precision relationship</td>
<td>.......................................................... 108</td>
<td></td>
</tr>
<tr>
<td>Figure 4.17: Ontology before using Fact++ reasoned</td>
<td>.......................................................... 109</td>
<td></td>
</tr>
<tr>
<td>Figure 4.18: Ontology after using Fact++ reasoned</td>
<td>.......................................................... 109</td>
<td></td>
</tr>
<tr>
<td>Figure 4.19: Visualization of ontology via OntoGraf from Protégé API for architectural services domain</td>
<td>......................................................................... 110</td>
<td></td>
</tr>
<tr>
<td>Figure 4.20: Visualization of ontology via OntoGraf from Protégé API for auto parts domain</td>
<td>......................................................................... 111</td>
<td></td>
</tr>
<tr>
<td>Figure 5.1: Three dimensions ontology modification matrix</td>
<td>.......................................................... 120</td>
<td></td>
</tr>
<tr>
<td>Figure 5.2: (d) distance</td>
<td>......................................................................... 122</td>
<td></td>
</tr>
<tr>
<td>Figure 5.3: Simple ontology from vehicles domain ontology version 1.0</td>
<td>................................. 122</td>
<td></td>
</tr>
</tbody>
</table>
Figure 5.4: Ontology from vehicles domain with more classes and subclasses
relationships and properties relationships (ontology version 3.0).............................. 123

Figure 5.5: Ontology from vehicles domain with little more classes and subclasses
relationships (ontology version 2.0)............................................................................ 123

Figure 5.6: Evolution trend.......................................................................................... 124

Figure 5.7: Efforts associated with modifying given class............................................. 125

Figure 5.8: University ontology versions 1.0................................................................. 132

Figure 5.9: University ontology versions 2.0................................................................. 133

Figure 6.1: Web generations....................................................................................... 139
LIST OF TABLES

Table 1.1: Main characteristics of each Web generations ............................................. 12
Table 2.1: Auto information table.................................................................................. 19
Table 4.1: Adjacency list representation of figure 4.1...................................................... 84
Table 4.2: Example of processed HTML sitemaps at newSitemapsID database ............ 90
Table 4.3: Example of newSitemapsID database at the time of processing ............... 91
Table 4.4: Representation of the array at the time of the processing............................. 92
Table 4.5: Representation of the array after processing matching segment function...... 93
Table 4.6: Representation of the array after processing all segments function .......... 95
Table 4.7: Recall and Precision results of Taxonomic phase ........................................ 107
Table 4.8: Recall and precision results of Enhancement phase .................................... 107
Table 5.1: Results of testing on real world ontologies..................................................... 128
Table 5.2: SWETO ontology case before and after ....................................................... 129
Table 5.3: Summary of “AutoPart” domain ontology .................................................. 130
Table 5.4: Summary of the compression ..................................................................... 134
ACKNOWLEDGEMENTS

In the beginning, I thank Allah for his blessings, who has helped me and gave me the power in doing this work.

I would like to also acknowledge my advisor Professor Austin Melton for his ultimate support and valuable insights, directions, advices, and guidances.

Also, I would like to extend my sincere thanks to the members of my doctoral committee, Professors Angela Guercio, Johnnie W. Baker, Alan Brandyberry for their feedback and guidance.

I have been lucky for being a member of Semantic Web Research Group (SWRG). It was very pleasant opportunity to work with enthusiastic researchers. I appreciate these opportunities for research and collaboration with them.

Last, but not least, I would acknowledge my parents for their love, support, and prayers. My wife was always there for me. I am very thankful for her support. Thank you, my daughter, for your simple words that have great impact. Finally, many thanks for my brothers, sisters, and friends for their warm wishes and prayers.

Abdulelah Algosaibi

August 2015, Kent, Ohio
CHAPTER 1

Semantic Web Introduction

As a background of the history of web generations and how Web evolved from Web 1.0 to Web 2.0, and then lately to Web 3.0, a short background will be presented for each of these three web generation versions. For each web generation, its own pros and cons, its main characteristics, its evolution to the next generation, and the main differences among each generation will be shown. Part of this chapter is accepted on The 2015 International Conference on Internet Computing and Big Data.

1.1 Web 1.0

Web 1.0 was simple in terms of information and how it was represented. In fact, it was considered to be a “static web”. Web 1.0 was limited to the features it provided and did not exceed the layout representation of a website. Most of the information on Web 1.0’s websites was on the webpages themselves. It is just about a website that does not interact with visitors to offer implicit functions. A good example of a Web 1.0 website would be the professor’s homepage that only has information about courses, publications or pictures. Web 1.0 was not implemented to offer a service that required further configurations.
Most of the attention was on unstructured documents. As a result, web technologies were simple in terms of the power of processing these documents, e.g., a web directory for navigating through a list of websites. Semi-structured documents had limited availability and did not exceed HyperText Markup Language (HTML) documents or data describing EXtensible Markup Language (XML) documents. Technology focused on how to process these unstructured documents. HTML, Cascading Style Sheets (CSS), XML, and web browser technologies were a clear attempt to enhance the functionality that users experienced. HTML is the publishing language of the World Wide Web. HTML tags on data give presentation capability that is processed by web browsers. These tags structure a webpage. This structure is used for representing information in a webpage. For example, a webpage in Web 1.0 that was a user-end interface consisted of HTML tags along with text or images. In Web 1.0, the websites interfaced with a web directory to browse these websites. Thus, Web 1.0 was a static web; simple, yet limited with respect to the available technologies.

1.2 Web 2.0

Web 2.0 is an extension of Web 1.0. It is more enhanced in terms of the features, functions, services, and usefulness than Web 1.0. Web 2.0 is considered to be “dynamic web”, e.g., social networking sites, wikis, video sharing sites, online shopping, and web applications. It is a Web-as-participation-platform [1]. If there is desktop-application software, then with Web 2.0 it is possible to have another version to act as the web-based
application. Web 2.0 is bi-directional communication [2]. All of these features possess number of back configurations to make them work. Some configuration processes, for example, start with choosing the domain (health, sports, news, etc), then building the necessary database, then writing the query code within web page code, then designing the layout of the page (drop-down menu, radio check, etc), and ending with presenting a result of a query. Web 2.0 often follows software engineering principles in order to build scalable web applications. Since they are software, following these principles will help in the maintaining and enhancing of the web applications.

Unstructured documents have a massive amount of information that increases the number of services done over the web. Semi-structured documents allow for more precise processing. Also, semi-structured documents get increased attention. The technology can more easily process these kinds of documents where it includes Natural Language Processing [3] and HTML documents parsing. Technologies in this web generation have enhanced the processing functionality of web documents. The technologies drove Web 1.0 to become more interactive. The technologies allowed the Web to be more dynamic, e.g., web forums. The web technologies allow web applications to be in place, for example, JavaScript, PHP, Python, JSP, ASP.NET and JAVA. Also, Web 2.0 became advanced enough to give the user the ability to choose the architecture of the web application whether it was client-side, such as JavaScript or a server-side, as with JSP. There are number of web editors that helped to make the creation of Web 2.0 much easier
and simpler, e.g. Microsoft Visual Studio that works with ASP.NET and C#. In fact, with the evolution of the Web 2.0, the Web became richer in its content. As a result, navigation through the web, particularly within the website, became harder. Navigation methodologies, e.g., sitemaps or onsite search and search engines, e.g. Google, were invented to help speed up the process of finding the desired content.

1.3 Web 3.0

Web 3.0 aims to help retrieve the right piece of information in a simpler and faster way. It is about putting semantics on the Web. Web 3.0 is considered to be an extension of the current Web, not its replacement [4]. The amount of the information, how to access it, and how to enable delegation were main issues in Web 2.0 [5]. The current content of the Web is for humans to read and not for machines to understand. Semantic Web creates an environment that allows software and agent to understand the Web content [4]. The development of the Web 3.0 or Semantic Web is implemented in steps as shown in figure 1.2. These steps or components are discussed in the next section.
To visualize these generations, figure 1.1 shows the line of development of the web, captured from (www.radarnetworks.com). Semantic Web is about making the Web more easily understood by machines. Thus, producing better results and overcoming the drawbacks of Web 2.0. It is a technology that assists in referring inquiries to the most preferable destination.

Figure 1.1 Web generation time line
In more details, Semantic Web technology has a number of components as shown by [6] in figure 1.2 below.

Next is a brief description of some basic Semantic Web components.

Figure 1.2: Semantic web stack layer
1.3.1 eXtensible Markup Language (XML)

The XML is a tag-name layer that provides a way of formatting documents and describing the data. It is mark-up language with no predefined tags. Thus, it does not provide semantics. Therefore, it is not the solution for propagating semantic.

1.3.2 Resource Description Framework (RDF)

RDF is a simple way to model the data. It allows describing of the web resources. This resource can be simply divided into subject, predicate, and object. It gives meaning to XML structure. As RDF is defined from\(^1\) “RDF is a standard model for data interchange on the Web.” An example below shows how RDF describes (http://www.kent.edu/ has title that is Kent State University) where the subject: http://www.kent.edu/, predicate: title, and object: "Kent State University". RDF sets some rules that modeling the data semantically allows the machine to understand it. RDF data model can be queried using SPARQL end point.

\[
\begin{align*}
<\text{rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"} \\
\text{xmlns:dc="http://purl.org/dc/elements/1.1/"}>

<\text{rdf:Description rdf:about="http://www.kent.edu/">}

<\text{dc:title}K\text{ent State University}</dc:title>

</\text{rdf:Description}>
\]

\(^1\) http://www.w3.org/RDF/
1.3.3 RDF schema (RDF-S)

It is designed to be a simple description or metadata model for RDF\(^2\). By using RDF Schema, RDF tags can be classified. RDF-S allows properties, classes, and relationships to be created. A visualization of an example can be seen in figure 1.3:

![Diagram of RDF Schema](http://www.w3.org/TR/rdf-schema/)

Figure 1.3: Visualization of RDFS

---

\(^2\) http://www.w3.org/TR/rdf-schema/
The figure 1.3 is representation of below example:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

  <rdfs:Class rdf:ID="Vehicle"/>
  <rdfs:Class rdf:ID="Car">
    <rdfs:subClassOf rdf:resource="#Vehicle"/>
  </rdfs:Class>
  <rdfs:Class rdf:ID="Truck">
    <rdfs:subClassOf rdf:resource="#Vehicle"/>
  </rdfs:Class>
</rdf:RDF>
```

1.3.4 **Ontology**

It is a “formal, explicit specification of a shared conceptualization” [7]. It explains the formal shared conceptualization of a specific domain. Ontology is well-suited for
describing heterogeneous, distributed, and semi-structured information sources [8]. By defining shared and common domain knowledge, ontologies help both people and machines to communicate concisely. Ontology consists of classes and relationships that define and describe concepts within a domain, e.g. in a university ontology, a college would be a class. Ontology Web Language (OWL) is a machine-readable language that expresses ontology. For example, a very simple ontology would appear as such:

```xml
<Ontology xmlns="http://www.w3.org/2002/07/owl#">
  <Declaration>
    <Class IRI="#Administrative_Staff"/>
  </Declaration>

