

## **ABSTRACT**

An ontology is a way of information categorization and storage. Web Ontologies provide help in retrieving the required and precise information over the web. However, the problem of heterogeneity between ontologies may occur in the use of multiple ontologies of the same domain. The integration of ontologies provides a solution for the heterogeneity problem. Ontology integration is a solution to problem of interoperability in the knowledge based systems. Ontology integration provides a mechanism to find the semantic association between a pair of reference ontologies based on their concepts. Many researchers have been working on the problem of ontology integration; however, multiple issues related to ontology integration are still not addressed. This dissertation involves the investigation of the ontology integration problem and proposes a layer based enhanced framework as a solution to the problem. The comparison between concepts of reference ontologies is based on their semantics along with their syntax in the concept matching process of ontology integration. The semantic relationship of a concept with other concepts between ontologies and the provision of user confirmation (only for the problematic cases) are also taken into account in this process. The proposed framework is implemented and validated by providing a comparison of the proposed concept matching technique with the existing techniques. The test case scenarios are provided in order to compare and analyse the proposed framework in the analysis phase. The results of the experiments completed demonstrate the efficacy and success of the proposed framework.

## ACKNOWLEDGMENT

First of all, I am gratified to Almighty ALLAH, the most genial, gracious and beneficial who helps me in every moment of my life.

I am deeply thankful to my supervisor Dr. Darren P. Mundy, for his guidance, encouragement, and support throughout the process of this research. I am really grateful to him for his valuable comments and suggestions about the thesis.

I would also like to express my deepest gratitude for the unconditional and constant support, understanding and love that I received from my parents, family and friends.

I would also like to appreciate my fellow students in research room WF1, with whom I spent countless time since last couple of years.

I acknowledge the financial support by the University of Hull, United Kingdom and Faisal Movers, Pakistan. I am also thankful to Muhammad Asif Nadeem who helped me for finding the funds.

Finally, I would like to thank all the people who assisted me in completing this research work.

Qasim Umer

Dedicated To My Grandfather

**HAJI ABDUL GHANI**

## LIST OF ABBREVIATIONS

<b>AI</b>	Artificial Intelligent
<b>ASCO</b>	American Society of Clinical Ontology
<b>ASP</b>	Active Server Pages
<b>CI</b>	Concept Integrator
<b>CM</b>	Concept Matcher
<b>CTXMATCH</b>	Context Matching
<b>DAML – OIL</b>	Agent Mark-up Language – Ontology Interface Layer
<b>FCA-MERGE</b>	Formal Concept Analysis MERGE
<b>IF-MAP</b>	Information Flow Theory
<b>IR</b>	Information Retrieval
<b>JSP</b>	Java Server Pages
<b>KM</b>	Knowledge Management
<b>oMAP</b>	Ontology MAP
<b>ONION</b>	ONtology compositION
<b>OWL</b>	Web Ontology Language
<b>OWL-DL</b>	Web Ontology Language for Description Logic
<b>PHP</b>	Hypertext Pre-processor
<b>RDF</b>	Resource Description Framework
<b>RiMOM</b>	Risk Minimization based Ontology Mapping
<b>SA</b>	Semantic Analyser
<b>SKAT</b>	Semantic Knowledge Articulation Tool
<b>SM</b>	Semantic Matching
<b>SR</b>	Semantic Relation

<b>TA</b>	Taxonomy Analyser
<b>T-TREE</b>	TROPES Taxonomy building Tool
<b>WEESA</b>	Web Engineering for Semantic Web Applications
<b>WWW</b>	World Wide Web
<b>XA</b>	Syntax Analyser
<b>XML</b>	EXtensible Mark-up Language

## Table of Contents

1	Chapter 1: Introduction .....	1
1.1	Brief Examples of the Research Problem.....	2
1.2	Research Problem.....	3
1.3	Aim and Objectives .....	4
1.4	Methodology .....	4
1.5	Research Contribution.....	5
1.6	Overview of the Thesis.....	5
2	Chapter 2: What are Ontologies?.....	6
2.1	Introduction .....	6
2.2	Philosophical Background of Ontology .....	7
2.3	Types of Ontologies .....	9
2.3.1	Top-Level Ontologies .....	10
2.3.2	Domain Ontologies .....	10
2.3.3	Task Ontologies .....	10
2.3.4	Domain Task Ontologies.....	10
2.3.5	Method Ontologies.....	10
2.3.6	Application Ontologies .....	10
2.4	Semantic Web.....	11
2.4.1	Ontologies in Semantic Web.....	13
2.5	Designing an Ontology.....	15
2.6	Populating the Ontology.....	17
2.7	Summary .....	17
3	Chapter 3: Integration of Ontologies.....	18
3.1	Ontology Integration .....	18
3.1.1	Ontology Reuse.....	19
3.1.2	Ontology Mapping .....	20
3.1.3	Ontology Merging.....	21
3.2	Concept Matching Techniques .....	22