  <Declaration>
    <Class IRI="#Employee"/>
  </Declaration>

  <SubClassOf>
    <Class IRI="#Employee"/>
    <Class IRI="#Administrative_Staff"/>
  </SubClassOf>
</Ontology>
```

Facts within the knowledge can be inferred using logic. Logic allows drawing conclusions from existing knowledge. It is expected to enhance ontology layer
functionality by deducing new facts or conclusions from an explicitly defined knowledge within an ontology.

1.3.5 Proof and Trust

Proof is step for validating the origin of data or provenance knowledge. It requires an authentication when working with individual metadata items. Some of the data may be provided from untrusted sources. Thus, trust is evaluated using some techniques, such as digital signatures or a trusted agent. A trust layer is usually supported by relying on the derivation of the formal logic or trusting the source using certification agencies e.g. trusting the agent or trusting the operations [9].

1.4 Web Generations’ Evolution

Conversion from Web 1.0 to Web 2.0 is optional. It depends on the need to have an interactive website. Rather, conversion from Web 1.0 or Web 2.0 to Web 3.0 is necessary because of the enormous amount of information and the way information is accessed, which were major concerns with the first two generations [5]. Semantic Web vision requires a number of sources to share their information in a semantically structured manner so that it allows machines to process these sources semantically. It is a technology that would have a great impact over the service provided by Web Wide Web. According to [10], “The Semantic Web provides a common framework that allows data
to be shared and reused across application, enterprise, and community boundaries.” The Semantic Web is about Read-Write-Execute the Web [11]. Table 1.1 shows a comparison among Web generations.

<table>
<thead>
<tr>
<th>Web 1.0</th>
<th>Web 2.0</th>
<th>Web 3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Web</td>
<td>Dynamic Web</td>
<td>Semantic Web</td>
</tr>
<tr>
<td>HTML, XML</td>
<td>JavaScript, PHP, JSP, ASP</td>
<td>OWL, RDF, SPARQL</td>
</tr>
<tr>
<td>Read Web</td>
<td>Read-Write Web</td>
<td>Read-Write-Execute Web</td>
</tr>
<tr>
<td>Information source but</td>
<td>Functional but enourmance</td>
<td>Semantic linkage of knowledge.</td>
</tr>
<tr>
<td>limited features.</td>
<td>amount of information.</td>
<td></td>
</tr>
</tbody>
</table>

Table 1.1: Main characteristics of each Web generations

1.5 Goals and Contributions

It is a fact that Web 3.0 is an extension of the previous web generations [4]. Thus, this work recruits some advantages that are available on these generations to contribute in this extension for Web 3.0 and future generations. This work mainly discusses these following points:

- To present web generations and their main characteristics, advantages, and shortcomings. All that, with respect to documents and technologies types available for each generation. Also, to study the evolution of the web generations from an information processing point of view for getting more
relevant content. Also, to briefly present the main components of the Semantic Web (Ontology) and discusses its roles.

- This work presents a novel approach to using sitemaps to help create Web 3.0 ontologies that link web generations by using the available information in sitemaps (HTML sitemaps). A short survey presents the types of sitemaps. Further, this research provides an explanation about Natural Language Processing (NLP) and some of its applications.

- To study, in depth, several well-known ontology learning tasks and then presenting in details the proposed approach. This work proposes a semi-automated approach for learning domain ontology. It first evaluates related sitemaps and proposes “Basic Mode” and “Advanced Mode” for better process an HTML sitemap. Then, how to appropriately and semantically group the HTML sitemaps are presented. Given a set of HTML sitemaps from one domain, the proposed approach will first identify taxonomic relations to start building an ontology and then enhance the ontology by enriching and refining it. Also, this work explains the implementation parts and shows how to extract concepts, taxonomic relationships, instances, and non-taxonomic relationships.

- To present a three dimensional ontology modification matrix. Indeed, ontologies are allowing sharing of knowledge so that it can be reused by different user/application. In this part of the dissertation, an approach is
being proposed that helps in estimating the effort associated with modifying it as an attempt at relatively indicating the complexity of an ontology. That is, this approach can assist in estimating ontology modification efforts and tracking ontology development processes. Such an approach, considers how far a given ontology is from the three dimensions origin coordinates based on the three dimensions distance function. An empirical analysis is presented and discussed. Also, a number of applications are presented to show the advantages of the proposed work.

- Also in this work, web generations (Web 1.0 and Web 2.0) and Semantic Web (or Web 3.0) are reviewed. Moreover, a projection of the future web generations (Web 4.0 and Web 5.0) is surveyed. That is, a prospecting study about the future Web and trends is presented.

### 1.6 Organization of the Dissertation

This chapter discusses web generations (Web 1.0 and Web 2.0) and some of their technologies employed to lead each generation with respect to document types and technologies. Also, Web 3.0 is introduced by explaining its main component. Chapter 2 describes ontology learning approaches with respect to document and learning types. Moreover, overview of ontology learning techniques is presented. Finally, an intensive review of the related work of this dissertation is presented. Chapter 3 gives an idea about the working environment for the proposed work. In this chapter, HTML document and its
processing are presented. Further, an explanation about Web searching paradigms is presented. Later, overviews about the types of sitemaps are presented. Also, a brief introduction about Natural Language Processing (NLP) is presented. Moreover, an overview about the employed text similarities matching system is provided. Last, it goes specific into Web 3.0 component and specifically into Ontologies to introduce ontology main engineering tasks. A detailed description about how to use HTML sitemaps together in order to learn an ontology is presented in chapter 4. This chapter presents the implementation of each phase of the proposed approach. In addition, an evaluation and discussion of the results are presented. In chapter 5, a three dimensional ontology modification matrix approach is proposed. Furthermore, a discussion about its insights, applications, and advantages are presented. Chapter 6 summarizes the work and discusses future web generations (Web 4.0 and Web 5.0). Finally, future recommendation and work are presented.
1.7 Summary

An introduction to the web generations has been discussed showing that web, so far, has three generations. Also, the characteristics of each web generations were discussed and the shortcomings of each web generation have been pointed out. Later, the stack layer of the Semantic Web was presented with some overview of basic components. Also, a discussion of the evolution of web generation was presented. Web 1.0 and Web 2.0 show that the Semantic Web is an urgent need to overcome the issues of vast amount of the information. To conclude this chapter, the goals and contributions of this dissertation and how the dissertation is organized were presented.
CHAPTER 2
BACKGROUND AND RELATED WORK

Ontology, as defined by [7], is “formal, explicit specification of a shared conceptualization.” In this context, formal refers to the ontology has to be a machine-readable language. The phrase explicit specifications mean that the concepts and their use are clearly defined. Shared conceptualization emphasizes the notion of an abstract model for a given domain whose concepts are shared and not kept private. Ontology aims to provide formal domain understanding to machines and humans. It is considered to be the backbone of Semantic Web technologies. Ontologies can be categorized according to how they are used [12]. For example, foundational ontology is concerned with being so general that all domains can use it; in contrast, application ontology is focused on the domain and the application for which it was created. Ontologies represent classes, and these classes have objects. Each class and object comes with a set of properties relationships. An example of this is class inclusion (owl:SubClassOf) and object properties (owl:equivalentProperty).

2.1 Introduction to Ontology Learning

Ontology learning aims to create ontology by learning it from given source. This source heavily affects the learning procedure. Input may come from three categories: structured, semi-structured, and unstructured [13]. The ontology learning methodology
relies on those sources. Ontology learning methodology also is categorized as automatic, semiautomatic, or manual learning methodologies.

2.1.1 Type of Documents

As mentioned earlier, the kind of document to be input to an ontology learning methodology is divided into three kinds. Below is a brief overview of each of them.

2.1.2 Unstructured Documents

On the unstructured document or information, the knowledge is available in free structure format that does not follow any predefined formation or representation structure, e.g. Text file.

2.1.3 Semi-Structured Documents

With semi-structured documents, the data are structured, at least, in part, based on semantics; however, the underlying structure and the semantics are not explicitly given. The structure is still helpful in extracting the semantics from the document. A semi-structured document, if handled carefully, can produce relatively rich semantic information because the semantics are embedded in the data and the data’s structure. In other words, there is more semantic provided in data [13]. Examples of semi-structured documents are HTML documents, WordNet [3], and XML documents.
2.1.4 Structured Documents

Structured documents mean that the formation of the document and inner data are structured. The formation of the document and the inner data are set up in a way that for each piece of information, it is explicitly known how that piece of information fits in with the other data. This leads to retrieve more relevant data. Thus, the semantic information extracted from structured documents is rich [9]. An example of a structured document is a database. An example of a database about auto prices is shown in table 2.1:

<table>
<thead>
<tr>
<th>Make</th>
<th>Model</th>
<th>Price</th>
<th>Date listed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audi</td>
<td>A8</td>
<td>200,000</td>
<td>4/2/2010</td>
</tr>
<tr>
<td>Toyota</td>
<td>Camery</td>
<td>100,000</td>
<td>6/7/2012</td>
</tr>
</tbody>
</table>

Table 2.1: Auto information table

Each table is considered to be a class, each attribute is considered to be a property, and each row is considered an instance. From the table above, Toyota instances of class called “auto”, each instance has property made, price, and date listed. Structured documents, however, suffer from deciding which piece of source information can be useful as “date listed” in table 2.1 above.
2.2 Types of Ontology Learning

Learning ontologies can be categorized according their learning methodology: automatic, semi-automatic, or manual. These methodologies differ in terms of the automation of information extraction or learning the ontology, and the amount of human intervention needed with the approach [13], [14]. Below is a brief overview of these methodologies.

2.2.1 Manual Learning

In manual learning, learning procedures are not automated. Thus, much time and effort go into building the ontology. It is also labor intensive because the ontology engineer must create each domain. Manual learning, however, is semantic rich because it is manually crafted [9]. According to [15][9], there are eight stages involved in the manual construction of an ontology. These stages are not linear, some steps are repeated more than others, and backtracking may be needed. These eight stages are determine scope, consider reuse, enumerate terms, define taxonomy, define properties, define facets, define instances, and check for anomalies.

2.2.2 Automatic Learning

With automatic learning, techniques are fully automated. This means that there is no human involvement in any stage of a given techniques. Research [16] mentioned possible approaches from machine learning that lead to the automated development of
ontologies. These approaches are propositional rule-learning algorithms, Bayesian learning, first-order logic rules learning, and clustering algorithms.

2.2.3 Semi-automatic Learning

The semi-automatic learning approach is the most applicable approach for ontology learning [12]. This type of learning takes the advantages of the both automatic and manual approaches. It also solves the disadvantages of both of them. It can apply the Machine Learning techniques used on knowledge acquisition, extraction, revision or maintenance [9]. It is guided by the human, thus the quality of the work is on a higher level [16]. With this so-called balanced cooperative modeling paradigm, “each step may be done by the user or by algorithm.” [17].

2.3 Reusing Available Ontology

When creating an ontology, it is sometimes worthwhile to look at existing ontologies to avoid starting from scratch. Creating an ontology from scratch is not preferable. Information sources, or ontologies, have become more widely available. As mentioned earlier, document types vary from structured to unstructured. Thus, these ontologies, or knowledge resources, vary from semi-structured documents to structured documents. Some sources that are introduced by [9] and may be considered as a starting point when creating an ontology. These sources are sorting expert knowledge that has been constructed by experts. These sources are integrated vocabularies such as the
Unified Medical Language System\textsuperscript{3} group 100 biomedical vocabularies, upper-level ontologies can help to connect different disciplines together, topic hierarchies, such as Open Directory Project\textsuperscript{4} hierarchy, are grouping the related domain sources together, linguistic resources, such as WordNet [3], are for linguistic purposes and it is possible to be extended for ontologies purposes. Also, encyclopedic knowledge, such as Wikipedia\textsuperscript{5}, that is based on free contribution from around the world. One project that took the advantages of the knowledge on Wikipedia is DBpedia\textsuperscript{6}. Moreover, ontology libraries such as Swoogle\textsuperscript{7} are online ontologies repository aimed to group number of ontologies from number of domains.

### 2.4 Ontology Creating Techniques

A number of techniques proposed for learning an ontology [14][18][19][13][12]. These techniques are a bit different because of the kind of source data they use. Some details may need to be adjusted depending on the nature of the input and how to process it. In here, a summary of some of the above approaches is provided.

\textsuperscript{3} http://umlsinfo.nlm.nih.gov

\textsuperscript{4} http://www.dmoz.org/

\textsuperscript{5} https://www.wikipedia.org/

\textsuperscript{6} www.dbpedia.org

\textsuperscript{7} http://swoogle.umbc.edu/
Starting with Ontology Learning for the Semantic Web [12], the author defined four main phases—import, extract, prune, and refine—as shown in figure 2.1, below. These phases can be applied to build an ontology or maintain an ontology.

During the import/reuse phase, the user selects relevant ontologies to be input with providing clear methodologies for dealing with the data. The transformation of representational languages has to be clearly defined and merged with the conceptual structure. During the extract phase, major parts are modeled. This phase takes advantage of the availability of previous knowledge. Thus, the learning strategies follow the iterative approach to learn the relevant data. An example is the bootstrapping model. The prune phase involves removing unneeded data from the ontology and adjusting the ontology for its needs, which includes identifying how updating one part of the ontology
affects other parts of the ontology and which items should not be changed. For example, a concept that has not been used within an ontology should be removed. Pruning should involve clear strategies to determine the level of pruning and not end up with large or small ontologies. The refine phase focuses on adapting user requirements and needs, checking for semantic conflicts, and testing the ontology so that it becomes more acceptable.

The research paper “Ontology Learning: Revisited” [13] included the main steps for building an ontology. As the figure 2.2 below shows, there are four phases: concept learning, taxonomic relationship learning, non-taxonomic relationship learning, and axiom and rule learning.
Concept learning involves retrieving terms on the domain to build the hierarchy structure (i.e., exploit the base knowledge for a given domain) before initial taxonomic concepts are initiated. Non-taxonomic relationship learning deals with finding relationships that are not associated with taxonomic relationship. This includes drawing a relationship between related concepts and getting the name of the relationship. The axiom phase is about learning the rules within the ontology.
Another technique proposed for ontology development process proposed by Knowledge Process and Ontologies [19] and has four phases: ontology kickoff, refinement, evaluation, and maintenance as shown in figure 2.3, below.

![Ontology Development stages](image)

Figure 2.3: Ontology Development stages by [19]

Starting with the ontology kickoff, the ontology development is guided by an ontology engineer that checks the modeling of the concepts. The concepts should describe the information about the goal of the ontology, the domain and scope of the ontology, the applications supported by the ontology, the knowledge sources (e.g. domain experts, organization charts), potential users and usage scenarios, a competency questionnaire, and potentially reusable ontologies. At the refinement phase, a draft of the ontology is generated according to the requirements collected during the kickoff phase.
(i.e., informal basic taxonomy and knowledge derived from the domain expert). The evaluation phase is the testing stage and has two steps involved. In the first step, the ontology engineer checks the ontology. In the second step, the ontology is checked against the target application. The evaluation phase is linked to the refinement phase, which continues until the ontology is ready. The maintenance phase concerns with updating the current ontology as needed. The authors recommend starting a new version of an ontology whenever an update is due and putting that version through the refinement phase and the evaluation phase.

2.5 Related Work

In this section, a brief overview of the related work will be presented. During the literature review, some interesting work was discovered and will be summarized here. The reader will be able to see how the proposed work in this research is distinguished from the work in the literature.

The research paper titled “OntoMiner: Bootstrapping and Populating Ontologies From Domain-Specific Web Sites” [20] introduces an automated technique to bootstrap and populate specialized domain ontologies by manually collecting sets of relevant websites, turning the websites into hierarchical semantic structures in Extensible Markup Language (XML) format and identifying domain concepts and their taxonomical relationships. The proposed approach starts with a flat partitioner algorithm that takes the
Document Object Model (DOM) tree of the webpage as input and flatly partitions it. Flat partitioning groups adjacent similarly structured webpages into partitions by calculating the highness of adjacent repeated nodes that are similar from the root to the leaf tag path. Hierarchical partitioning that deduces the hierarchical relationships among leaf nodes of HTML tree is done by applying binary semantic partitioning, grouping, and promotion. For the binary semantic partitioning, a dynamic programming algorithm approach is applied to determine nodes to be grouped with minimal cost. The grouping works by finding pairs that are similar by post-order traversal of the semantic binary partition. On the promotion, an algorithm is proposed for identifying the leaf nodes that should be upgraded above their siblings. This stage of the system is measured by manually creating an ontology and calculating the precision and recall with average of 81% for precision and 80% for recall. Later on, while mining the taxonomic relationship, the authors took advantage of an observation that important concepts in one domain are repeated. They identified similar concepts and mining relationships among the concepts. This stage also was measured by manually creating an ontology and calculating the precision and recall with 75% for precision and 92% for recall. Then, the authors extracted instance segments by taking the segments region from webpages. Each segment holds number of URLs that are tested and flatly partitioned and then aligned according to their content similarity. The dissimilar segments are used to extract attribute tag paths. If the paths are similar, then the webpage segment is counted as the one with instances as shown in figure 2.4, below. Then, the authors extracted the frequent concepts from these segments. These frequent concepts correspond to the name of the attributes of the
segments. The authors mentioned that in some case the labels are not a frequent label above or below the labels. Thus, they group data values from their associated paths. The authors deemed the issue of labeling to be an issue for future research. The precision was 64%, and the recall was 97% for the extracting instances.

The research paper “Managing knowledge on the Web – Extracting ontology from HTML Web” [21] presents an ontology extractor, called OntoSpider, for extracting ontology from HTML web documents. The extractor uses a six-phase process that includes preparation, transformation, clustering, recognition, refinement, and revision. It is a semiautomatic approach. Basically, it starts loading a corresponding webpage and then converts it to be a text-based, well-formed HTML page. Further, the researchers
studied outgoing hyperlinks of a webpage (content) and assign a destination page that does not point to itself. The content of the page is then summarized. In a later step, the researchers studied the Document Object Model (DOM) tree that focused on a <head> and <body> HTML tag structure. They further dug deep on HTML <table> tags. The researchers concluded that the HTML <table> tag would have sections from A to E that are emulated from a <tr> HTML tag as shown in figure 2.5, below.

The researchers followed the Breadth First Search algorithm to remove any duplication. In instances of clustering, each webpage is represented in a t-dimensional vector for clustering. Later, they reached the conclusion that measuring the structural similarity between two webpages is complicated. Thus, to simplify the calculation of clustering instances, they applied the Breadth First Search algorithm on each webpage of the website and calculated the content similarity of their child webpages. They assumed that

Figure 2.5: Studying of the HTML tags
instances have a similar structural only if the webpages followed the same hyperlink path from the home page to the parent webpage. On instances recognition phase, the task is to identify patterns of the clustered webpages and use frequently occurring patterns to identify non-clustered web pages to improve webpage recall. Later, the researchers applied the Breadth First Search algorithm to traverse the webpages to measure their similarity. If the similarity result between the vectors is above a preset threshold, then the unassigned webpage is added to the relevant clusters. Later, they clustered the webpages into groups using four rules to extract concepts: hyperlinks between unclustered webpages, hyperlinks between clustered pages (concepts), hyperlinks between clustered web pages (concepts) and unclustered webpages are kept as unclustered web pages, and hyperlinks within clustered web pages (concepts) are ignored. The researchers then refined the relationships, which are in their case the hyperlinks. Finally, the ontology engineer maintains the ontology in order to be ready for delivery. The result of their work is measured at each of the six phases. At the clustering phase, the threshold tested from .4 to 1.0. When the testing was done and the researchers took into consideration the structure of a website, the average result of recall was 20.85%, and the average result of precision was 87.25%. At the recognition phase, the threshold tested from .4 to 1.0. When the structure of a website was taken into consideration, the average result of recall was 70.37%, and the average result of precision was 61.80%; however, when the threshold was set to .7, the recall was 85.39%, and precision was 70.05%. Also, they compared the output of ontoSpider with another project called “SHOE” [22]. They could get more concepts and more relationships. They declared four limitations of their work:
the issue of direct hyperlink points to related pages; the command-button issue; the document formatting issue; and the lexical semantics and natural language issue. The research “Domain Ontology Learning from Websites” [23] proposes an approach for learning domain ontologies from semi-structured data. The researchers followed the hierarchical structure of websites to extract the ontology. This paper focuses on learning organizational ontologies from websites. In an organizational website, the individual-relevant pages, that are webpages describe individual, are retrieved from the target domain. To identify those individual pages, the researchers adopted a Hierarchical Navigation Path (HNP). They modeled individual-relevant pages by applying a two-step technique: path-query and content-query. The purpose of path-query “is to retrieve individual-relevant pages by querying the HNP’s text nodes.” For example, the user can retrieve “products” from a company website by querying “products.” The result from the path-query contains more than individuals. Thus, they support the result by a content-query. The content-query “is designed for picking out the individual profile pages.” The researchers then began extracting concept and relation identification by adopting the Visual-Layout Based Segmentation (VIPS) method. They observed that webpages having two adjacent blocks with an above/left block contain one hyperlink, while the below/right block contain groups of hyperlinks. In addition, the below/left block is a potential ancestor webpage of the below/right block. The researchers claim that nodes referring to individual entry pages at the bottom level of the hierarchy could be assigned as individuals. They observed that each node referring to an individual webpage that contains a category of individuals could be mapped to concepts. Also, they observed that
the relationships in ontology could be mapped between nodes (webpages) in a hierarchy from hierarchical hyperlinks. To get the title of a group of webpages, they observed that the title is located adjacent (above or left) to the group. They also observed that titles with common terms mentioned on the groups might be a title of the group. This observation led to generating an organizational ontology. The researchers declared that their contribution was the ability to handle the case where a concept or an individual is spread on the entirety of the webpages. They found the precision of the product bit low (41%). The reason behind this, as they claimed, is a context-aware product group (e.g., related products or featured products that led to the first drawback). The other drawback is that the user needs to manually select the websites.

These mentioned researches have considered the layout of the webpages in the work they proposed. Different from these research papers, the proposed work does not consider the layout of the webpages because layout varies from website to website. These research papers studied intensively the structure of websites that differ from one website to another. Consequently, the proposed approach exploits the HTML sitemap that is not affected by the layout and allows a user to navigate a website in a related manner. Thus, some issues, such as direct links to other websites and document formatting mentioned in [21] are not applicable in the proposed work.

The research paper titled “Ontology Learning from Domain Specific Web Documents” [6] proposed an ontology learning approach that starts with the seeding of
two inputs. The first input is the manually seeded domain concepts, and the second input is the HTML documents. Then, the researchers processed these two inputs by using the structure of HTML headings and by identifying the concepts with their taxonomical relationships between seed concepts and between each other. They proposed a seven-module architecture for ontology learning: heading extractor, heading preprocessor, N-gram based ontology learner, ontology refiner, HTML structure based ontology learner, HTML ontology refiner, and ontology merger. On the heading extractor, the headings are extracted from an HTML document. On the heading preprocessor, the titles are normalized by removing any stop words and by stemming the titles. The N-gram based ontology learner module extracts the taxonomic relationships using the seeds concept as a parent and finding a relationship within words in headings. The ontology refiner module works by applying a number of filters to refine extracted concepts. The HTML structure based ontology learner module is introduced to resolve issues with decreased recall of concepts and to place the concepts in the right spot. Thus, the concepts input by the user are to be in the first level with taking into consideration any frequent information on concepts. The goal here is to set the concept as child of the most specific parent. The researchers mentioned the following example: If there is “fungal disease” concept that is the child of “diseases” and “powdery mildew” appeared as a direct child for “diseases” 14 times and as child of “fungal disease” two times, it will be still be a child of the “fungal diseases” HTML. The ontology refiner module would work to extract concepts that have a sibling relationship with concepts learned in the previous module. Ontology merger finalized the output ontology by merging the ontology of the HTML structure.
based ontology learner and the N-gram based ontology learner into one ontology. Their results are evaluated based on two methods. They are the lexical evaluation and taxonomic evaluation. The best refinement result found when applying a term frequency filter for the recalls results are 42.6% for N-gram ontology and 73.5% for the HTML structure based ontology. The recalls for taxonomic evaluation are 46% for the N-gram ontology and 76.7% for the HTML structure based ontology. The precision results for the lexical evaluation are 76.7% for the N-gram ontology and 70.3% for the HTML structure based ontology. The precision results for taxonomic relationships are 73.7% for the N-gram ontology and 66.7% for the HTML structure based ontology. The authors claimed that the best obtained results came when merging the N-gram ontology after applying a frequency term filter with the HTML structure based ontology without refinement. The authors are considering the involvement of a domain expert to improve the recall and precision results. Research titled “Concept Extraction Based on HTML Documents Structure” [24] shows a method that considers using an HTML document to extract concepts of ontology within a field. The process starts with removing HTML tags to get the corpus of the HTML document. This corpus gets more refinement by removing stop-lists words (e.g., article or introduction). Next, the process extracts candidate terms by using two types of syntagms based on their structure (i.e., one-word name or name entity and two words as syntactic structure sequence, such as an adjective name) and filtered by some criteria such as their appearance in titles or style. In concept, the researchers tested candidate concepts by measuring their relevance. If a candidate concept passed a given threshold, it would be selected to represent the concept of the field. The researchers
found words in titles or hyperlinks of documents with their style generally containing relevant terms of the field and therefore easier to detect. The researchers noted that the length of the title affects the syntagms’ relevance. Also, they considered the less depth of a title within HTML documents, the more relevant the syntagms. In addition, they studied some HTML style tags (e.g., bold and italic). Their results were tested on 23 documents from Wikipedia. The researchers extracted 2,801 concepts and after these refinements, there were 134 concepts. They used a specialized dictionary to validate the extracted concepts. The best precision they got was 66.43% when the document structure was considered. In research [25] the work is based on considers hyperlink relationships and has two phases: preparation and annotation. The process starts with crawling the web document. After that, an ontology dictionary is defined in order to assign dictionary labels for concepts and relationships. Then, the pages were analyzed for tags and collecting vector data for all matches between tags and concepts. HTML tags such as <title>, <head>, and <body> were prioritized to determine the highest priority. The process also analyzed links based on the text around the hyperlink and categorized the links based on information contained in them. Then, the researchers calculated the concept weight based on each link size (e.g., 50 words would be 25 words before and 25 after). Later, and after getting the relationships, the researchers matched the relationships with ontological concepts and annotated them based on voting algorithms for concept matching that match a page or a link. Voting is done if there is more than one concept matching for a given page or link. Finally, the researchers defined a relationship based on the concept of the page with the link and the page to which the link pointed. To obtain
their results, the researchers crawled 200 webpages and 4,599 weblinks. They could match some concepts to 174 pages and assigned concepts to 3,463 weblinks. They claim that they need to tune the ontology because there were no concept matches for some of the webpages. Furthermore, they claimed that using some labels for a given concept may not be best idea if their work were compared to Natural Languages Processing.

Different from the above work, the proposed work; takes in consideration further steps that these research papers have not addressed. The proposed approach includes extracting concepts, taxonomic relationships, and non-taxonomic relationships. Also, a Natural Languages Processing (NLP) tool being considered in order to refine and to get more precise relationships. NLP techniques would be used in number of phases such that all together allow to get precise ontology.

The research paper [26] talks about improving the original sitemap protocol to deal with Semantic Web datasets. The authors mainly want to show the problem of RDF data discovery and the computing performance when the RDF data are retrieved by clients and search engines. In other words, they want to take advantage of the existing of XML sitemap and put in new XML tags (e.g., <sc:dataset>, <sc:dataDump>, and <sc:sparqlEndpoint>) for announcing the existence of RDF data and to support existence requirements. The researchers mentioned couple of challenges related to the previous publication of the RDF, e.g. crawling performance and disconnected datasets. The proposed extension is based on announcing and describing the dataset to resolve the
mentioned challenges and to announce the presence of RDF data. As for their results, they were reported to a Sindice Semantic Web indexing engine [27] and indexed it. They reached more than 26 million RDF documents. They were considering reporting that to SWSE Semantic Web Search Engine [28] service. The research paper “Automatic Annotation of Content-Rich HTML Documents: Structural and Semantic Analysis” [29] contributed to bridge the gap between semantics encoded into HTML documents and machines via an automatic approach. The researchers tried to partition a web document to be semantically structured. To do this, two main tasks are involved. First, identify segments that correspond to semantic concepts. Second, to assign labels to these identified segments. The researchers propose a semantic analysis technique and incorporate that with lexical source (e.g., WordNet) [3]. They analyzed the structure and transformed an HTML page into the corresponding semantic structure. Also, they took advantage of a location from a Document Object Model (DOM) tree as shown in figure

Figure 2.6: DOM tree of New York Times front page snapshot from [29]
2.6, below. Regarding the semantics underneath the HTML tags, however, these tags may or may not have led to semantics. The researchers tried to find the consistency in presenting the HTML item. They tried to overcome this by finding how the presentation looks in other sections based on the rule they had defined. Later, they used the linguistic approach to semantically relate two pieces of text into one segment. They labeled the partitioned segment and then matched the pieces of text by the keywords of a concept into an ontology. The experiment resulted in more than 50 HTML documents from eight different news portals (New York Times, CNN, Yahoo News, Google News, ZdNet, CNet, Bloomberg News, and Recorder News). They calculated the recall and the precision. First, they manually identified all of the ontology concepts and the correct instances for each document. Then, they applied the approach and counted the number of partitions. Detailed news pages (e.g., CNN and CNet) have very high recall and precision. The recall for less detailed news pages (e.g., Google News and ZdNet), however, was low. The researchers claim that this is due to the incompleteness of their semantic analysis. Additionally, the research paper “Bootstrapping Domain Ontology for Semantic Web Services from Source Websites” [30] focuses on the websites with query interfaces as shown in figure 2.7, below.
The authors proposed a system called “DeepMiner” that starts with learning base ontology from a webpage interface and then following a cycle to grow the ontology. At each cycle, the developing ontology was utilized to train label classifier and instances classifier. Then, it queries the system and gets data pages returned. In here, the label of concept and its instances usually are located close to each other. To determine relative positions, the authors employed Document Object Model (DOM) tree and a traverse data region using a Pre-Order traversal algorithm. They claimed that data regions often correspond to either left-right or top-down ordering. And since the webpages are generated automatically, it often follows the HTML <table> tag in populating and presenting the results. Finally, the obtained concepts and instances were merged with the ontology and iterated through the system again to mine additional concepts and instances. If a label-instances pair that is extracted from a data record belongs to a concept that is already defined, that pair will be added as labels and instances for that concept.

Figure 2.7: A snapshot from [30] to show query interface and the result of a query
Otherwise, a new concept will be created. The results of their work were on discovering unique concepts over base ontology, identifying the data region, and discovering concepts and their instances over three domains: automobile, book, and job. For the base domain, the recall is given by the domain expert and was 98.9%, 90.4%, and 91.2% for the automobile, book, and job domains, respectively. For the base domain, the precision was identified by the system and was 100%, 100%, and 94.6% for the automobile, book, and job domains, respectively. On identifying the data region and discovering concepts, the results were done manually and then compared to the proposed system. The precision was 85% for the automobile domain and 100% for both the book and job domains. As for discovering concepts and their instances, the precision was 95% for the automobile domain and 100% for the book and job domains. The authors mentioned the issue presentation pattern and said they relied on analyzing the appearance of the attributes on the Document Object Model (DOM) tree. The research “Web-based Ontology Learning with ISOLDE” [31] authors show how to bootstrap the semi-structured web documents such as Wikipedia, Wiktionary and a German online dictionary (DWDS) along with a domain corpus and a general purpose named-entity tagger to construct a domain ontology. Their work is basically divided into three steps. First, they use a domain-specific corpus, a base ontology and use Name Entity Recognition (NER) system to extract instances for the classes in the base ontology. Then, they collected the linguistic contexts of the instances derived in the previous step and found class candidates by using lexico-syntactic patterns. The result is a list with extracted candidate classes for each named entity. Third, they retrieved the information from data around the extracted
candidate classes from online resources Wikipedia, Wiktionary, and a German online dictionary (DWDS). Their results showed that extracting ontology classes is better (precision 35.3%) than extracting relationships (precision 21.6%) between these classes. They refer that to other relations besides class levels because there are many relations that system will be incorrectly assigned.

All of the above research papers differ fundamentally from the proposed work in terms of the architectures being processed. The proposed work is bootstrapping different kind of documents and follow a different architecture, on that matches the nature of an HTML sitemap. Besides, the proposed work is extracting all potential relationships between instances. In this work, the semantic of HTML sitemaps for number of websites is utilizing.

Evaluating ontology during construction and development allows improving the overall quality of an ontology. For example, it allows ontology owner or developer to estimate complexity, enhancing quality, and reduce future maintenance cost. On research [32], a number of ontology metrics has been proposed for measuring and evaluating the quality of an ontology. The contribution was two categories: schema metrics and instance metrics. Schema metrics evaluate the modeling of the ontology and the knowledge richness (e.g., attribute richness that computes the number of attributes for each class). Instance metrics evaluate the placement of the instance data and the modeling of the data. Instance metrics are divided into two categories: knowledge-based metrics and class
metrics. Knowledge-based metrics can measure knowledge as a whole (e.g., a metric for class richness that measures how instances are distributed across classes). Another metric calculates average population and measures the ratio of instances to classes. Class metrics evaluate how the classes defined in the schema have been used in the knowledge-based metrics (e.g., a readability metric that shows the existence of human readable explanations within the ontology such as comments, labels, or captions). The researchers claim that these metrics are helpful in measuring the quality of the ontology (e.g., these metrics allow developers to automatically recognize areas that might need more work and what might cause problems). Another research paper [33] proposed analytics metrics with the aim of addressing the impact on ontology quality process. These metrics consider some ontology characteristics such as size and structural characteristics. They claim that the larger the metric result, the more effort required to understand and maintain the ontology. For example, the size of vocabulary metric that computes the amount of classes, individuals, and user-defined properties and the edge node ratio metric that computes the ratio of the number of edges divided by the number of nodes. Also, they proposed depth of inheritance metric that computes the longest path from a given class C to the root class in an ontology inheritance hierarchy. The researchers claim that their proposed metrics can help control the quality of the ontology. They also claim that these metrics can help managers to understand the development status. They have made an experiment on each metrics proposed. For example, size of vocabulary metric showed results range from 52 (the Amino-acid ontology) to 134K (the Go_daily-termdb ontology) that reflect the amount of vocabulary used in these two ontologies. Research
[34] proposed a hierarchical framework that includes 160 features distribute across five dimensions to assist the quality and relevance of ontologies to users’ program specifications. The dimensions are: content and its arrangement within ontology, language used, development methodology, building tools to manage the ontology, and usage costs. The authors claim this multilevel framework allows examining an ontology applicability. User would need to manually input values into OntoMetric that will be used to evaluate the relevance of an ontology for the given program requirements. Research paper [35] has proposed three metrics that are number of root classes, number of leaf classes, and average depth of inheritance tree of all leaf nodes. Number of root classes is a metric that counts the number of root classes defined within a given ontology. Number of leaf classes is a metric that counts the leaf classes (classes with no subclasses defined within a given ontology). Average depth of inheritance tree of all leaf nodes metric takes the sum of depths of all paths divided by the total number of paths. The authors claim that the proposed metrics help in better understanding about ontology structures and approximate the cost and maintenance of an ontology. Research [36] focuses on the reusability of the ontologies. The research studied methods and tools for reusing ontologies. The research focuses on the point that the ontology should be shared and reused. The author claims that sharing and reusing ontologies increase the quality of the applications applying these two principles. That is, these applications become interoperable because they are provided with machine-processable and commonly agreed-upon understanding about the domain of interest. Furthermore, as in other engineering disciplines, reusing one ontology can reduce the costs of ontology
development. The reuse of ontology components avoids reimplementation of these components. Reusing, of course, is not straightforward procedure. However, modest modification of an ontology that will be reused is expected. Also, reuse as a technique can improve the quality of the reused ontologies. These reused ontologies would be checked frequently and evaluated by various parties through reuse. The researcher adds that ontologies rarely become application-independent and seldom build with the expectation of being shared or reused. Moreover, the researcher claims that the Semantic Web field is under development, meaning that the field’s theories, methods, and technical aspects are still in the tuning stage. Reusability is an important factor with respect to the Semantic Web and ontologies and the ability to interlink humans and machines and reach the vision of agreed ontologies. Also, this research provided some case studies on ontology reuse. The conclusion was that current ontology engineering tasks need to emphasize and adopt the reusability of existing ontological sources. This implies propose methodologies for reuse strategies; methods and tools for evaluation, customization, and integration; and finding a way to combine human and computational intelligence. The author in [12] mentioned that fast-changing environments lead to outdated ontology and therefore outdated application of the Semantic Web. When the author defines the four main phases of ontology creation and especially in the prune phase, he mentions the balance between the completeness and the incompleteness of the domain model, saying, “It is widely held belief that targeting completeness for the domain model appears to be practically inmanageable.” Research paper [19], advises that the ontology should be frequently maintained to reflect any specification of the ontology. Now, any update on
the environment should be reflected on the ontology so the ontology becomes more precise and consistent.

Different from [32][33], the proposed work is considering classes, subclasses relationships, and properties relationships as matrix. That is, the fundamental different from the above mentioned research papers is that using this matrix of major ontology pillars within an ontology, the effort of modifying an ontology can be estimated. Furthermore, one metric is not giving an accurate understanding about a given ontology. These three elements are considered as they relatively influence complexity of an ontology as schema. It focuses on providing a better understanding about ontologies. It countenances the idea of that an ontology should shared and reused proposed by [36]. An ontology that is not updated would lead to poor Semantic Web applications. Consequently, it is very helpful to scale the ontology by simple and easy-to-use approach for better understanding about an ontology. Such an approach can assist in estimating ontology modification effort, tracking ontology development process, and understanding consequences.

2.6 Motivation to the Proposed Work

Web 3.0 is the advanced version of the Web and the newest version at the chain of the Web development. The Semantic Web or Web 3.0 is a collection of components that work together so that a machine is able to process and understand information, i.e., the semantics of a website is understandable by a computer. In order for this vision to be realized, formal standards for representing and interpreting data must be formulated and
imposed. These formal standards include the Resource Description Framework (RDF) and machine understandable ontologies Ontology Web Language (OWL). Indeed, the Semantic Web vision provides the current Web with necessary infrastructure that allows computer to process knowledge and one of the cornerstones of this infrastructure is the Ontology. In this dissertation, a novel approach that is domain independent and emulates the nature diversity of the Web is present. Some available advantages on these generations (sitemaps) are recruited in order to contribute in this extension for Web 3.0 and future generations. This work does not contemplate the layout of the webpages. Also, the nature of architecture of our problem is different. In here, an approach that matches the nature of HTML sitemaps is proposed. Besides, the proposed work is extracting all potential relations between instances. This work is capable to extract concepts, taxonomic relations, instances, and non-taxonomic relationships. Furthermore, the proposed work is considering NLP that is substantial part of it. On the one hand, crafting an ontology manually is a difficult, time-consuming, and labor-intensive task. On the other hand, an automatically generated ontology may suffer from poor quality semantics. A semi-automated approach in ontology development is proposed. Such an approach can limit the disadvantages of each method when used in isolation. Many current websites have sitemaps which have taxonomic and other semantic information, especially, when related sitemaps are evaluated together. This research focuses on mining semantic information from sitemaps and using this information to create ontologies. Several well-known ontology learning tasks have been studied. Also, a review of similar work which also uses semantics in existing websites to help generate ontologies has been made. The current
Web itself has rich semantics, though not in a machine-readable format, and an important part of the current Web is sitemaps. HTML sitemaps are useful in helping users to easily navigate websites, and they have rich taxonomic information. Thus, sitemaps could be of significant value in ontology development. The approach evaluates related sitemaps by focusing on mining semantic information from sitemaps and then using this information to help create ontologies. Given a set of HTML sitemaps from one domain, the proposed approach will first learn taxonomic relations to start building an ontology and then enhance it by enrichment and refinement. The approach have evaluated on several domains. The empirical experiments indicate that the proposed approach is effective.

Indeed, ontology allows sharing structure understanding of the knowledge among human or application agents and enabling reuse of domain knowledge[15]. For example, Web Ontology Language (OWL) defines the <owl:import> that allows to reuse a desired content of another ontology. In fact, share and reuse of knowledge is an important goal of developing an ontology. Ontologies, in some task such ontology mapping and modularizing, are reused to support the notion the knowledge share and reuse. Ontologies are expected to be modified in order to be accommodated with added, removed or refined knowledge that allow fulfilling new requirements. Otherwise, they become outdated and irrelevant that led to poor Semantic Web application [12]. Thus, when an ontology is about to be built or chosen, it is worthwhile to make sure that this ontology will be easily modified by other parties in order to suit new tasks. Given that, there is a demand to measure the cost of understanding about an ontology that allow user to better reveal the
cost associated with modifying it. The ultimate objective of this part of the dissertation is to propose three dimensional ontology modification matrix that is generic scale for testing out an ontology in order to better understand and modify it. Number of research papers, that proposed some metrics in regard to understand about some schema characteristics, have surveyed. An empirical analysis on the proposed three dimensional ontology modification matrix has performed. That includes evaluating on real world ontologies to validate and show the usefulness and effectiveness of the proposed work. The results indicate that the proposed work is very promising and competitive. Moreover, some applications for the proposed work have noted. As matter of fact, ontology will change. Thus, maintaining an ontology periodically will help in tracing the development of an ontology. Altogether, this work is expected to allow an ontology to be reused that help in making the vision of the Semantic Web closer.

Furthermore in this dissertation, a study of the Web evolution is provided in response to understand the trend of the development and main characteristics, advantages and shortcomings. The World Wide Web, or simply the Web, is considered one of the main sources for in accessing information. Over the last decades, a number of improvements have been achieved that helped the Web reach its current state. For many, the World Wide Web has become indispensable to their daily lives. There are a number of research projects going on to enhance the current status and develop the future of the Web. It is therefore important to look into newer versions of the Web in order to improve the way that information is expressed to make more intelligent choices and obtain a better
meaning of the information over the Web. The evolution of the Web from Web 1.0, Web 2.0, Web 3.0, Web 4.0, to Web 5.0 is studied. The survey is pointing out documents types and technologies employed to understand the changes from Web 1.0 to Web 3.0 and to predicate future Web (Web 4.0 and Web 5.0).
2.7 Summary

In this chapter, the potential sources that an ontology creator should take into consideration for reuse have been presented. Also, an introduction to ontology learning including types of data documents and the types of ontology learning methodologies has been presented. Furthermore, techniques for creating ontologies have been addressed. In terms of a rule of thumb in ontology development, there is no one correct way to develop an ontology. Moreover, it is clear that the community shows some agreement on the general bold lines that are extracting concepts and taxonomic relationships, extracting the nontaxonomic relationships, and finally maintaining or refining the ontology. Also in this chapter, several research papers have been discussed. The literature review showed that the proposed work is different from all of the mentioned research papers. In this first of this dissertation, the proposed work take whatever advantages are available in an HTML sitemap. It has always been the case that the Semantic Web is an extension of the current web. And based on that, building the ontology should be somehow extended on part of the current web. Because ontologies are considered to be backbone of the Semantic Web, in this research a three dimensions ontology modification matrix approach is proposed. Such an approach can assist in relatively understanding the cost of ontology modification and tracking ontologies development processes as an attempt of relatively indicating the complexity of an ontology.
CHAPTER 3

ENVIRONMENT DESCRIPTION

The current state of the information over the Web, however, requires more formal standards to make it suitable for precise automatic semantic application. In this chapter, a spotlight is focused over the environment that proposed work dealt.

3.1 HyperText Markup Language (HTML)

HTML is the publishing language of the World Wide Web (WWW)\(^8\). It is a markup language that consists of markup tags that have fixed structure and are processed together. These tags together form HTML documents. These documents are read and interpreted by some tools, e.g., web browsers. Each tag of HTML tags has a special interpretation. For example, “<B> What</B>” will be interpreted as bold What like “**What**” and “<I> What </I>” will be interpreted as italic what like “*What.*” These tags have the power to embed images, text, … etc, as figure 3.2 below shows the source HTML document and in figure 3.1 shows the output of the HTML document

\(^8\) http://www.w3.org/html
<html>
<head>
  <title>Hello World</title>
</head>
<body>
  <B>Hello World</B>  
  <I>Hello World</I>
  <br>
  <img src="Sandy.jpg" style="width:304px;height:228px">
</body>
</html>

Figure 3.1: HTML output page

Figure 3.2: HTML source page
3.1.1 Processing HTML Document

Processing HTML documents refers to decomposing these documents in order to process it as set of HTML tags. This processing allows for further revealing information embedded between these tags. Processing may reveal texts, images, videos, sounds, links, read inputs … etc. Basically, any data presented on a web browser can be extracted by a HTML documents parser. An HTML document is an example of semi-structured document. In this type of document, the data are structured, at least, in part, based on semantics, but the underlying structure and the semantics are not explicitly given. The structure is still helpful in extracting the semantics from the document. This kind of document, if carefully handled, can produce relatively rich semantic information because the semantics are embedded in data and the data’s structure. Examples of semi-structured documents are HTML documents, WordNet [3], and XML documents. An example of an HTML parser is HtmlAgilityPack that is a .NET code library used to process HTML documents.

3.2 Searching Paradigms

On the web, there are vast amounts of websites, multiple domains, different languages, and different contents with high redundancy. Thus, it is a very difficult task to locate one website without knowing the exact URL. The Web has become a primary source of information. However, this information, regardless its size, is neither classified

http://htmlagilitypack.codeplex.com/
nor easily accessible. Thus, it either needs to be listed on a web directory and this allows users to navigate through a hierarchy to reach a desired website. Or this website be accessible for a search engine to be indexed and then located using a keyword-based searching paradigm. If the website is not listed on a web directory or not accessible for a search engine, it goes to what is called the deep web [37].

3.2.1 Keyword-based Search Paradigm

The keyword-based searching technique works with what is called a search engine. Search engines index the webpages data in a database. And whenever there is a query about a certain keyword, it searches these databases and returns to the user the most updated version of the webpage that contains that keyword. It does not differentiate domains nor content as long as the keyword is there. It results in getting more general hits. Then, the user evaluates these websites and avoids unrelated ones to reach what he or she is looking for. This means keyword-based searches are more general and require more overhead. For example, in figure 3.3 below, using the most popular search engine Google, the search for “Barack Obama” returned 233,000,000 results as Figure 3.3 below shows.
Figure 3.3: Number of results for single search

3.2.2 Directory-based Search Paradigm

A web directory lists websites by domains via hierarchal representation. It works like a Yellow Book but for websites. Thus, it is a human responsibility to find the website that they are looking for. Usually, web directories allow websites to be submitted. These websites within a web directory are tested and maintained by editors. An example of a web directory is shown below in Figure 3.4.

Figure 3.4: Web directory shows category and subcategory for some domains
Web directories such as Yahoo web directory\textsuperscript{10} or Open Directory Project\textsuperscript{11} are manually maintained, thus they have high accuracy in that related websites are grouped together. They are free, a large number of people maintain it, hierarchically arranged by subject - from broad to specific, and the user simply submits it by himself. It is advised that in submitting a website to choose the category that should reflect the content on the submitted website. The editors have the authority to move some websites to a different category, or maybe create a new one. Sometimes, going with a paid web directory is preferable because it is updated, free of spam, and contains active websites. Web portals usually groups related concepts into domains. Thus, taking advantage of already grouped content would result in connected related content which is preferable\textsuperscript{12}.

3.2.3 Direct Navigation

It is a method that is normally used when the user remembers the exact URL for the website [38]. For example, a student at Kent State University can directly write into the web browser address bar www.Kent.edu. It is a simple method for immediate access. However, it is not very helpful if the user wanted to access a specific webpage. As in the

\textsuperscript{10} https://dir.yahoo.com

\textsuperscript{11} http://www.dmoz.org/

\textsuperscript{12} http://www.w3.org/TR/webont-req/#section-use-cases
previous example, the student may not find it easy and straightforward to access the bursar office webpage at the Kent State University website.

3.3 Type of Sitemaps

Sitemaps were invented to allow easy access to the website by web crawlers or users, e.g. an XML sitemap. It is “A sitemap is a file where you can list the web pages of your site to tell Google and other search engines about the organization of your site content. Search engine web crawlers like Googlebot read this file to more intelligently crawl your site.” [39] Sometimes, search engines refer the user to a HTML sitemap if there is some error in finding a requested query as in Figure 3.5 below.

![Figure 3.5: Google referring the user to HTML sitemap](image)

An HTML Sitemap, in its simple meaning, is a webpage designed by the web owner or webmaster to help the visitor navigate the website in a smooth and related
manner. When navigating the web, it is preferable to guide users to find the information that they are looking for.

In general, sitemaps other than XML sitemaps might include navigation methodologies, a menu, or a hierarchy structure. Its main advantages or uses are in showing the website’s main structure and various links to that structure. Some argue that following the hierarchy navigation is more suitable for the adult navigator. Some websites do not apply the sitemap on their website and prefer the local search. Some fall in between, arguing that some content, e.g., adult content, should be accessible through navigation links and not accessible through a local search. Some researchers think the cultural aspect is playing a role in navigation of the website. Others claim gender may have a role in this game in terms of the way of thinking, the security aspect, preferred color, … etc [40][41][42][38].

Sitemaps come in a number of types [43][44][45]. They are XML sitemap, HTML Sitemap, Images Sitemap, Video Sitemap, News Sitemap, ROR Sitemap, Mobile Sitemap, and Text Sitemap.

### 3.3.1 XML Sitemaps

Starting with a XML sitemap, as figure 3.6 below shows, it is basically a file that holds a list of the hyperlinks within a given website. The XML sitemap is for a machine and not for a human to read or navigate. Also, within that text file, it is possible to
include additional information about each hyperlink, e.g., date of last update. The robots.txt files can be used in order to allow search engines to index a website. It is under the name of XML because it follows XML (eXtensible Markup Language) scripting language standards [38][46].

![XML sitemap example](image)

Figure 3.6: XML sitemap example

3.3.2 HTML Sitemaps

HTML sitemap is a webpage that is designed by a website owner or webmaster to help visitors navigate the website in smooth and related manner. As stated by [43] “Typically the traditional sitemap breaks down the referenced pages into groupings for easy reading. This kind of sitemap is generally designed to assist humans in navigating”.

60
An HTML sitemap, as figure 3.7 below shows, is about having the hyperlinks list under an HTML representation so that it gives more ability to be read. It should reflect the actual semantics of its website. It is a common practice that the web master is responsible for building a related and consistent website. HTML Sitemap works as a table of contents to a website that users can walk through to easily find the desired content. HTML sitemaps should have all the webpages within a website in hierarchy grouping.

Figure 3.7: Example of HTML sitemap captured from (www.tigerdirect.com)

### 3.3.3 Media Sitemaps

Media sitemaps concern with Images Sitemap, Video Sitemap, and News Sitemap [47][48][44]. All of them follow the same idea. They break down the hyperlinks and add some content or metadata to describe their content so that the search engine can locate them easily and help the owner to edit them easily, too.
3.3.4 ROR Sitemaps

ROR Sitemap is an extension to XML sitemap to have more content in the sitemap than just hyperlinks. ROR stands for (Resources of a Resource). It is mainly for search engines [49].

3.3.5 Mobile Sitemaps

On the mobile sitemap, it is just a version of the normal XML sitemap to support mobile version of the owner website. It basically leads to a mobile version of webpage [50].

3.3.6 Text Sitemaps

Finally, there is a text sitemap, which is very simple and holds only the hyperlinks or URLs for a website. In fact, it is an old version of the XML sitemap [51].

3.4 Text Similarity Matching

Texts similarities are automatic techniques to find out how two texts are similar in terms of meaning. It is part of Natural Language Processing (NLP). NLP, as defined by [52], it is “an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things”. NLP has a number of tasks such as: machine translation, natural language text processing and summarization, speech recognition, artificial intelligence, and expert systems. One of
NLP applications is WordNet [3] discussed in next section. Last, an introduction about WordNet.NET framework is presented.

3.4.1 WordNet

WordNet, as defined by [53], is a “lexical database … which was designed to establish the connections between four types of Parts of Speech (POS) - noun, verb, adjective, and adverb. The smallest unit in a WordNet is synset, which represents a specific meaning of a word”. Thus, it is a large lexical database proposed in English language aimed to network semantically related words and concepts. It includes total of 155287 words and 117659 synsets. Parts of speech including (nouns, verbs, adjectives and adverbs) are collected into (synsets) that are linked together via conceptual-semantic and lexical relations. The results is a semantic meaningfully net that relate words [54].

3.4.2 WordNet.NET

WordNet.NET [53] is an open source framework that is available on a C# .NET library environment. By providing two sentences and using semantic similarity steps, it is possible to measure the similarity of the meaning of these two sentences. High result between two sentences is meant to be they are more semantically related. Semantic similarity steps, that are followed by measuring similarity between two sentences [53], include partitioning sentences into tokens, part-of-speech (POS) disambiguation, stemming words, disambiguate word sense, creating semantic similarity relative matrix.
between each word of the two sentences, then finally, measuring the similarity of the two input sentences based on the similarity of the pairs of words. The proposed work used a Matching Average (MA) function in computing the overall similarity score between two sentences.

\[
\frac{2 \times \text{Match}(X, Y)}{|X| + |Y|}
\]

MA function to compute similarity score

Let us say there are two sentences X and Y. Matching (X, Y) are the results of matching tokens between X and Y. Then, sum up all candidates’ matching result. Then, it is divided by the sum of these tokens. For example, let us say there are two sentences. Sentence X = “Defense Ministry” and sentence Y = “Department of defense”. Note in this example X[1], X[2], Y[1], and Y[2] are “Defense”, “Ministry”, “Department”, and “defense”, respectively.

X[1] and Y[1] = .83
X[1] and Y[2] = 1
X[2] and Y[1] = .91
X[2] and Y[2] = .8
Using MA function to compute the overall score, 

\[ \text{Score} = \frac{2 (.91+1)}{4} = \frac{3.84}{4} = .955 \]

Thus, the semantic similarity between two sentences (Defense Ministry) AND (Department of defense) is 0.955

This score is used as threshold. Thus, the user can set the threshold to determine how acceptable the meaning is. It is important to note that the score of the semantic measurement should set to a threshold. Raising the level of the threshold would result in fewer relations. Lowering the threshold would open the door for a more relations that may not be semantic. Below are sample examples of the used framework.

- The semantic similarity between two sentences (Tom is a doctor) AND (tom is a teacher) is 0.855
- The semantic similarity between two sentences (He lay on shore) AND (He sit at bank) is 0.765
- The semantic similarity between two sentences (book room) AND (reserve suite) is 0.93
3.5 Ontology Development

This work is concerned with the availability of the information from web generations in order to contribute linking it with the advanced web generations. In some sense, ontology should be easily modified from the stage of building it to the stage of using it and ensure its permanence. In practice, one domain may have different ontologies. And one ontology can be used for number of applications. For example, let us say there is a need for an application in restaurant domain. Then, an ontology for restaurant menu pricing and another ontology for restaurant food services can be built. Both of them are from one domain. However, these ontologies have to share some well-defined understanding context that facilitates ontology development tasks. Managing ontologies, as in other contexts, e.g., software engineering and knowledge engineering, have a number of development procedures that are discussed on the next section [14][9][19][12]. Ontology development includes some activities such as Ontology Learning, Ontology Mapping, Ontology Evaluation, and Ontology Maintenance. First, Ontology is considered to be formal. That formalization needs to be understood by everyone working with it. Examples of formal language are RDF\textsuperscript{13} and OWL\textsuperscript{14}. In here, a brief introduction of the major ontology developments tasks is presented.

\textsuperscript{13} http://www.w3.org/RDF/

\textsuperscript{14} http://www.w3.org/TR/owl2-overview/
3.5.1 Ontology Learning

It aims to create ontology by learning from many sources. It is preferable to be either an automatic or semi-automatic approach. Now, there are good reasons to avoid learning the ontology manually or constructing it manually. That is, constructing ontology manually is expensive. It is expensive in terms of being time-consuming and labor-intensive. However, it is semantically rich since it is manually crafted [13]. Thus, approaches of automatic or semi-automatic are demanded and each of them has its own specifications and engineering methodologies.

3.5.2 Ontology Mapping

It concerns with aligning two or more ontologies. Indeed, ontology aimed to offer shared understanding to be used by distributed application and ontologies. To facilitate this sharing, some ontologies may be required to be mapped with other ontologies. Thus, ontology mapping aim to get over the disadvantages and restrictions that individual ontology may offer such as concepts limitation and interpretation. There are number of approaches proposed for mapping ontologies [55][56]. They are linguistic, statistical, structural and logical methods [9]. The linguistics approach tries to take advantage of the concept labels found on two ontologies and then maps them together. The statistical approach uses some functions or methods to measure the correlation among concepts in order to determine the relationships between one and another. The structural approach is focused on the internal structure of the ontology trying to map that internal graph between two ontologies. Logical methods are concerned with the relations/logic inside
ontologies and how to map them to other ontology. Ontologies, in the first place, represent a formal specification that holds the semantic. It is important to note that ontology mapping is a challenging area in the field of the Semantic Web.

3.5.3 Ontology Evaluation

It is a procedure to assess and revise an ontology during some conditions such as throughout developing an ontology [32][19]. It concerns with testing the ontology if it works and gives the expected results. Ontology evaluation allows avoiding inconsistent or wrong content within a given ontology [57]. Thus, one dimension of the evaluation is whether it met the intended semantic or not. In this context, there is no one method to evaluate all ontologies in the world. It differs from one domain to another or one application to another.

3.5.4 Ontology Maintenance

Ontology maintenance concerns with upgrade knowledge within an ontology to generate more accurate and reliable ontology. Updating ontology is normal procedure in order to insert/edit/delete concept or relationship. That is, ontologies are expected to be enriched or adjusted. Thus, ontologies have to be maintained regularly [19]. For example, in public service ontology, some rules or regulations may be different from state to another state. Another example, a company in expanding stage would require more concepts and properties to model this expanding. In this context, Web is a dynamic environment by its nature. Maintaining this nature is an urgent task in the field of the Semantic Web [9].
3.6 Summary

The environment that this dissertation is dealing with has been presented. That is, in order to automate some of learning procedures within an ontology, this work is exploring and understanding the nature of the Web. In this chapter, the environment that proposed work dealt with including Hyper Text Markup Language (HTML), searching paradigms, types of sitemaps, and text similarity and the framework used has been reviewed and briefly described. Moreover, formal standards, such as Ontology, are required in order for machine to semantically process information. In this chapter, since the ontology is essential part of the Semantic Web and part of the dissertation is ensuring its permanence, major ontology development tasks and definitions have been briefly reviewed.
CHAPTER 4

Using the Semantics Inherent in Sitemaps to Learn Ontologies

The Semantic Web is a collection of components that work together so that a machine is able to process and understand information. In order for this vision to be fully realized, formal standards for representing and interpreting data must be determined and enforced. These formal standards include the Resource Description Framework (RDF) and machine processible ontologies. On the one hand, crafting an ontology manually is a difficult, time-consuming, and labor-intensive task. On the other hand, an automatically generated ontology may suffer from poor quality, including poor semantics. Thus, a semi-automated approach in ontology development is proposed. An approach that limits the disadvantages of each above mentioned method when used in isolation. The Semantic Web is an extension of the current web. The current Web itself has rich semantics, though not in a machine-readable format, and an important part of the current Web is sitemaps. HTML sitemaps are useful in helping users to easily navigate websites, and they have rich taxonomic information. Thus, sitemaps could be of significant value in ontology development. The approach evaluates related sitemaps by focusing on mining semantic information from these sitemaps and then using this information to help create ontologies. Given a set of HTML sitemaps from one domain, the proposed approach will first identify taxonomic relations to start building an ontology and then enhance the ontology by enrichment and refinement. The approach has been evaluated on several domains. The empirical experiments indicate that the approach is effective. This chapter
has presented and published on 2014 IEEE 38th International Computer Software and Applications Conference Workshops.

4.1 Introduction

The Semantic Web is about making the Web understandable by the machines. Thus, it can produce better results and overcome the drawbacks of Web 2.0, e.g., the enormous number of hits for a single query and issues in web accessibilities [58]. The Semantic Web or Web 3.0 aims to get the precise piece of information in a simpler and faster way. It is about adding formal semantics to the data on the web. Web 3.0 is considered to be an extension to the current version of the web, while not replacing it [4]. Reaching the Semantic Web vision requires creating formal standards for the semantics. These standards will enable machines to understand information and to help in browsing, searching, and organizing the information. One of these formal standards is ontologies.

An ontology is a “formal, explicit specification of a shared conceptualization” [7]. It aims to provide a formal domain of understanding to machines and humans. It is considered to be the backbone of the Semantic Web in that it relates global domain content in a formal semantics context.

Learning ontology refers to creating ontologies from one or more sources. Naturally, the sources themselves heavily affect the learning procedures. Input sources are collected in three categories [13]. They are structured, semi-structured, and
unstructured documents. Ontology learning methodologies are categorized as automatic, semi-automatic, and manual learning methodologies. These methodologies differ in terms of the amount of automatic development of the ontology versus the amount of human intervention [14][13]. Manual learning is expensive in terms of time, labor, and effort; however, it is semantically rich because it is manually crafted [9]. Automatic learning procedures are fully automated; there is no human intervention at all. According to [12] the semi-automatic learning approach is preferable for ontology learning. Because it takes the advantage of both automatic and manual approaches, it partially resolves the disadvantages of both. It can apply the machine learning techniques used on knowledge acquisition, extraction, revision or maintenance [9]. Also, it is guided by humans; thus, the quality of the work is on a higher level [16]. In the proposed system, a semi-automatic approach is implemented using HTML web documents. HTML documents are semi-structured documents. This kind of document, if carefully handled, can produce relatively rich semantic information because the semantics are embedded in the data and the data structures. This kind of document has been used in order to develop ontologies from HTML sitemaps.
Web documents have been used to learn ontologies as in [23][20]. In fact, HTML sitemaps are stages where the approach is not starting building from scratch. Also, HTML sitemaps have semantics. Figure 4.1 shows an example of a HTML sitemap. From figure 4.1, it can be clearly seen that “Home Improvements” is divided into five tasks including “Interior Home Remodeling” and “Exterior Improvements”. Then “Interior Home Remodeling” includes “Kitchen” and “Lighting”. The HTML sitemaps for websites work similarly to the tables of content in books. HTML sitemaps are often defined based on underlying semantic relationships. For example, as figure 4.1 shows, it relates topics and explicitly isolates non-related content.

Figure 4.1: Example of a HTML sitemap
When observing related domain revised HTML sitemaps, it seems that there is a high level of agreement when it comes to semantic content and also they complement each others. For example, in auto parts domain, it can be seen that Cooling and Heating, Brakes, Exhaust System, Transmission, Drivetrains, and Suspension and Steering … etc as figures 4.2, 4.3, and 4.4, below shows revised version of HTML sitemap from AutoZone (website#1), AdvancedAutoPart (website#2), and NAPA (website#3) auto parts websites, respectively. The proposed approach focuses on using the semantics from the HTML sitemaps to learn ontologies.

![Diagram](image.png)

Figure 4.2: Revised HTML sitemap from website#1
Figure 4.3: Revised HTML sitemap from website#2

Figure 4.4: Revised HTML sitemap from website#3
4.2 Methodology

Websites from the same domain area tend to express their HTML sitemaps’ content in semantically similar ways. On the one hand, a single HTML sitemap can reveal taxonomical relationships. On the other hand, the approach grouped HTML sitemaps that reveal domain concepts. Also, these related concepts may be linked together in a semantically structured ontology making it possible to connect knowledge, rather than just connecting information. The proposed approach follows incremental ontology updates or a snowball methodology technique. That is, each step is based on a previous step. It is a domain independent approach. It is learning domain concept, taxonomy relations, and non-taxonomy relations to design a domain specific ontology by means of the HTML sitemaps.

It is important to note that semantic is everywhere whether it is implicitly or explicitly expressed or whether it is noticeable or not. Bootstrapping HTML sitemaps semantic structure helps reduce the effort of discovering semantics later. In fact, the approach is considered to be reusing existing ontologies because HTML sitemaps may be considered to be very lightweight ontologies. More time spent utilizing the sitemaps means less time spent in making the ontology ready to use. The approach is divided into two phases. They are the taxonomic phase and the enhancement phase as shown in figure 4.5 below.
Taxonomic Phase

Taxonomic extraction starts with selecting an HTML sitemap for a website. HTML sitemaps depict explicit relationships that the proposed work can use as a good starting point as can be seen on figure 4.1. The website is to be chosen from web-directories, e.g., The Open Directory Project (ODP). OPD is a source for domain related websites. It is noteworthy that the websites that belong to web-directory are manually maintained, and thus, there is a high degree of accuracy in the way the related websites
are grouped together. For example, in OPD, subjects are arranged from broad to specific, apply a peer review process, and decisions are made after consulting with category’s editors.

4.2.1.1 Input Preprocessing

The source document structure is studied, which in this case, is an HTML sitemap. It is considered to be a preprocessing input document. There are number of methods that can help to build an HTML sitemap. However, most of them are based on the physical implementation of the websites e.g. folder and subfolder structure. Some websites that have an HTML sitemap have built their HTML sitemap via HTML table tags, tab/bullet spaces, or <UL> <OL> <LI> HTML tags. In order to automatically handle this diversity, the proposed approach will mimic one standard methodology. The proposed work applies the methodology that is followed as the standard way for most web designers and official websites. It is the methodology that applies <UL> <OL> <LI> HTML tags, which are, in our case, Basic Mode (BM). If a sitemap is not available, it can be created manually using one of the above mentioned methods or automatically by using an online sitemap generator as in [59]. A sitemap needs to meet certain requirements in order to be ready as input for the proposed system. The sitemap’s links must have appropriate domain-related titles with a modest number of meaningful words (avoid non-understandable or product-specific terms), and the sitemap must apply the <UL> or <OL> HTML tag for grouping titles and the <LI> HTML tag for listing titles. If a
sitemap fails to meet these minimum requirements, then it needs to be modified for the system to use it. Since using HTML sitemaps is our focus, the proposed system helps in making the HTML sitemaps follow system requirement through an Advanced Mode (AM). It requires HTML sitemap as input that can be automatically generated via an HTML sitemap generator mentioned in [59]. In case of an HTML sitemap have automatically generated, there may be a chance that it may need to be revisited via AM. After that, the HTML sitemap is ready to be processed by the system.

First, the system asks the user to enter HTML sitemap link (it can be uploaded from local device or from website over the internet). User also required to choose the type of the algorithm to be used. Algorithm (1) for Basic Mode (BM) and algorithm (2) for Advanced Mode (AM) as figure 4.6 below shows:

![Proposed system interface](image)

Figure 4.6: Proposed system interface.
Let says the system have the below HTML sitemap as an input on Algorithm (2):

- About
- Client Feedback
- Contact
- Services
- Portfolio
  - Residential Architecture, Santa Fe
    - Contemporary Designs
    - Traditional Architecture
    - Sustainable Architecture
  - Commercial Architecture, Santa Fe
    - Fire Station Designs
    - Hospitality Architecture
    - Industrial + Warehouse Designs
    - Medical Architecture
    - Museum Architecture
    - Office Designs
    - Restaurant Designs
    - Retail Architecture

The output would be shown to the user as the figure 4.7 below:

![Figure 4.7: Output of advanced mode](image_url)
AM helps websites to express their semantics and give an overview of the website’s content. This may include renaming titles, removing titles, or rearranging the sitemap’s hierarchy. In arranging the sitemap, the AM builds the sitemap’s hierarchy by implementing an adjacency list model [60][61][62][63]. After the HTML sitemap is created, the system makes sure that the HTML sitemap links have titles. At this step, it checks the submitted sitemap for Sitemap Usefulness (SiU). This requires titles, and if the sitemap does not have titles, the system cannot use it. Each sitemap that is to be added to the system must show the ratio of usefulness as in the formula (1) below,

\[
SiU = \frac{\text{Titles’ Links}}{\text{Total Links}}
\]

This ratio is calculated by dividing the number of all the sitemap links that have titles (Titles’ Links) over the number of all sitemap links (Total Links). Sitemap Usefulness produces a ratio from 0 to 1, where 0 means that the sitemap is not useful for the system, and 1 means that the sitemap is potentially very useful. Next, the system requires meaningful titles. Thus, it check the how far the entered HTML sitemap can be semantically utilized. A sitemap may include titles that are Void Titles (VT), which are not semantically related to a domain such as blog, feedback, or welcome. Thus, the system checks for Semantic Utilization (SeU) as in formula (2) below,
\[ SeU = \frac{\text{Meaningful Titles}}{\text{Titles' Links}} \]

In formula (2), \textit{Meaningful Titles} is the result of subtracting \textit{Void Titles} (VT) from \textit{Titles' Links}. Some of these VTs are pre-defined in the proposed system. Also, they may be updated by listing or removing any desired VT. Also, they can be removed via AM. The ratio will be calculated by dividing the number of \textit{Meaningful Titles} by the number of all \textit{Titles' Link} in the HTML sitemap. (SeU) produces a ratio from 0 to 1, where 0 means that the sitemap does not include domain related titles, and 1 means that all the titles are potentially semantically meaningful. For example, let us say the system got these sitemaps from architectural services domain:

- Home
- Why AEA?
  - Mission Statement
  - The Owner
  - History
- Solar
  - Solar Projects and Incentives
  - Solar Gallery
  - Arizona Solar
  - More Information
- Wind
  - Incentives
- Green Architecture
  - Il-Crete AAC
  - LEEDs and Pilot Home
- Services
  - Air Quality
    - Environmental Audits
    - Remediations
    - Resources
  - Green Construction
  - Water Harvesting
- Contact
- News

- About
- Client Feedback
- Contact
- Services
- Portfolio
  - Residential Architecture, Santa Fe
    - Contemporary Designs
    - Traditional Architecture
    - Sustainable Architecture
  - Commercial Architecture, Santa Fe
    - Fire Station Designs
    - Hospitality Architecture
    - Industrial + Warehouse Designs
    - Medical Architecture
    - Museum Architecture
    - Office Designs
    - Restaurant Designs
    - Retail Architecture

Figure 4.8: Example of number if HTML sitemap to show how to calculate SeU and SiU
SiU tells the user how far the usefulness of the submitted HTML sitemap is. In here, all of these sitemaps have titles, which means they are useful and are assigned a score of 1. Later, the system measures the Semantic Usefulness (SeU) of each of these sitemaps. From the sitemaps above, the SeU from left to right, are 0.875, 0.88, and 0.6. The function below shows how the system reads user inputs and counts SiU:

```csharp
foreach “href” node
{
    count++
    if (node.InnerText is not empty)
    {
        titles++
    }
}
SiU = titles/count
```

The system is built based on three Databases. They are: “newSitemaps” for buffering brand new HTML sitemap, “newSitemapsID” for storing what already stored processed HTML sitemaps, and “tempSitemapsID” for temporarily processing new HTML sitemaps before merging with old HTML sitemaps. They follow adjacency list model that each node in the table store a pointer to its parent. For example, the table below stores data from figure 4.1 above.
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Parent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Home Improvement</td>
<td>NULL</td>
</tr>
<tr>
<td>2</td>
<td>Design</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Exterior improvements</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Hardware &amp; Tools</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>Interior Home Remodeling</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Household</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>Bedroom</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>Interior design</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>Roofing</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>Garden</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>Power tools</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>Construction</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>Kitchen</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>Lighting</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>Maintenance</td>
<td>6</td>
</tr>
<tr>
<td>16</td>
<td>Security</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 4.1: Adjacency list representation of figure 4.1

By the end of the previous step, system adds this sitemap to the “newSitemaps” database. In building an ontology for a specific domain, the system potentially processes many HTML sitemaps for websites in the domain. After processing one sitemap, the system can process additional sitemaps. When the system encounters a new title in a new sitemap, it needs to determine its semantic similarity to titles that the system already has encountered. To determine this similarity, an open source Natural Language Processing (NLP) tool called Measuring Similarity between Sentences is used [53]. The approach
assumes all titles are concept-candidates for the developing ontology, and it segments an HTML sitemap from its root as Figure 4.9 shows.

Figure 4.9: Segmentation of HTML sitemap from its root

The algorithm runs as following:
// The system reads the database “NewSitemaps” and puts them in
// “dictSitemap <string, string>” Dictionary data structure collection.
// Also, keep the ID of each entry in “sqlBuffer <string, int>” Dictionary data
// structure collection

while (Reading from NewSitemaps)
{
    sqlValue = parentLink
    sqlValue2 = parent
    parentID = ID
    if (sqlValue2 == "AddToRoot")
        {
            // The system inserts root values into the list “rlist”
            rlist.Add(sqlValue2)
        }
    sqlBuffer.Add(sqlValue2, parentID)
    dictSitemap.Add(sqlValue2, sqlValue)
} // end while

Then, the system checks if it is the first sitemap on “newSitemapsID” database to be
added or not, as function below shows:

if (!newSitemapsID.HasRows)
{
    for each valuepair on sqlBuffer dictionary
    {
        INSERT INTO newSitemapsID (valuepair.value, valuepair.key, sqlBuffer[dictSitemap[ valuepair.key]])
    }
}

Else, the system inserts a new sitemap into “tempSitemapsID” database and snippets
below show its dynamicity:
For each pairtest in sqlBuffer
{
    //Add “candidate-concept” directly if they are root values
    if (root list contains(pairtest.Key))
    {
        INSERT INTO tempSitemapsID (pairtest.Value, pairtest.Key)
    }
    else
    {
        INSERT INTO tempSitemapsID (pairtest.Value, pairtest.Key, sqlBuffer[dictSitemap[pairtest.Key]])
    }
}

Then, the system counts how many records are in “newSitemapsID” and store the result into a variable called “count”, and counts how many records are in “tempSitemapsID” and store the result into variable called “count2”. The system initials an array called “tempSM” to store new stored sitemap and initials another array called “jold” to store older stored sitemaps, as function below shows:
The approach tests similarity of the proposed HTML sitemaps segments in order. That is, it starts from first generation level and goes to the last generation level. The new segments from new HTML sitemap get tested to previously processed HTML segments. Also, the approach does try match proposed-concepts of a different generation’s levels. That means it tests each proposed-concept from recent HTML sitemap segment in the