3.2.1	Schema-based Concept Matching Techniques.....	22
3.2.2	Schema-based Concept Matching Systems.....	22
3.3	Instance-based Concept Matching Techniques .....	24
3.3.1	Instance-base Concept Matching Systems .....	25
3.4	Hybrid Concept Matching Techniques.....	25
3.4.1	Hybrid Concept Matching Systems .....	26
3.5	Comparison of Existing Matching Tools .....	27
3.6	Limitations of the State of the Art.....	29
4	Chapter 4: Proposed Framework for Semantically Intelligent Ontology Integration .....	31
4.1	The Foundations of the Proposed Framework .....	31
4.1.1	The Concept Comparison should based on Semantics Instead of Terms Used .....	31
4.1.2	The Semantics of the Concepts should be Domain Independent.....	32
4.1.3	The Semantic Similarity Matching should also be based on Semantics of Super Concepts .....	32
4.1.4	The Semantic Relation (SR) of Concepts should be based on their Explicit Meanings .....	32
4.1.5	A Layer Based Approach should be used in Concept Matching .....	33
4.1.6	Considerations Towards a Proposed Framework.....	33
4.2	Motivational Example for Proposed Framework .....	33
4.3	An Enhanced Framework for Ontology Integration.....	35
4.3.1	A Block Diagram of Proposed Framework.....	35
4.3.2	Layer Based Approach of Concept Matching.....	36
5	Chapter 5: Implementation.....	41
5.1	The Comparison of WordNet with other Vocabulary Resources.....	41
5.1.1	Cyc .....	41
5.1.2	WordNet.....	41
5.1.3	HowNet .....	42

5.1.4	Preference of WordNet for the Proposed Framework.....	42
5.2	Implementation.....	43
5.2.1	Brief Overview of the Proposed Framework .....	43
5.2.2	Implementation Tools and Languages .....	44
5.2.3	Pseudo Codes and Algorithms .....	45
5.3	Implementation Limitations .....	51
6	Chapter 6: Experiments and Results of the Implemented Proposed Framework	52
6.1	Methodology .....	52
6.2	Experiments.....	55
6.2.1	Test Case Scenario 1 (as defined in Farooq, 2010).....	55
6.2.2	Test Case Scenario 2 (as defined in Farooq, 2010).....	58
7	Chapter 7: Conclusion and Future Work .....	64
7.1	Conclusion.....	64
7.2	Future Work .....	65
	References .....	67

## Chapter 1: Introduction

Improvements in global communication technologies over the past ten years have resulted in substantial changes in the speed of information exchange. This improvement has transformed the ways in which we work and spend our leisure time. On the other hand, this change has raised problems for information sharing in the field of information technology. These problems have roots in the fact that each enterprise has developed its own applications, protocols, data structures, architectures and interfaces. At present, enterprises have moved towards the concept of embracing globalization and internationalisation strategies. An enterprise needs to increase in resources to expand business either by pooling more resources internally or by working with another enterprise. This perception has introduced the concepts of mergers, acquisitions, joint ventures and partnerships. Independent businesses often have their own knowledge base. Traditionally, organizations only share physical assets in collaboration but now they also need to share and integrate their knowledge. Knowledge sharing is a mechanism, by which an organization obtains access to its own and other organisation's knowledge. The communication of knowledge between organisations has increased through technological advancements (Achterbergh and Vriens, 2002).

The semantic web as proposed by Tim Berners-Lee (1993) provides a solution to the problem of knowledge sharing between enterprise applications over the World Wide Web (WWW) (Berners-Lee et al., 2001). The basic premise of the semantic web is to describe web data in a way that machines can understand i.e. the machine should know what the content means (Michael, 2003; Stumme et al., 2006). According to the main vision of the semantic web, web contents need to be structured, stored and standardized to be understood by machines. The semantic web provides a mechanism through ontologies to structure the data in a way which is machine understandable. An ontology is defined as “a formal, explicit specification of a shared conceptualization” (Gruber, 1993a). Within this definition: formal indicates to the meaning of the specification; explicit refers to explicitly defined concepts, properties and axioms; shared means is machine understandable; and conceptualization refers to how an object/concept is defined in a particular area of interest.