```sql
// Read from temporary database by selecting
td1.parent as L1, td2.parent as L2, td3.parent as L3, td4.parent as L4, td5.parent as L5, td6.parent as L6, td7.parent as L7, td8.parent as L8
from tempSitemapsID as td1
left join tempSitemapsID as td2 on td2.parentLink = td1.ID
left join tempSitemapsID as td3 on td3.parentLink = td2.ID
left join tempSitemapsID as td4 on td4.parentLink = td3.ID
left join tempSitemapsID as td5 on td5.parentLink = td4.ID
left join tempSitemapsID as td6 on td6.parentLink = td5.ID
left join tempSitemapsID as td7 on td7.parentLink = td6.ID
left join tempSitemapsID as td8 on td8.parentLink = td7.ID
where td1.parent = 'AddToRoot'

// insert them appropriately into “tempSM”

// read from newSitemapsID database by selecting
nsd1.parent as L1, nsd2.parent as L2, nsd3.parent as L3, nsd4.parent as L4, nsd5.parent as L5, nsd6.parent as L6, nsd7.parent as L7, nsd8.parent as L8
from newSitemapsID as nsd1
left join newSitemapsID as nsd2 on nsd2.parentLink = nsd1.ID
left join newSitemapsID as nsd3 on nsd3.parentLink = nsd2.ID
left join newSitemapsID as nsd4 on nsd4.parentLink = nsd3.ID
left join newSitemapsID as nsd5 on nsd5.parentLink = nsd4.ID
left join newSitemapsID as nsd6 on nsd6.parentLink = nsd5.ID
left join newSitemapsID as nsd7 on nsd7.parentLink = nsd6.ID
left join newSitemapsID as nsd8 on nsd8.parentLink = nsd7.ID
where nsd1.parent = 'LOroot'

// then, insert them appropriately into “jold”
```
first generation to previously processed proposed-concepts HTML segments, and when
the best pair is found, it groups the concept-candidates. Else, matching proceeds to next
generation to previously processed proposed-sub-concepts. In this case, the matching
similarity result (threshold) must be >.95. Then, the approach starts testing each pair in
the same segment generation, and when the best pair is found, it groups the concept-
candidates. For example, in Figure 4.10, if \{G\} and \{C\} are the best pair, they are
grouped as \{G, C\}. Then similarities among their next generations, if there are next
generations, are tested, and matches, as appropriate, are made. Also, proposed-sub-
concepts may be matched only if their ancestors’ proposed-concepts have already been
matched. Thus, the approach only considers \{K\} and \{D\} or \{K\} and \{E\} because \{A\}
and \{M\} are matched.

Figure 4.10: Match proposed-concepts of different generations.

For example, let us say there are set of HTML sitemaps already processed and
stored on newSitemapsID. And let us then say there is a new HTML sitemap on
tempSitemapsID. At the time of merging them, let us assume newSitemapsID have these
processed proposed-concepts from HTML sitemap segments:
<table>
<thead>
<tr>
<th>ID</th>
<th>parent</th>
<th>parentLink</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOroot</td>
<td>NULL</td>
</tr>
<tr>
<td>2</td>
<td>Drive Shaft</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>Fluids &amp; Chemicals</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>Manual Transmissions</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>Transfer Case Parts</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>Transmission Gaskets</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>Charging System</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>Lights, Flasher Units, Fuses</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>Generators</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>Electrical Connectors &amp; Sockets</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>Coolant Reservoirs*Coolant Reservoirs</td>
<td>15</td>
</tr>
<tr>
<td>12</td>
<td>Fuel Systems*Fuel Systems</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>Exhaust system*Exhaust Systems</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>Transmissions &amp; Shifters</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>cooling systems*cooling systems</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>Batteries &amp; Electrical &amp; wiring</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>Intake Systems</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>Radiator<em>Radiator</em>Radiators</td>
<td>15</td>
</tr>
<tr>
<td>19</td>
<td>Radiator Fan Blades</td>
<td>18</td>
</tr>
<tr>
<td>20</td>
<td>Radiator Fan Clutch*Radiator Fan Clutch</td>
<td>18</td>
</tr>
<tr>
<td>21</td>
<td>Radiator Hardware</td>
<td>18</td>
</tr>
<tr>
<td>22</td>
<td>Radiator Hoses</td>
<td>18</td>
</tr>
<tr>
<td>23</td>
<td>Radiator Mounts*Radiator Mounts</td>
<td>18</td>
</tr>
<tr>
<td>24</td>
<td>Radiator Sensors</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 4.2: Example of processed HTML sitemaps at newSitemapsID database

This will be represented, at the time of processing, on “jold” multidimensional array as table 4.3 below shows.
<table>
<thead>
<tr>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOroot</td>
<td>Fuel Systems*Fuel Systems</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Exhaust system*Exhaust Systems</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Transmissions &amp; Shifters</td>
<td>Drive Shaft</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Transmissions &amp; Shifters</td>
<td>Fluids &amp; Chemicals</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Transmissions &amp; Shifters</td>
<td>Manual Transmissions</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Transmissions &amp; Shifters</td>
<td>Transfer Case Parts</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Transmissions &amp; Shifters</td>
<td>Transmission Gaskets</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Intake Systems</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Batteries &amp; Electrical &amp; wiring</td>
<td>Charging System</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Batteries &amp; Electrical &amp; wiring</td>
<td>Lights, Flasher Units, Fuses</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Batteries &amp; Electrical &amp; wiring</td>
<td>Generators</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Batteries &amp; Electrical &amp; wiring</td>
<td>Electrical Connectors &amp; Sockets</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Coolant Reservoirs*Coolant Reservoirs</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiator Hardware</td>
<td></td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Ttator</em>Radiator Hardware</td>
<td></td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Ttator</em>Radiator Hardware</td>
<td></td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Ttator</em>Radiator Hardware</td>
<td></td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Ttator</em>Radiator Hardware</td>
<td></td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Ttator</em>Radiator Hardware</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3: Example of newSitemapsID database at the time of processing

And the new HTML sitemap on tempSitemapsID is represented as on “tempSM” multidimensional array as table 4.4 below shows:
Then, system starts with matching segmented part as algorithms below shows:

```csharp
// preI: is an index on “tmpSM” array vertical dimension
// preJ: is an index on “jold” array vertical dimension
// preK: is an index on horizontal dimension
// globalK, globalPreI, globalPreJ, for keep track of right segment index
For preI = 0 to preI < count2
{
    max = 0
    for preK = 1 to preK < 8
    {
        for preJ = 0 to preJ < count
        {
            // split concept-candidates by “*” symbol
            String[] parentRead = jold[preJ, preK].Split("*")
            foreach word in parentRead
            {
                score = semsim.GetScore(word, tempSM[preI, 1])
                // semantic similarity must be .95 above
                if (score > .95 && score > max)
                {
                    globalK = preK
                    globalPreI = preI
                    globalPreJ = preJ
                    max = score
                }
            }//foreach word
        }//preJ
    }//preK
    if (max != 0)
    {
    }
}//preI
```

Table 4.4: Representation of the array at the time of the processing
The result of applying the above matching segment function is shown below on table 4.5:

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOroot</td>
<td>Fuel Systems*Fuel Systems</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Exhaust system*Exhaust Systems</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Transmissions &amp; Shifters</td>
<td>Drive Shaft</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Transmissions &amp; Shifters</td>
<td>Fluids &amp; Chemicals</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Transmissions &amp; Shifters</td>
<td>Manual Transmissions</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Transmissions &amp; Shifters</td>
<td>Transfer Case Parts</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Transmissions &amp; Shifters</td>
<td>Transmission Gaskets</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Intake Systems</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Batteries &amp; Electrical &amp; wiring</td>
<td>Charging System</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Batteries &amp; Electrical &amp; wiring</td>
<td>Lights, Flasher Units, Fuses</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Batteries &amp; Electrical &amp; wiring</td>
<td>Generators</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>Batteries &amp; Electrical &amp; wiring</td>
<td>Electrical Connectors &amp; Sockets</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Coolant Reservoirs*Coolant Reservoirs</td>
<td>NULL</td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiators*Radiators</td>
<td>Radiator Fan Clutch*Radiator Fan Clutch</td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiators*Radiators</td>
<td>Radiator Hardware</td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiators*Radiators</td>
<td>Radiator Hoses</td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiators*Radiators</td>
<td>Radiator Mounts*Radiator Mounts</td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiators*Radiators</td>
<td>Radiator Fan Blades</td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiators*Radiators</td>
<td>Radiator Sensors</td>
</tr>
<tr>
<td>LOroot</td>
<td>cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiators*Radiators</td>
<td>Radiator Hardware</td>
</tr>
</tbody>
</table>

Table 4.5: Representation of the array after processing matching segment function
The objectives are to preserve the semantics of each HTML sitemap, to bring the collective semantics of all domain sitemaps together, and to preserve what the domain community has agreed on, which is depicted in the “developing” ontology. As each HTML sitemap is “added” to the other sitemaps, the lightweight ontology will potentially be enriched. At each time of adding a new HTML sitemap, and after the segmentation part is done, there are two options. First, there are similarities between previously proceeded concepts and newly encountered concept-candidates. Or, there are no similarities between them. The algorithm below is an updated version HTML segmentation algorithm. Below is the sequence of steps that has been added to previous segmentation algorithm:

- Matching by locking horizontal dimension “K” for both arrays “jold” and “tempSM”
- If similarity score<.65
  - Then add this new concept-candidate
- Else
  - Store the indices of best match between jold and tempSM arrays
  - Store concept-candidate at Temp= bestMatchOfJold+bestOfTempSM
  - Get former generation of jold array “[globelJ, k - 1].Split(*)”
  - Get former generation of tempSM array “[globelI, k - 1].Split(*)”
    - If former generations are already matched
      - Then updated concept (replace bestOfJold by Temp)
    - Else Add bestOfTempSM as new concept
The result of adding segments on table 4.4 to segments on 4.5 will result in table 4.6.

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOroot Fuel Systems*Fuel Systems</td>
<td>NULL</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Exhaust system*Exhaust Systems</td>
<td>NULL</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Transmissions &amp; Shifters</td>
<td>Drive Shaft</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Transmissions &amp; Shifters</td>
<td>Fluids &amp; Chemicals</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Transmissions &amp; Shifters</td>
<td>Manual Transmissions</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Transmissions &amp; Shifters</td>
<td>Transfer Case Parts</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Transmissions &amp; Shifters</td>
<td>Transmission Case Gaskets</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Intake Systems</td>
<td>NULL</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Batteries &amp; Electrical &amp; wiring</td>
<td>Charging System</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Batteries &amp; Electrical &amp; wiring</td>
<td>Lights, Flasher Units, Fuses</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Batteries &amp; Electrical &amp; wiring</td>
<td>Generators</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Batteries &amp; Electrical &amp; wiring</td>
<td>Electrical Connectors</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot cooling systems*cooling systems</td>
<td>Coolant Reservoirs*Coolant Reservoirs</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiator*Radiators</td>
<td>Radiator Fan Clutch*Radiator Fan Clutch</td>
<td></td>
</tr>
<tr>
<td>LOroot cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiator*Radiators</td>
<td>Radiator Hardware</td>
<td></td>
</tr>
<tr>
<td>LOroot cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiator*Radiators</td>
<td>Radiator Hoses</td>
<td></td>
</tr>
<tr>
<td>LOroot cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiator*Radiators</td>
<td>Radiator Mounts*Radiator Mounts</td>
<td></td>
</tr>
<tr>
<td>LOroot cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiator*Radiators</td>
<td>Radiator Fan Blades</td>
<td></td>
</tr>
<tr>
<td>LOroot cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiator*Radiators</td>
<td>Radiator Sensors</td>
<td></td>
</tr>
<tr>
<td>LOroot cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiator*Radiators</td>
<td>Radiator Hardware</td>
<td></td>
</tr>
<tr>
<td>LOroot Brakes</td>
<td>Brake Calipers</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Brakes</td>
<td>Brake Pads &amp; Shoes</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot Brakes</td>
<td>Brake Drums &amp; Rotors</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>LOroot cooling systems*cooling systems</td>
<td>Radiator<em>Radiator</em>Radiator*Radiators</td>
<td>Radiator Overflow Hoses</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.6: Representation of the array after processing all segments function
In this process, the system will encounter a number of proposed-concepts that represent similar things, e.g., things that are semantically related. These semantically related concepts are grouped appropriately. The most repeated and related ones should be the fundamental domain concepts, and this will often be reflected in the titles on the sitemaps.

4.2.1.2 Semantic Similarity System

The proposed work uses an NLP [52] to measure the semantic similarity between two proposed words or sentences. The proposed work considers an open source [53] that is based on WordNet [3]; it performs number of steps in order to compute the semantic similarity between two sentences. For each sentences, the steps include partitioning sentences into tokens, tagging, stemming words, disambiguating word senses, and finally, computing the similarity between two sentences. Thus, between each pair of sentences, the system builds a relative matrix with an entry $R[m, w]$ for each pair of words. Suppose there are two sentences X and Y with m being X’s length and w being Y’s length. $R[i, j]$ is the semantic similarity between the most common sense of the words at the $i^{th}$ location of X and the $j^{th}$ location of Y. The system computes the “Matching Average” as previously explained on chapter 3.

Assume two sentences X and Y. The semantic similarity system divides the sum of the similarity results of all the best match candidates between X and Y by the total
number of tokens of X and Y. Assume X has length 3 and Y has length 4 and the results
of measuring similarities each pair are 0.6 between X[1] and Y[1], 0.7 between X[2] and
Y[4], and 0.7 between X[3] and Y[3]. Then, by using the Matching Average formula, \( S = \frac{(2 \times (0.6 + 0.7 + 0.7))/(3+4))}{5} = 0.57 \). The reader is referred to chapter 3 for more information about
measuring the semantic similarity between two sentences.

4.2.2 Enhancement Phase

After the concepts and taxonomic relations are defined, the next step is to see if
the taxonomic relations may be enriched with relations among concepts/instances, and
then the ontology moves to the refinement stage.

4.2.2.1 Enrichment

The input to this phase is taxonomic relationships with concepts/instances. First,
system identifies instances as the function below shows.

```csharp
// system identifies individuals.
// “individualsList” is a dictionary data structure with key for individual and value for parent.
// Read from newSitemapsID database by selecting
// Tb1.parent FROM newSitemapsID AS Tb1 LEFT JOIN newSitemapsID as Tb2
// ON Tb1.ID = Tb2.parentLink WHERE Tb2.ID IS NULL;
// while (thisReaderForFinal.Read()){
//     if (!individualsList.ContainsKey(thisReaderForFinal["parent"].ToString().Trim()))
//         individualsList.Add(thisReaderForFinal["parent"].ToString().Trim(), string.Empty)
// }```

Then, it tries to enrich the taxonomic relationships with non-taxonomic relationships. Between each instance of a concept (1) the semantic similarity between instances is tested, and (2) a potential candidate to relationship between tested pairs is proposed. A tool like [53] can generate the semantic similarity between them. The point here is that if there is a similarity between two pairs, there might be a relation between these two pairs. Eventually, there will be a list of potential relations that are proposed by the system for a domain expert to review them at the refinement stage next section. Non-taxonomic relationship discovery algorithm is below. On the algorithm, the array “all” contains all instances within the ontology. The “individualList” dictionary contains the concept of each instance.
for (int i = 0; i < all.Length; i++)
{
    for (int j = i+1 ; j < all.Length; j++)
    {
        //check if they belong to same concept
        if (individualsList[all[i]] == individualsList[all[j]])
            continue;

        String[] allI = all[i].Split('*');
        String[] allJ = all[j].Split('*');

        foreach (string itemI in allI)
        {
            foreach (string itemJ in allJ)
            {
                if (itemI == itemJ)
                    continue;
                if (dublicateItem == itemJ)
                    continue;
                else
                    dublicateItem = itemJ;

                score = semsim.GetScore(itemI, itemJ);

                if (score > .7)
                {
                    threshold7++;
                    //Append result to rsowl file in OWL/XML format
                    rsowl.WriteLine(" <ObjectPropertyAssertion>");
                    rsowl.WriteLine(" <ObjectProperty IRI="#Auto_generated+7"/>");
                    rsowl.WriteLine(" <NamedIndividual IRI="# + itemI.Replace(" ", "_") + ""/> ");
                    rsowl.WriteLine(" <NamedIndividual IRI="# + itemJ.Replace(" ", "_") + ""/> ");
                    rsowl.WriteLine(" </ObjectPropertyAssertion>");
                }
                else if (score > .6) { //do same }

                //complete checking on semantic similarities if score > .5, > .4, and > .3
            }
        }
    }
}
4.2.2.2 Refinement

The refinement phase emphasizes an expert point of view to test and evaluate the proposed output ontology. The expert’s task is to evaluate the ontology correctness by tracing each proposed relationships. To better visualize an ontology, it is suggested to use Protégé API editor rather than source file as figure 4.15 shows below. Protégé API is an open source, support OWL syntax, with various available plug-ins. Tracing helps in discovering unnecessary relationships. For example, the expert can look over entities from “entity” tab as figure 4.11 shows.
When the expert trace relationships, he/she may name or delete proposed relationships as figure 4.12 and 4.13 show. By the end of this phase, the output is the domain ontology that is based on HTML sitemaps of domain of interest.
Figure 4.12: Tracing proposed relationships using Protégé API
4.3 Experiment and Evaluation

The proposed work has been evaluated on number of domains including apparel, electronics, and architectural services with an average of 4 HTML sitemaps for each domain. The ontology gets evaluated after the refinement stage. HTML sitemaps used in this experiment were taken from DMOZ web directory. The output ontology is in OWL/XML format as figure 4.14 and 4.15 show. The full version of the output ontology of auto part domain is available in Appendix A as a sample of the learned ontologies. At
the end of the taxonomic phase, recall shows the successfully retrieved concepts. That is, the system retrieved the relevant concepts and did not miss any concept. Precision shows the results of the expert judging on relevant concepts as shown on table 4.7. That is, the expert excluding non-relevant concepts. For example, in this architectural services domain, the system determines that “Portfolio” and “Service” as concepts. Later, at the enhancement phase, recall shows the average of all system proposed relations for each threshold starting from .3 - .7. That is, the system, on average, proposes number of relevant relationships. Precision here shows the average, based on expert judging, of the proposed ontology relationships for each threshold starting .3 - .7 as shown in table 4.8. That is, the expert, on average, excludes some of the non-relevant relationships. For example, in the architectural services domain, it matched “Architecture” and “Services” with a 0.8 semantic similarity. Thus, “Architecture” is will be added as a synonym to “Services”.

Table 4.7 shows the recall and precision with threshold (> .65). The precision was 90.21%, which reflects the ratio of the relevant extracted concepts. For each extracted concept, there may be some synonym concepts because the system grouped related terms to a given concept. These synonym concepts are mapped to <EquivalentClasses> classes. Lowering the threshold in the taxonomic phase increases the overhead as the system picks the highest similarity result regardless. On electronics domain learned ontology, the system drew 347+ relationships including <Declaration>, <ObjectPropertyAssertion>, <EquivalentClasses>, <ClassAssertion>, and <SubClassOf>.
Figure 4.14: Sample of output ontology at apparel domain
<Declaration>
    <Class IRI="#Intake_Systems"/>
</Declaration>

......

<SubClassOf>
    <Class IRI="#Radiators"/>
    <Class IRI="#cooling_systems"/>
</SubClassOf>

......

<ClassAssertion>
    <Class IRI="#Brakes"/>
    <NamedIndividual IRI="#Brake_Calipers"/>
</ClassAssertion>

......

<ObjectPropertyAssertion>
    <ObjectProperty IRI="#effect"/>
    <NamedIndividual IRI="#Automatic_Transmissions"/>
    <NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>

......

<TransitiveObjectProperty>
    <ObjectProperty IRI="#part_of"/>
</TransitiveObjectProperty>

......
</Ontology>

Figure 4.15: Sample of auto parts domain ontology result
The average of precision 78.87% at table 4.8 because the system proposed a number of false positive object properties when the threshold was set .3 and .4.

<table>
<thead>
<tr>
<th></th>
<th>Arch. Services</th>
<th>Apparel</th>
<th>Electronics</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>Precision</td>
<td>93.70%</td>
<td>100.00%</td>
<td>76.92%</td>
<td>90.21%</td>
</tr>
</tbody>
</table>

Table 4.7: Recall and Precision results of Taxonomic phase

In enrichment stage, best results seem to be reached when the threshold is >.5. There is no minimum threshold set for SeU or SiU because there is still a chance to edit that sitemap. However, higher ratios are better because then the HTML sitemap is well-defined and has high number of meaningful titles.

<table>
<thead>
<tr>
<th></th>
<th>Arch. Services</th>
<th>Apparel</th>
<th>Electronics</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>45.60%</td>
<td>73.30%</td>
<td>74.40%</td>
<td>64.43%</td>
</tr>
<tr>
<td>Precision</td>
<td>75.20%</td>
<td>90.40%</td>
<td>71.00%</td>
<td>78.87%</td>
</tr>
</tbody>
</table>

Table 4.8: Recall and precision results of Enhancement phase

The statistical results show a clear positive relationship between precision average at enhancement phase and threshold at enhancement phase, as shown in Figure 4.16. This means the more semantically similar the matching result, the more meaningful the extracted relationship would be.
After the enrichment stage, the ontology becomes richer, and thus, easier to be refined by an expert. In this stage, it is important to keep individual concepts in mind. For example, in the architectural services domain, “Portfolio” class has “Residential_Architecture” as subclass. “Residential_Architecture” has “Traditional_Architecture” as an individual. When drawing a relation between “Green_Construction” that is an individual under the “Service” concept and “Traditional_Architecture”, the relation may be named as “Built_As”.

Some object properties are symmetric in their nature as is “Built_As”. For example, “Traditional_Architecture” “Built_As” “Green_Construction”, and “Green_Construction” “Built_As” “Traditional_Architecture”. The output ontology was tested on the Protégé 4 OWL API editor. Also, the Protégé FaCT++ built-in reasoned has been used in order to test the classifications of the ontology and no inconsistencies were noted as figures 4.17 and 4.18 show before and after using the reasoner.
Figure 4.17: Ontology before using Fact++ reasoned.

Figure 4.18: Ontology after using Fact++ reasoned
Figures 4.19 and 4.20 show a visualization of portions of the output ontologies from the architectural services and auto parts domains.

Figure 4.19: Visualization of ontology via OntoGraf from Protégé API for architectural services domain
Figure 4.20: Visualization of ontology via OntoGraf from Protégé API for auto parts domain.
4.4 Summary

The proposed approach has demonstrated that versions of HTML sitemaps can be useful when building ontologies. Semi-structured documents have lot of semantic information which can be processed. The diversity of tested domains shows that the approach is domain-independent. To address limitations of the proposed approach, it has been noted that the outputs are affected by the quality of HTML sitemaps. Finally, the approach can be improved to cover any hierarchical representation such as field classifications or domain thesauri in addition to HTML sitemaps.
CHAPTER 5

THREE DIMENSIONS ONTOLOGY MODIFICATION MATRIX

Indeed, the Semantic Web vision provides the current Web with the necessary infrastructure that allows the computer to process knowledge [36]. One of the cornerstones of this infrastructure is the Ontology [14]. Ontology is a formal way of sharing knowledge via a machine-readable format in order to facilitate the reuse of the knowledge among different parties [13]. For example, Web Ontology Language (OWL) defines the <owl:import> that allows the reuse of a desired content of another ontology. The evolution of the Semantic Web showed that there are a number of ontologies available. According to the Semantic Web search engine Swoogle\(^\text{15}\), there are more than 10,000 ontologies. Ontologies promote the goal of sharing common knowledge that advances the leverage of the Semantic Web vision. That is, ontology is expected to be modified in order to be accommodated with added, removed, or refined knowledge in order to fulfill new requirements. Otherwise, it would be outdated; resulting in poor Semantic Web application [12]. Also, when an ontology is about to be built or chosen, it is worthwhile to make sure that this ontology can be easily modified by other parties in order to suit new tasks. Given that, maintaining an ontology periodically will help in tracing the development of an ontology. Thus, in this part of the dissertation, a three

\(^{15}\)http://swoogle.umbc.edu/
dimensions ontology modification matrix approach is proposed as an attempt to relatively indicate the complexity of an ontology. That is, this approach can assist in estimating ontology modification efforts and tracking the ontology development process; yet, it is a simple approach. An empirical analysis has been performed to show the effectiveness of the proposed work. The proposed work concerned with identifying and quantitatively measuring the characteristics that reflect the knowledge expression elements rather than describing a given ontology as being functional or not. Altogether, this work is expected to allow an ontology to be reused in order to help make the vision of the Semantic Web more closer.

### 5.1 Background and Related Work

Verily, ontology allows the sharing of a structured understanding of knowledge among human or application agents, and enables the reuse of domain knowledge [15]. In fact, ontologies, in some task such ontology mapping and modularizing [64][56][55][65], are reused to support the notion of knowledge share and reuse.

In ontologies, there are basic fundamental elements that must be used in order to model an ontology. They are classes and relationships. Classes are used to represent concepts or terms in a domain. A class, within an ontology, shares some attributes with its subclasses. For example, class of “musical_instruments” would describe all musical instruments. A class describes a set of individuals that share the same characteristics. Classes are the most important elements when modeling knowledge that allow to specify
things [9], [10], [15]. Relationships allow describing concepts such as properties relationships that allow the describing of some internal characteristics about the concepts [15]. For example, origin, maker, weight, and dimensions are properties relationships for the “musical_instruments” concept [15]. Subclasses are special form of the relationships that define the hierarchy of a set of classes and allow for more control on a class. In ontologies, classes are organized in taxonomies. Subclasses are important because it shows inheritance relationships from the superclass. Subclasses allow more specific classes to be represented than the superclass. For example, the “musical_instruments” class would have guitar, violin, and drum as subclasses. Given that, addressing the complexity is a big job, and thus, this work is investigating these major foundation pillars as an attempt to relatively indicate the complexity of an ontology at the schema level.

A number of research papers have introduced some metrics as an attempted to maintain some aspects within an ontology such as provide some insights about ontology complexity, identify areas that might need investigation, support managing ontology development tasks, and quantify the quality by examining classes, relationships and instances. Given that, some metrics have been introduced based on the major foundation bases of an ontology, such as relationship richness, size of vocabulary, number of properties, and number of children, which will be discussed later in this section [32][33][66][67]. In this context, these research papers have been reviewed as a start for this work. The research paper [32] has proposed some metrics based on statistical aspects within ontologies. It allows for the automated measurement of ontologies and emphasize
on key characteristics of an ontology. They suggested schema metrics to address ontology modeling and its prospective for representing a knowledge, such as Relationship Richness (RR) metric that shows the variety of relationships in the ontology. Based on the RR metric, an ontology that has many properties relationships other than inheritance relationships is richer than taxonomy with only inheritance relationships. Below is the metric that has been proposed for measuring the (RR), where \( P \) is the number of properties relationships, and \( SC \) is the number of inheritance relationships.

\[
RR = \frac{|P|}{|SC| + |P|}
\]

Ontology with RR value that is close to zero, this would show that most of the relationships are inheritance relationships. Whereas, the ontology with an RR value that is close to one, would show that most of the relationships are other than inheritance relationships. Also, they proposed instance metrics to evaluate the placement of instance data and the effective usage of the ontology, such as average population metric that measures the number of instances compared to the number of classes. Ontology with low result may indicate that the instances are insufficient to represent all of the knowledge in the schema. Moreover, research [33] proposed analytics metrics with the aim of addressing their impact on the ontology quality process. These metrics consider some ontology characteristics such as size and structural characteristics. They claim that the
larger the metric result, the more effort that is required to understand and maintain the ontology. They proposed Size of Vocabulary (SOV) metric, as shown below, is an attempt to quantify how large an ontology is.

\[ SOV = |Nn| + |Pn| \]

Where (Nn) represents classes and individuals, and (Pn) represents user-defined properties. In this metric, the vocabulary includes named classes, properties relationships, and individuals. A high SOV value shows the large size of an ontology. Thus, it would need more time and effort in order to be built and maintained. In addition, they proposed Number of Children (NOC). It is a metric that takes into account the direct subclasses relationships from a given class. It computes only the direct inheritance. A high NOC value shows high reuse from a given class. Also, a high NOC value indicates that if a change occurs to a given class, its subclasses may require further effort and analysis in order to maintain changes needed for them. Also, research [66] aimed to analyze some feature such as quality and completeness of constructed ontology via some metrics, such as the number of properties and the average properties per class. The Number of Properties (NoP) metric counts the total number of all properties relationships within an ontology. The authors claim that this metric can show how completely the terms within an ontology are defined. More properties to classes show that it has been fully described. The average properties per class metric gives an indication of how well classes within the ontology have been defined. The research paper [67] introduced quantitative measures.
that estimate the complexity of ontology and its proliferation. These metrics analyze quantity, such as the number of concepts metric, which implies that the more concepts that are introduced by ontology, the more entity knowledge the ontology conveys; and the average path length per concept metric that shows the average modeling distance between specific concept and the most general concept in the ontology. Also, the same research paper proposed average relationships per concept metric, which shows the average connectivity degree of a concept. They claim that complexity of ontology can be tremendously increased based on the values of the proposed metrics.

Furthermore, research [12] mentioned that fast changing environments lead to outdated ontology and that indicates an outdated application of the Semantic Web. Later, when the author defines four main phases of ontology creating and specially on Prune phase, he mentioned on the balance between the completeness and the incompleteness of the domain model that “It is widely held belief that targeting completeness for the domain model appears to be practically inmanageable”. Large ontologies such as Gene Ontology\textsuperscript{16} and SNOMED CT\textsuperscript{17} require considerable effort in managing them. They are enormous ontologies that it would be very helpful for many applications to reuse part of them instead of the entire ontologies [68]. However, it is worth mentioning that some ontologies, by their nature, need to be complex; and the proposed work does not deal

\textsuperscript{16} http://geneontology.org/

\textsuperscript{17} http://www.ihtsdo.org/snomed-ct
with these aspects. Additionally, research paper [36] mentioned that Semantic Web field itself, is still under development and ontologies are seldom reflecting an application-independent model or community-agreed conceptualization. Also, it claims that sharing and reusing of ontologies increase the quality of the applications that are applying these two principles. That is, these applications become interoperable because they are provided with machine-processable and commonly agreed-upon understanding about a domain of interest.

From the above, these research papers have attempted to measure and understand about a given ontology. In other words, these metrics have provided helpful demonstration and disclosed some facts, such as what it means to have a high number of subclasses. Moreover, there is some emphasis on the amount of classes, subclasses relationships, and properties relationships within an ontology. Given that, investigating these characteristics allows better understanding about a given ontology and its internal characteristics. Ontology, that presents high amount of these elements would require more time to be understood or modified in order to be reused and shared. Different from all the above research, the three dimensions ontology modification matrix is proposed to measure the effort of modifying an ontology. In this research, the work on these metrics is extended. The proposed work is employing classes, subclasses relationships, and properties relationships. It shows a quantitative perspective that considers these characteristics and presents three dimensions ontology modification matrix, which facilitates understanding about an ontology.
5.2 Three Dimensions Ontology Modification Matrix

The literature shows that classes, subclasses relationships, and properties relationships have been considered as an attempt to study and understand about the effort involved when an ontology is modified. However, they independently consider these elements. Furthermore, one metric is not enough to identify the cost of modifying an ontology. That is, these three elements are considered, which relatively influence the complexity of an ontology as schema. To our interest, the proposed work is trying to estimate the effort of modifying a given ontology through these elements that can influence the complexity of an ontology. Such an approach would take into consideration classes, subclasses relationships, and properties relationships as figure 5.1 shows.

![Three dimensions ontology modification matrix](image)

Figure 5.1: Three dimensions ontology modification matrix

This work is taking into consideration and employing these characteristics in order to scale the effort of modifying a given ontology. From the literatures, research [66] has proposed Number of Properties (NoP) metric that counts the total number of properties relationships. It gives an indication of how completely the concepts within an ontology
are defined. Properties relationships allow describing some internal characteristics about the classes [15]. However, in order to encompass other basic elements an ontology, the amount of classes and subclasses relationships need to included. Total number of concepts metric [67] shows how many concepts or classes have been placed within an ontology. A greater number of classes introduced within an ontology, the more entity knowledge the ontology conveys. In practice, any subclass or classes to be added to an ontology, usually has more characteristics, has some restrictions characteristics, or holds different relationships than a superclass [15]. Also, Number of Children (NOC) metric [33] showed a way to compute the direct inheritance. Number of subclasses has been used to show the amount of the inheritances that one ontology has. High number of subclasses relationship shows the effort that may be needed when a given class is modified in order to inspect associated subclasses or superclass. In the figure 5.1 above, (C) stands for the number of classes metric, (SC) stands for number of subclasses metric, and (P) stands for the number of properties metric. This is denoted as (C,SC,P) point. The far result from (0,0,0) point is higher in cost to modify a given ontology that has a higher level of complexity. And vise versa, the closer it is, the less cost in modification. In figure 5.2 below, (d) shows the distance between the given ontology result and the origin coordinate.
The proposed work employs (C,SC,P), which are direct correspondence to uncloud, the effort associated with modifying an ontology as schema. For example, let us consider an ontology from vehicles domain as figure 5.3 shown. This ontology could get expanded by more classes and relationships. In figure 5.3 (vehicles ontology version 1.0), there are 9 classes and 5 subclasses relationships. As the ontology gets evolved, as shown in figure 5.5 (vehicles ontology version 2.0) and figure 5.4 (vehicles ontology version 3.0), more classes, subclasses relationships, and properties relationships from the same ontology domain (vehicles domain) inserted.

Figure 5.2: \((d)\) distance

Figure 5.3: Simple ontology from vehicles domain ontology version 1.0
Figure 5.5: Ontology from vehicles domain with little more classes and subclasses relationships (ontology version 2.0)

Figure 5.4: Ontology from vehicles domain with more classes and subclasses relationships and properties relationships (ontology version 3.0)
In figure 5.5, number of classes, subclasses relationships, and properties relationships have increased to be 14, 13 and 2, respectively. In figure 5.4, number of classes, subclasses relationships, and properties relationships have been magnified to be +130 classes, +150 subclasses relationships and +7 properties relationships. Notice that all of the three versions are from the same domain (vehicle domain). In figure 5.4, more classes are added, such as “aircraft” and “armored_truck” … etc and properties relationships have been added, such as “gear_type” and “origin_at” … etc. Figure 5.6 shows the distance from origin coordinate when plugging the number of classes, subclasses relationships and properties relationships from vehicle domain ontologies (V 2.0 and V 3.0) into the proposed three dimensions ontology modification matrix. It illustrates the trend of the evolution, which helps in tracking the development of an ontology.

Figure 5.6: Evolution trend
Also, in case a modification is needed, for instance, in the circled class as in figure 5.7, there are more efforts associated with it. Thus, considering these factors together is a way to disclose the modification cost about an ontology.

Figure 5.7: Efforts associated with modifying given class

In calculating the distance \((d)\) between ontologies v 2.0 to origin coordinates, the values are plugged into the distance formula \((d)\):
The values of V 2.0 are (14,13,2) and \( d_1 \) is the distance to origin coordinates as formula below shows:

\[
d_1 = \sqrt{(0 - 14)^2 + (0 - 13)^2 + (0 - 2)^2}
\]

\[
d_1 = \sqrt{(-14)^2 + (-13)^2 + (-2)^2}
\]

\[
d_1 = \sqrt{196 + 169 + 4}
\]

\[
d_1 = \sqrt{369}
\]

\[
d_1 = 19.209
\]

In calculating the distance between ontologies v 3.0 to origin coordinates, the values are plugged into the distance formula \( d \). The values of V 3.0 are (132,151,8) and \( d_2 \) is the distance to origin coordinates as formula below shows:

\[
d_2 = \sqrt{(0 - 132)^2 + (0 - 151)^2 + (0 - 8)^2}
\]

\[
d_2 = \sqrt{(-132)^2 + (-151)^2 + (-8)^2}
\]

\[
d_2 = \sqrt{17424 + 22801 + 64}
\]

\[
d_2 = \sqrt{40289}
\]
The two results $d_1$ and $d_2$ illustrate the cost of modification and level of complexity when employing these three factors. That is, the evolution trend that relatively indicates the complexity of an ontology can be understood. Therefore, using the proposed three dimensions ontology modification matrix, the effort associated with an ontology can be observed based on the distance from the origin coordinate.

5.2.1 Three Dimensions Ontology Modification Matrix Empirical Study

Based on the three dimensions ontology modification matrix, it can be estimated whether a given ontology is costly to modify or not. There are many cases where the proposed work can play a significant role when a decision is about to be made regarding an ontology. To clarify some practical cases, let us consider some real-world ontologies. The proposed work has been tested on Beta Cell Genomics Ontology (BCGO)\textsuperscript{18}, Plant Ontology (PO)\textsuperscript{19}, Symptom Ontology (SO)\textsuperscript{20}, Emotions Ontology (EO)\textsuperscript{21}, and Human

\begin{align*}
  d_2 &= 200.721
\end{align*}

\begin{flushright}
\textsuperscript{18} http://www.obofoundry.org/cgi-bin/detail.cgi?id=BCGO  \\
\textsuperscript{19} http://www.obofoundry.org/cgi-bin/detail.cgi?id=plant_ontology  \\
\textsuperscript{20} http://www.obofoundry.org/cgi-bin/detail.cgi?id=gemina_symptom  \\
\textsuperscript{21} http://www.obofoundry.org/cgi-bin/detail.cgi?id=MFOEM
\end{flushright}
Disease Ontology (HDO). These ontologies are large and require a great effort to be modified. The results are on table 5.1 below.

<table>
<thead>
<tr>
<th>Ontology name</th>
<th>No. of C.</th>
<th>No. of P.</th>
<th>No. of SC.</th>
<th>Dis. to origin coordinate (d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCGO</td>
<td>1882</td>
<td>117</td>
<td>3644</td>
<td>4102.96</td>
</tr>
<tr>
<td>PO</td>
<td>1691</td>
<td>10</td>
<td>2687</td>
<td>3174.83</td>
</tr>
<tr>
<td>SO</td>
<td>936</td>
<td>1</td>
<td>840</td>
<td>1257.65</td>
</tr>
<tr>
<td>EO</td>
<td>906</td>
<td>90</td>
<td>1114</td>
<td>1438.72</td>
</tr>
<tr>
<td>HDO</td>
<td>8946</td>
<td>15</td>
<td>6918</td>
<td>11308.84</td>
</tr>
</tbody>
</table>

Table 5.1: Results of testing on real world ontologies

Results from table 5.1 on these real world ontologies shows that using the three dimensions ontology modification matrix can assist in providing general understanding about an ontology. That is, using the proposed work and calculating the \((d)\) result; the proposed work can estimate ontology modification effort and complexity by looking at how far a given ontology is from the origin coordinates. This can be significantly beneficial when a decision is about to be made concerning an ontology.

It is worthwhile to mention that ontology modularization is a task that has some advantages such as ontology reuse and exploitation of the Semantic Web [64]. If an

\[22\] http://www.obofoundry.org/cgi-bin/detail.cgi?id=disease_ontology
ontology ends up having a high cost for the modification, it is advisable to be considered for partitioning. As defined by [69], ontology partitioning is a process of splitting the ontology into certain modules such that each module is considered as an ontology and grouping all modules is semantically equivalent to the original ontology. In order to get an ontology modularized, a web-based Manchester OWL Module Extractor\textsuperscript{23} has been used that can assist in partitioning one ontology based on signatures and options that a user may choose. Studying modularizing techniques of an ontology is outside the area of this thesis.

In the next case, let us say that there is an interest in getting ontology about “Places”. Moreover, there is an ontology called Semantic Web Technology Evaluation Ontology (SWETO)\textsuperscript{24} that is large scale Semantic Web ontology. It covers some topics including person, places, academic department, event, organization, etc. Now, SWETO ontology offers the modeling that is needed about “Places”. Here is the summary of SWETO ontology before and after testing as presented in table 5.2.

<table>
<thead>
<tr>
<th></th>
<th>No. of C.</th>
<th>No. of P.</th>
<th>No. of SC.</th>
<th>Dist. to origin coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before</td>
<td>114</td>
<td>69</td>
<td>111</td>
<td>173.43</td>
</tr>
<tr>
<td>After</td>
<td>8</td>
<td>14</td>
<td>7</td>
<td>17.75</td>
</tr>
</tbody>
</table>

Table 5.2: SWETO ontology case before and after

\textsuperscript{23} http://mowl-power.cs.man.ac.uk:8080/modularity/

\textsuperscript{24} http://lsdis.cs.uga.edu/proj/semdis/testbed/
Before applying the proposed work, it is a little vague how detailed this ontology is, what these numbers mean, and their consequences. In this case, before reusing “Places” concept and its associated relationships, the SWETO is tested on the three dimensions ontology modification matrix. After testing it, the proposed work indicated that it is far from the (0,0,0) point. Therefore, after considering the proposed work, the expected associated effort can be observed when modifying SWETO ontology. Thus, the “Places” module can be chosen and extracted alone. After extracting “Places” and testing it again using the three dimensions ontology modification matrix, it became closer to (0,0,0) point with a difference of -155.68 (subtract newer version from older version), which shows how much less detailed the “Places” concept and its associated relationships are compared to the older version.

Another case of practical uses of the proposed work is when the owner of the ontology would like to scale his ontology in order to make sure that the ontology is easy to be understood and updated. In this context, a test has been made on the developed ontology from “AutoPart” domain. The summary of the ontology is presented in table 5.3.

<table>
<thead>
<tr>
<th>No. of C.</th>
<th>No. of P.</th>
<th>No. of SC.</th>
<th>Dis. to origin coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>5</td>
<td>2</td>
<td>13.15</td>
</tr>
</tbody>
</table>

Table 5.3: Summary of “AutoPart” domain ontology
This shows that the ontology is close to (0,0,0) point. That is, using the proposed work; the ontology owner can keep track of the ontology development process. Also, this ontology should be frequently tested on the three dimensions ontology modification matrix after any updates in order to ensure that it remains close to (0,0,0) point.

Furthermore, as this work is showing competitive results, the proposed work is compared with SOV metric proposed by [33]. SOV is a metric that takes into consideration the number of classes, properties relationships, and individuals as formula below shows:

\[ SOV = |Nn| + |Pn| \]

The authors claim a High SOV value shows a large size of an ontology. Thus, it would need more time and effort to be built and maintained. In comparing the proposed work with SOV metric, let us consider a university ontology that has two versions v 1.0 and v 2.0 as figures 5.8 and 5.9 below shown.
The version 1.0 of university ontology has 13 classes, 12 subclasses relationships, and 3 properties relationships. On v 2.0, more subclasses relationships have been added to show that the grad student is subclasses of part time faculty, full time faculty, system staff, and administrative staff as figure 5.9 below shown.

Figure 5.8: University ontology versions 1.0
The version 2.0 has 13 classes, 16 subclasses relationships, and 3 properties relationships. Table 5.4 shows the results of the comparison. In regards to the evolution between the versions, it can be observed that the proposed work has interpreted and noted the modification occurring in the ontology. The updated version of the ontology ends up having more subclasses relationships, which show more inheritance relationships. That is, the proposed work showed how these added subclasses relationships can influence (d) result.

Figure 5.9: University ontology versions 2.0

133
Table 5.4: Summary of the compression

<table>
<thead>
<tr>
<th>Ontology version</th>
<th>No. of C.</th>
<th>No. of P.</th>
<th>No. of SC.</th>
<th>Dis. to origin coordinate ($d$)</th>
<th>SOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>V 1.0</td>
<td>13</td>
<td>3</td>
<td>12</td>
<td>17.23</td>
<td>16</td>
</tr>
<tr>
<td>V 2.0</td>
<td>13</td>
<td>3</td>
<td>16</td>
<td>20.22</td>
<td>16</td>
</tr>
</tbody>
</table>

5.3 Advantages of Three Dimensions Ontology Modification Matrix

This work provides a fast way to decide about an ontology. Also, it allows and provides better understanding of the overall ontology. Moreover, the three dimensional ontology modification matrix provides some insights about the internal structure of a given ontology that may lead to modulization. This proposed work measures how much a given ontology can cost to be modified based on the how far it is from the (0,0,0) point. Moreover, the proposed work can evaluate ontologies in order to differentiate between the numbers of ontologies to choose from. In addition, the proposed work is helpful in tracking and evaluating ontologies in order to control the development process. Finally, the three dimensions ontology modification matrix approach is employing (C,SC,P) elements to understand the effort of modifying given ontology at schema level as an attempt relatively indicate the complexity of a given ontology. Yet, it is simple that allows the domain expert or the end-user (with no experience in ontology engineering) to be able to better understand and make a decision about an ontology.
5.4 Summary

In this chapter, and based on identifying the major pillars in building an ontology, a three dimensional ontology modification matrix is proposed. It shows how these pillars can assist in disclosing the effort of modifying an ontology. The proposed approach measures the \((d)\) distance for a given ontology from origin point, and based on that, it can assist in number of application, such as noting ontology modification cost and tracking ontologies development processes. In this context, the future work of the proposed would be investigating how these foundation pillars are interwoven and measuring and analyzing the variety of the relationships. In addition to this chapter, number of research papers that propose some metrics in regard to understand about these elements have been surveyed. Also, an empirical study on the proposed work in a number of cases that include real world ontologies has been made. Also, a comparison of the proposed work and the result of SOV metric [33] has been made. The empirical study shows the effectiveness and usefulness of the proposed work. In short, the proposed work provides a better understanding of an ontology that allows revealing the effort of the modification as an attempt to relatively indicate the complexity of an ontology. As a matter of fact, ontology will change. Thus, maintaining an ontology periodically will ensure its permanence.
CHAPTER 6

CONCLUSION AND FUTURE WORK

In this research, a novel methodology for learning ontologies from HTML sitemaps has introduced. This work serves in filling the gap between web generations and one step in web generation linkage. Semantic Web is an extension of current Web generation [4]. This work, through easy steps, can group multiple HTML sitemaps to reveal populated domain ontology. First, it learns the taxonomic relationship, and then enhance it by enriching and refining. The output ontology is considered as a populated ontology (produce instances, assertions) that reduces the issue semantics ambiguity. That is because the class meaning can be known by both class name and class instances. The proposed work is a semi-automated approach in ontology development that takes the advantages of manual and automatic approaches and limits the disadvantages of each above mentioned methods when used in isolation.

Also, on this research, the three dimensional ontology modification matrix has introduced. Ontologies promote the goal of sharing common knowledge that advances the leverage of the Semantic Web vision. Hence, ontology is expected to be modified in order to be accommodated with added, removed, or refined knowledge in order to fulfill new requirements. Thus, it is very helpful to scale the ontology by easy-to-use methods for better understanding about an ontology. In other words, this study attempts to relatively indicate the complexity of an ontology. Such an approach can assist in estimating ontology modification effort and tracking ontology development process and its
consequences. The proposed work considers distance function to measure how far a
given ontology from the origin coordinate. The further the result is from (0,0,0), the
higher in cost it is to modify a given ontology, which indicates a higher level of
complexity. And vise versa, the closer it is, the less cost in modification and complexity.

Semantic Web (Web 3.0) is considered to be the infrastructure of the future
generations Web 4.0 and Web 5.0 as figure 6.1 below. Future Web generations are
expected to heavily depend on structured documents. Although as of yet there is no clear
technology for the Web 4.0, but it is widely believed that Web 4.0 will be a “Symbiotic
Web” or “WebOS” that is Read-Write-Execution-Concurrency Web [70]. As Web 2.0
enhanced the functionality of Web 1.0, Web 4.0 would enhance the functionality of Web
3.0. The main reason is that Web 4.0 would more effectively process structured
documents. Web 4.0 would merge with Semantic Web technologies to produce agent-
based applications that provide products that work as operating systems to facilitate the
interaction between humans and computers. It would introduce intelligent applications
based on Semantic Web technologies. Web 3.0 is expected to provide the infrastructure
to develop Web 4.0. It is important to understand that Web evolution is based on needs.
That is, Web 2.0 complemented Web 1.0 by adding more functionality and Web 3.0
solves major issues in Web 2.0. Thus, Web 4.0 should enhance the user experience of
Web 3.0.
Research in [70] mentioned that the Web is moving to artificial intelligence. Web 4.0 is considered as WebOS, where smart agents do smart actions over the web. These agents would be able to make a decision based on what a user wants, based on current conditions, along with other information to create an ideal respond. Having considered rich ontologies and knowledge sharing in the Web 4.0, it is also reasonable to look at automated and on-the-fly reasoning. This requires good quality ontologies in order to get smart and correct decisions. Web 5.0 is expected to take into account the feeling of the user. It is guided by technologies that already exist to measure feelings and their effects. Research [71] described Web 5.0 as “Symbionet Web”. They mentioned that the Web is "emotionally" neutral and does not count on a user’s feelings. It could consider structured data along with structured documents. Web 5.0 may incorporate with date fusion and multi-agent systems [72][73] fields theories to create a better experience. Also, the Web 5.0 may take advantage of data fusion algorithms and applications to merge with pervious web generations. The data fusion field has already proven successful in a number of domains including discovery science and business intelligence. Web 5.0 could be about “Fusion Web” where machines and people will process data in forms they can deal with, interact with, and make decisions with. Web 5.0 could be a Read-Write-Execute-Concurrent-Fusion web. Web 5.0 or “Fusion Web” may join “Web For All” to support people with special needs.
Figure 6.1: Web generations

---

25 Part of this chapter is accepted on The 2015 International Conference on Internet Computing and Big Data.
As for future work, the proposed ontology learning methodology is considered to be improved to allow learning ontologies from field classifications or domain thesauri, in addition to HTML sitemaps. Moreover, a work is expected to be done in order to make it language-independent and starting with Arabic language. Also, future work includes investigating more about how the classes, subclasses relationships, and properties relationships are interwoven; and on measuring and analyzing the variety of the relationships.
APPENDIX A

Output of learned ontology from auto parts domain.

<?xml version="1.0"?>
<!DOCTYPE Ontology [
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#” >
<!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#” >
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#” >
]
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base="http://www.semanticweb.org/aalgosai/ontologies/2013/10/untitled-ontology-2"
ontologyIRI="http://www.semanticweb.org/aalgosai/ontologies/2013/10/untitled-ontology-2">
<Declaration>
<Class IRI="#Air_Filters_or_Intake_Systems"/>
</Declaration>
<Declaration>
<Class IRI="#Brakes"/>
</Declaration>
<Declaration>
<Class IRI="#Electrical"/>
</Declaration>
<Declaration>
<Class IRI="#Exhaust_system"/>
</Declaration>
<Declaration>
<Class IRI="#Fuel_Systems"/>
</Declaration>
<Declaration>
<Class IRI="#Radiator"/>
</Declaration>
<Declaration>
<Class IRI="#Thermostats"/>
</Declaration>
<Declaration>
<Class IRI="#Transmissions_or_Shifters"/>
</Declaration>
</Ontology>
<Class IRI="#cooling_systems"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#attached_with"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#effect"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#part_of"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#related_to"/>
</Declaration>
<EquivalentClasses>
  <Class IRI="#Cooling_Systems"/>
  <ObjectSomeValuesFrom>
    <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
    <Class IRI="#cooling_systems"/>
  </ObjectSomeValuesFrom>
</EquivalentClasses>
<EquivalentClasses>
  <Class IRI="#Exhaust_system"/>
  <ObjectSomeValuesFrom>
    <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
    <Class IRI="#Exhaust_Systems"/>
  </ObjectSomeValuesFrom>
</EquivalentClasses>
<EquivalentClasses>
  <Class IRI="#Ignition_Systems"/>
  <ObjectSomeValuesFrom>
    <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
    <Class IRI="#Cooling_Systems"/>
  </ObjectSomeValuesFrom>
</EquivalentClasses>
/SubClassOf>
  <Class IRI="#Radiator"/>
  <Class IRI="#cooling_systems"/>
</SubClassOf>
/SubClassOf>
  <Class IRI="#Thermostats"/>
  <Class IRI="#cooling_systems"/>
</SubClassOf>
<ClassAssertion>
  <Class IRI="#Air_Filters_or_Intake_Systems"/>
  <NamedIndividual IRI="#Air_Fitting"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Air_Filters_or_Intake_Systems"/>
  <NamedIndividual IRI="#Air_Flap_Actuator"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Air_Filters_or_Intake_Systems"/>
  <NamedIndividual IRI="#Air_Intake_Duct"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Air_Filters_or_Intake_Systems"/>
  <NamedIndividual IRI="#Air_Intake_Snorkel"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Air_Filters_or_Intake_Systems"/>
</classAssertion>
</ClassAssertion>
</ClassAssertion>
</ClassAssertion>
</ClassAssertion>
</ClassAssertion>
</classAssertion>
</ClassAssertion>
</ClassAssertion>
</ClassAssertion>
<NamedIndividual IRI="#Air_Velocity_Stack"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Transmissions_or_Shifters"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Brakes"/>
  <NamedIndividual IRI="#Brake_Calipers"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Brakes"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Brakes"/>
  <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Air_Filters_or_Intake_Systems"/>
  <NamedIndividual IRI="#Breather_Filter"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Air_Filters_or_Intake_Systems"/>
  <NamedIndividual IRI="#Breather_Hose"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Exhaust_system"/>
  <NamedIndividual IRI="#CATALYTIC_CONVERTERS"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Fuel_Systems"/>
  <NamedIndividual IRI="#Carburetors"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Electrical"/>
  <NamedIndividual IRI="#Charging_System"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#cooling_systems"/>
  <NamedIndividual IRI="#Coolant_Reservoirs"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Transmissions_or_Shifters"/>
  <NamedIndividual IRI="#Cruise_Control"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Air_Filters_or_Intake_Systems"/>
  <NamedIndividual IRI="#Deceleration_Pump"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Fuel_Systems"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Transmissions_or_Shifters"/>
  <NamedIndividual IRI="#Differential"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Transmissions_or_Shifters"/>
  <NamedIndividual IRI="#Drive_Shaft"/>
<ClassAssertion>
  <Class IRI="#Exhaust_system"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Electrical"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Exhaust_system"/>
  <NamedIndividual IRI="#Exhaust_Brakes"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Exhaust_system"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Exhaust_system"/>
  <NamedIndividual IRI="#Exhaust_Systems"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Exhaust_system"/>
  <NamedIndividual IRI="#FUEL_FILTERS"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Exhaust_system"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Exhaust_system"/>
  <NamedIndividual IRI="#FUEL_TANKS"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Air_Filters_or_Intake_Systems"/>
  <NamedIndividual IRI="#Flame_Trap"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Transmissions_or_Shifters"/>
  <NamedIndividual IRI="#Fluids_or_Chemicals"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Fuel_Systems"/>
  <NamedIndividual IRI="#Fuel_Pumps"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Fuel_Systems"/>
  <NamedIndividual IRI="#Fuel_Savers"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Electrical"/>
  <NamedIndividual IRI="#Generators"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#cooling_systems"/>
  <NamedIndividual IRI="#HOSES"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="#Air_Filters_or_Intake_Systems"/>
  <NamedIndividual IRI="#Intake_Cover_Cap"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Electrical"/>
  <NamedIndividual IRI="#Lights,_Flasher_Units,_Fuses"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Transmissions_or_Shifters"/>
  <NamedIndividual IRI="#Manual_Transmissions"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Exhaust_system"/>
  <NamedIndividual IRI="#Mufflers"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#cooling_systems"/>
  <NamedIndividual IRI="#OIL_COOLERS"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Radiator"/>
  <NamedIndividual IRI="#Radiator_Fan_Blades"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Radiator"/>
  <NamedIndividual IRI="#Radiator_Fan_Clutch"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Radiator"/>
  <NamedIndividual IRI="#Radiator_Hardware"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Radiator"/>
  <NamedIndividual IRI="#Radiator_Hoses"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Radiator"/>
  <NamedIndividual IRI="#Radiator_Mounts"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Radiator"/>
  <NamedIndividual IRI="#Radiator_Overflow_Hoses"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Radiator"/>
  <NamedIndividual IRI="#Radiator_Sensors"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Radiator"/>
  <NamedIndividual IRI="#Radiators_Caps"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#cooling_systems"/>
  <NamedIndividual IRI="#Temperature_Sensors"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Transmissions_or_Shifters"/>
  <NamedIndividual IRI="#Transfer_Case_Parts"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Transmissions_or_Shifters"/>
  <NamedIndividual IRI="#Transmission_Gaskets"/>
</ClassAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#attached_with"/>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#HOSES"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#Carburetors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#Temperature_Sensors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#CATALYTIC_CONVERTERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#Flame_Trap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#Mufflers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#Exhaust_Pipes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#EXHAUST_HEADERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#FUEL_FILTERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#Exhaust_Systems"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#FUEL_INJECTORS"/>
</ObjectPropertyAssertion>
<NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#part_of"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
  <NamedIndividual IRI="#Fuel_Savers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#part_of"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
  <NamedIndividual IRI="#OIL_COOLERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#part_of"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
  <NamedIndividual IRI="#Air_Intake_Snorkel"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
  <NamedIndividual IRI="#Air_Fitting"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
  <NamedIndividual IRI="#Air_Velocity_Stack"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
  <NamedIndividual IRI="#Fuel_Pumps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
  <NamedIndividual IRI="#Deceleration_Pump"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
  <NamedIndividual IRI="#Breather_Hose"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
  <NamedIndividual IRI="#Intake_Cover_Cap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
  <NamedIndividual IRI="#Breather_Filter"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
<NamedIndividual IRI="#Air_Flap_Actuator"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Brake_Calipers"/>
<NamedIndividual IRI="#Flame_Trap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Brake_Calipers"/>
<NamedIndividual IRI="#Fuel_Pumps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Brake_Calipers"/>
<NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Brake_Calipers"/>
<NamedIndividual IRI="#Fuel_Savers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Brake_Calipers"/>
<NamedIndividual IRI="#Differential"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Brake_Calipers"/>
<NamedIndividual IRI="#Automatic_Transmissions"/> 
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Brake_Calipers"/>
<NamedIndividual IRI="#Carburetors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Brake_Calipers"/>
<NamedIndividual IRI="#FUEL_INJECTORS"/> 
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Brake_Calipers"/>
<NamedIndividual IRI="#FUEL_TANKS"/> 
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#part_of"/>
<NamedIndividual IRI="#Brake_Calipers"/>
<NamedIndividual IRI="#OIL_COOLERS"/> 
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#part_of"/>
<NamedIndividual IRI="#Brake_Calipers"/>
<NamedIndividual IRI="#Lights,_Flasher_Units,_Fuses"/> 
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Brake_Calipers"/>
<NamedIndividual IRI="#Cruise_Control"/> 
</ObjectPropertyAssertion>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#Fuel_Pumps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#FUEL_TANKS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#Differential"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#Carburetors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#Flame_Traps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#Fuel_Savers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#part_of"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#OIL_COOLERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#part_of"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#HOSES"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#part_of"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#Lights_Flasher_Units_Fuses"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Brake_Drums_or_Rotors"/>
  <NamedIndividual IRI="#Cruise_Control"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
  <NamedIndividual IRI="#Carburetors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
  <NamedIndividual IRI="#Flame_Trap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
  <NamedIndividual IRI="#Differential"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
  <NamedIndividual IRI="#Fuel_Savers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
  <NamedIndividual IRI="#FUEL_TANKS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
  <NamedIndividual IRI="#Fuel_Pumps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
  <NamedIndividual IRI="#Fuel_INJECTORS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
  <NamedIndividual IRI="#Deceleration_Pump"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#part_of"/> 
  <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
  <NamedIndividual IRI="#OIL_COOLERS"/>
</ObjectPropertyAssertion>
<NamedIndividual IRI="#CATALYTIC_CONVERTERS"/>
<NamedIndividual IRI="#Breather_Filter"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Carburetors"/>
<NamedIndividual IRI="#Fuel_Pumps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fuel_TANKS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#part_of"/>
<NamedIndividual IRI="#Carburetors"/>
<NamedIndividual IRI="#Fuel_Savers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#part_of"/>
<NamedIndividual IRI="#Carburetors"/>
<NamedIndividual IRI="#Breather_Hose"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#part_of"/>
<NamedIndividual IRI="#Carburetors"/>
<NamedIndividual IRI="#Air_Velocity_Stack"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Air_Flap_Actuator"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Systems"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Carburetors"/>
<NamedIndividual IRI="#Flame_Trap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Carburetors"/>
  <NamedIndividual IRI="#CATALYTIC_CONVERTERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Carburetors"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Carburetors"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Carburetors"/>
  <NamedIndividual IRI="#Air_Fitting"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Carburetors"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Carburetors"/>
  <NamedIndividual IRI="#Exhaust_Brakes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Carburetors"/>
  <NamedIndividual IRI="#Mufflers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Carburetors"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Carburetors"/>
  <NamedIndividual IRI="#Deceleration_Pump"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Carburetors"/>
  <NamedIndividual IRI="#Air_Intake_Snorkel"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Fuel_Pumps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Deceleration_Pump"/>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Temperature_Sensors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Radiator_Fan_Clutch"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Carburetors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Differential"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Radiator_Fan_Blades"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Fuel_Savers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Radiator_Sensors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Cruise_Control"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Charging_System"/>
  <NamedIndividual IRI="#Flame_Trap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Coolant_Reservoirs"/>
  <NamedIndividual IRI="#Coolant_Reservoirs"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Coolant_Reservoirs"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Coolant_Reservoirs"/>
  <NamedIndividual IRI="#OIL_COOLERS"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Air_Intake_Snorkel"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Air_Fitting"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#CATALYTIC_CONVERTERS"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Deceleration_Pump"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Fuel_Pumps"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Air_Velocity_Stack"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Intake_Cover_Cap"/>
</ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#Breather_Hose"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#EXHAUST_HEADERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#Flame_Traps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#FUEL_FILTERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Air_Intake_Duct"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#Breather_Filter"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#Air_Flap_Actuator"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#Exhaust_Systems"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#part_of"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#Fuel_Savers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#FUEL_INJECTORS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Cruise_Control"/>
<NamedIndividual IRI="#Carburetors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Air_Intake_Duct"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Mufflers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Exhaust_Systems"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Exhaust_Brakes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#CATALYTIC_CONVERTERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Air_Velocity_Stack"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Intake_Cover_Cap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Air_Fitting"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Air_Intake_Snorkel"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Breather_Hose"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>
<NamedIndividual IRI="#Breather_FIlter"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Air_Flap_Actuator"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#part_of"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Flame_Traps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Deceleration_Pump"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#FUEL_TANKS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#HOSES"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Drive_Shaft"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Drive_Shaft"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
  <NamedIndividual IRI="#Flame_Traps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#DriveShaft"/>
  <NamedIndividual IRI="#Carburetors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#DriveShaft"/>
  <NamedIndividual IRI="#DieselInjectors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
  <NamedIndividual IRI="#AirFlapActuator"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
  <NamedIndividual IRI="#FlameTrap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
  <NamedIndividual IRI="#DecelerationPump"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
  <NamedIndividual IRI="#AirIntakeSnorkel"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
  <NamedIndividual IRI="#BreatherFilter"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
  <NamedIndividual IRI="#AirFitting"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
  <NamedIndividual IRI="#BreatherHose"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
  <NamedIndividual IRI="#AirIntakeDuct"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
  <NamedIndividual IRI="#IntakeCoverCap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
  <NamedIndividual IRI="#AirVelocityStack"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#Differential"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#Fuel_Savers"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#OIL_COOLERS"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#Deceleration_Pump"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#Carburetors"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#Fuel_Pumps"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#Radiator_Fan_Blades"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#Temperature_Sensors"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#FUEL_TANKS"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
<NamedIndividual IRI="#Cruise_Control"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
<NamedIndividual IRI="#Flame_Traps"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
<NamedIndividual IRI="#Air_Intake_Duct"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
<NamedIndividual IRI="#Air_Fitting"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
<NamedIndividual IRI="#Air_Velocity_Stack"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
<NamedIndividual IRI="#Intake_Cover_Cap"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
<NamedIndividual IRI="#Breather_Hose"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
<NamedIndividual IRI="#Air_Intake_Snorkel"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Exhaust_Pipes"/>
<NamedIndividual IRI="#Flame_Traps"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
<NamedIndividual IRI="#Flame_Traps"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
<NamedIndividual IRI="#Flame_Traps"/>
</ObjectPropertyAssertion>
<NamedIndividual IRI="#Exhaust_Pipes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
  <NamedIndividual IRI="#Deceleration_Pump"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
  <NamedIndividual IRI="#Air_Intake_Snorkel"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
  <NamedIndividual IRI="#Breather_Filter"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
  <NamedIndividual IRI="#Air_Intake_Duct"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
  <NamedIndividual IRI="#Breather_Hose"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
  <NamedIndividual IRI="#Air_Velocity_Stack"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
  <NamedIndividual IRI="#Intake_Cover_Cap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
  <NamedIndividual IRI="#Air_Fitting"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Exhaust_Systems"/>
  <NamedIndividual IRI="#Flame_Trap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Exhaust_Systems"/>
  <NamedIndividual IRI="#Deceleration_Pump"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Exhaust_Systems"/>
  <NamedIndividual IRI="#Air_Flap_Actuator"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Exhaust_Systems"/>
<NamedIndividual IRI="#Air_Fitting"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Systems"/>
<NamedIndividual IRI="#Air_Intake_Snorkel"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Systems"/>
<NamedIndividual IRI="#Breather_Filter"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Systems"/>
<NamedIndividual IRI="#Air_Velocity_Stack"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Systems"/>
<NamedIndividual IRI="#Intake_Cover_Cap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Systems"/>
<NamedIndividual IRI="#Air_Intake_Duct"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Exhaust_Systems"/>
<NamedIndividual IRI="#Breather_Hose"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#FUEL_FILTERS"/>
<NamedIndividual IRI="#Deceleration_Pump"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#FUEL_FILTERS"/>
<NamedIndividual IRI="#CATALYTIC_CONVERTERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#FUEL_FILTERS"/>
<NamedIndividual IRI="#Exhaust_Pipes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#FUEL_FILTERS"/>
<NamedIndividual IRI="#EXHAUST_HEADERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#FUEL_FILTERS"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#FUEL_FILTERS"/>
<NamedIndividual IRI="#Mufflers"/>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#FUEL_FILTERS"/>
  <NamedIndividual IRI="#Exhaust_Systems"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#FUEL_FILTERS"/>
  <NamedIndividual IRI="#FlameTrap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
  <NamedIndividual IRI="#Air_Intake_Snorkel"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
  <NamedIndividual IRI="#Air_Fitting"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
  <NamedIndividual IRI="#Mufflers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
  <NamedIndividual IRI="#Air_Intake_Duct"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
  <NamedIndividual IRI="#CATALYTIC_CONVERTERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
  <NamedIndividual IRI="#Intake_Cover_Cap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
  <NamedIndividual IRI="#Air_Velocity_Stack"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
  <NamedIndividual IRI="#Breather_Hose"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
  <NamedIndividual IRI="#Air_Flap_Actuator"/>
</ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#FUEL_FILTERS"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#Radiator_Fan_Blades"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#Exhaust_Pipes"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#Exhaust_Systems"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#Fuel_Savers"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#EXHAUST_HEADERS"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#Exhaust_Brakes"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#Lights,_Flasher_Units,_Fuses"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#FUEL_TANKS"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#Radiator_Mounts"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#Radiators_Caps"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
</ObjectPropertyAssertion>
<NamedIndividual IRI="#Fluids_or_Chemicals"/>
<NamedIndividual IRI="#Radiator_Hardware"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fuel_Pumps"/>
<NamedIndividual IRI="#Mufflers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fuel_Pumps"/>
<NamedIndividual IRI="#Exhaust_Pipes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fuel_Pumps"/>
<NamedIndividual IRI="#CATALYTIC_CONVERTERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fuel_Pumps"/>
<NamedIndividual IRI="#EXHAUST_HEADERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#related_to"/>
<NamedIndividual IRI="#Fuel_Pumps"/>
<NamedIndividual IRI="#Flame_Traps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fuel_Savers"/>
<NamedIndividual IRI="#Exhaust_Pipes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fuel_Savers"/>
<NamedIndividual IRI="#Mufflers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fuel_Savers"/>
<NamedIndividual IRI="#Air_Flap_Actuator"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fuel_Savers"/>
<NamedIndividual IRI="#Air_Fitting"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Fuel_Savers"/>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#HOSES"/>
  <NamedIndividual IRI="#FUEL_INJECTORS"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#HOSES"/>
  <NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#part_of"/>
  <NamedIndividual IRI="#HOSES"/>
  <NamedIndividual IRI="#Breather_Hose"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#HOSES"/>
  <NamedIndividual IRI="#Breather_Filter"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#HOSES"/>
  <NamedIndividual IRI="#FUEL_FILTERS"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#attached_with"/>
  <NamedIndividual IRI="#Lights,_Flasher_Units,_Fuses"/>
  <NamedIndividual IRI="#Radiator_Sensors"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#attached_with"/>
  <NamedIndividual IRI="#Lights,_Flasher_Units,_Fuses"/>
  <NamedIndividual IRI="#OIL_COOLERS"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#attached_with"/>
  <NamedIndividual IRI="#Lights,_Flasher_Units,_Fuses"/>
  <NamedIndividual IRI="#Cruise_Control"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#attached_with"/>
  <NamedIndividual IRI="#Lights,_Flasher_Units,_Fuses"/>
  <NamedIndividual IRI="#Temperature_Sensors"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#attached_with"/>
  <NamedIndividual IRI="#Lights,_Flasher_Units,_Fuses"/>
  <NamedIndividual IRI="#Automatic_Transmissions"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#attached_with"/>
  <NamedIndividual IRI="#Lights,_Flasher_Units,_Fuses"/>
  <NamedIndividual IRI="#FUEL_TANKS"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#attached_with"/>
  <NamedIndividual IRI="#Lights,_Flasher_Units,_Fuses"/>
  <NamedIndividual IRI="#Coolant_Reservoirs"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights, Flasher Units, Fuses"/>
<NamedIndividual IRI="#Fuel_Pumps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights, Flasher Units, Fuses"/>
<NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights, Flasher Units, Fuses"/>
<NamedIndividual IRI="#Carburetors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights, Flasher Units, Fuses"/>
<NamedIndividual IRI="#Radiator_Fan_Blades"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights, Flasher Units, Fuses"/>
<NamedIndividual IRI="#Radiator_Overflow_Hoses"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights, Flasher Units, Fuses"/>
<NamedIndividual IRI="#Radiator_Fan_Clutch"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights, Flasher Units, Fuses"/>
<NamedIndividual IRI="#Deceleration_Pump"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights, Flasher Units, Fuses"/>
<NamedIndividual IRI="#Radiator_Mounts"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights, Flasher Units, Fuses"/>
<NamedIndividual IRI="#FUEL_INJECTORS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights, Flasher Units, Fuses"/>
<NamedIndividual IRI="#FUEL_FILTERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights, Flasher Units, Fuses"/>
<NamedIndividual IRI="#Flame_Trap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights, Flasher Units, Fuses"/>
<NamedIndividual IRI="#Fuel_Savers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<?xml version="1.0" encoding="UTF-8"?>
<NamedIndividual IRI="#Lights,_Flasher_Units,_Fuses"/>
<Nameworthy IRI="#Radiator_Hardware"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Lights,_Flasher_Units,_Fuses"/>
<NamedIndividual IRI="#Radiators_Caps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Manual_Transmissions"/>
<NamedIndividual IRI="#EXHAUST_HEADERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Manual_Transmissions"/>
<NamedIndividual IRI="#Intake_Cover_Cap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Manual_Transmissions"/>
<NamedIndividual IRI="#Breather_Fuel"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Manual_Transmissions"/>
<NamedIndividual IRI="#Diesel_Injectors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Manual_Transmissions"/>
<NamedIndividual IRI="#Air_Velocity_Stack"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Manual_Transmissions"/>
<NamedIndividual IRI="#Flame_Trap"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Manual_Transmissions"/>
<NamedIndividual IRI="#FUEL_INJECTORS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Manual_Transmissions"/>
<NamedIndividual IRI="#Air_Intake_Snorkel"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Manual_Transmissions"/>
<NamedIndividual IRI="#Air_Fitting"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Manual_Transmissions"/>
<NamedIndividual IRI="#Carburetors"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Manual_Transmissions"/>
<NamedIndividual IRI="#Air_Flap_Actuator"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
 <ObjectProperty IRI="#effect"/>
 <NamedIndividual IRI="#Manual_Transmissions"/>
 <NamedIndividual IRI="#Brake_Calipers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
 <ObjectProperty IRI="#effect"/>
 <NamedIndividual IRI="#Manual_Transmissions"/>
 <NamedIndividual IRI="#CATALYTIC_CONVERTERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
 <ObjectProperty IRI="#effect"/>
 <NamedIndividual IRI="#Manual_Transmissions"/>
 <NamedIndividual IRI="#Fuel_Savers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
 <ObjectProperty IRI="#effect"/>
 <NamedIndividual IRI="#Manual_Transmissions"/>
 <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
 <ObjectProperty IRI="#effect"/>
 <NamedIndividual IRI="#Manual_Transmissions"/>
 <NamedIndividual IRI="#Deceleration_Pump"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
 <ObjectProperty IRI="#effect"/>
 <NamedIndividual IRI="#Manual_Transmissions"/>
 <NamedIndividual IRI="#FUEL_FILTERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
 <ObjectProperty IRI="#effect"/>
 <NamedIndividual IRI="#Manual_Transmissions"/>
 <NamedIndividual IRI="#Brake_Pads_or_Shoes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
 <ObjectProperty IRI="#effect"/>
 <NamedIndividual IRI="#Manual_Transmissions"/>
 <NamedIndividual IRI="#Exhaust_Brakes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
 <ObjectProperty IRI="#effect"/>
 <NamedIndividual IRI="#Manual_Transmissions"/>
 <NamedIndividual IRI="#Exhaust_Pipes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Mufflers"/>
  <NamedIndividual IRI="#Breather_Filter"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Mufflers"/>
  <NamedIndividual IRI="#Air_Intake_Duct"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#related_to"/>
  <NamedIndividual IRI="#Mufflers"/>
  <NamedIndividual IRI="#Air_Velocity_Stack"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Temperature_Sensors"/>
  <NamedIndividual IRI="#Exhaust_Pipes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Temperature_Sensors"/>
  <NamedIndividual IRI="#CATALYTIC_CONVERTERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Temperature_Sensors"/>
  <NamedIndividual IRI="#EXHAUST_HEADERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Temperature_Sensors"/>
  <NamedIndividual IRI="#Mufflers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Temperature_Sensors"/>
  <NamedIndividual IRI="#Exhaust_Systems"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Temperature_Sensors"/>
  <NamedIndividual IRI="#Exhaust_Brakes"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Transfer_Case_Parts"/>
  <NamedIndividual IRI="#Flame_Traps"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Transfer_Case_Parts"/>
  <NamedIndividual IRI="#Electrical_Connectors_or_Sockets"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#effect"/>
  <NamedIndividual IRI="#Transfer_Case_Parts"/>
  <NamedIndividual IRI="#Lights, Flasher_Units, Fuses"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Transfer_Case_Parts"/>
<NamedIndividual IRI="#OIL_COOLERS"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Transfer_Case_Parts"/>
<NamedIndividual IRI="#Fuel_Savers"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Transmission_Gaskets"/>
<NamedIndividual IRI="#Lights, Flasher_Units, Fuses"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty IRI="#effect"/>
<NamedIndividual IRI="#Transmission_Gaskets"/>
<NamedIndividual IRI="#OIL_COOLERS"/>
</ObjectPropertyAssertion>
<SubObjectPropertyOf>
<ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
<ObjectProperty IRI="#attached_with"/>
</SubObjectPropertyOf>
<SymmetricObjectProperty>
<ObjectProperty IRI="#attached_with"/>
</SymmetricObjectProperty>
<SymmetricObjectProperty>
<ObjectProperty IRI="#part_of"/>
</SymmetricObjectProperty>
<SymmetricObjectProperty>
<ObjectProperty IRI="#related_to"/>
</SymmetricObjectProperty>
<TransitiveObjectProperty>
<ObjectProperty IRI="#part_of"/>
</TransitiveObjectProperty>
</Ontology>
REFERENCES