Ontology integration is an approach to exchanging knowledge between enterprises. The process of ontology integration involves the comparison of the concepts of different ontologies. This comparison of the concepts can be made by comparing the syntax (spelling of the terms) of the concepts, semantics (meanings/definitions of the terms) of concepts and/or taxonomy (relationships) of the concepts. For example; the two terms Principle and

Principal are different as both terms do not match on the level of syntax and semantics. Similarly, the Figure 1- 1 also represents the difference in taxonomy of the same concepts in four general purpose ontologies: CYC, WordNet, GUM and Sowa's (Chandrasekaran et al., 1999; Seco, 2005).

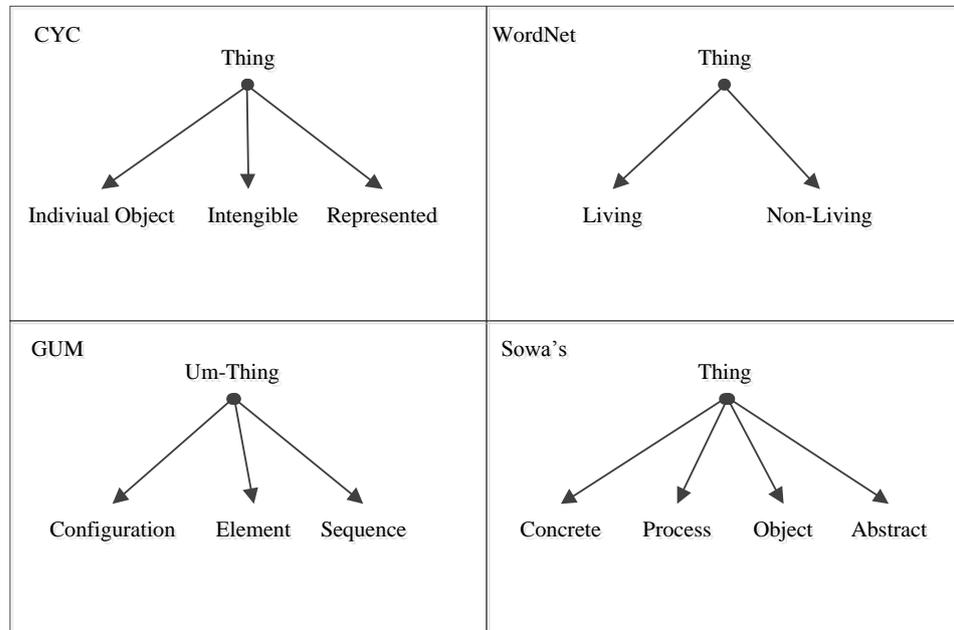


Figure 1-1: Illustration of How Ontologies Differ in their Analysis of the Most General Concepts (Chandrasekaran et al., 1999; Seco, 2005)

In order to integrate the ontologies, concept matching between two ontologies is necessary to overcome the problem of ontology heterogeneity. Over the last couple of decades, a number of researcher have been active in the area of concept matching between ontologies (Le et al., 2004; Li, 2004; Aleksovski et al., 2006a; Aleksovski et al., 2006b; Khodaei et al., 2006; Vernadat, 2007; Cassia et al., 2008). However, the area of ontology integration to solve the heterogeneity problem between ontologies (Pedersen et al., 2007; Bendeck, 2008; Farooq, 2009; Batet et al., 2010; Farooq and Shah, 2010; Shah, 2010) is still open for research.

### 1.1 Brief Examples of the Research Problem

The Figure 1-2 presents two sample Person ontologies, O1, and O2, and their taxonomies. The concepts of O1 and O2 are A1, A2, A3 and B1, B2, B3 respectively. The example can be used to identify important scenarios that have to be considered in ontology integration. In order to integrate O1 and O2, the existing techniques (discussed in chapter 3) use matching algorithms to identify the similar concepts.

These techniques perform concept matching based on instance, schema or hybrid (instance and schema) approaches. The absence of concept semantics (meaning) comparison in the above approaches can produce incorrect results. For instance, the concept Male in O1 does not have any syntactic match in O2. On the other hand, the concept Person in O1 directly matches with the concept Person in O2 syntactically. An intelligent ontology integration approach should understand the linkages between A2 and B2 and A3 and B3. These matches can only be established through an understanding of meaning. The proposed solution should consider the semantics to integrate the concepts which are not matched syntactically but semantically. For example, the concept Male in O1 should be mapped to Boy in O2, and the concept Female in O1 should be mapped to Girl in O2. The proposed solution should also consider the taxonomy of concepts. In the given example, the concept of Male and Boy should map and fall under the concept of Person.

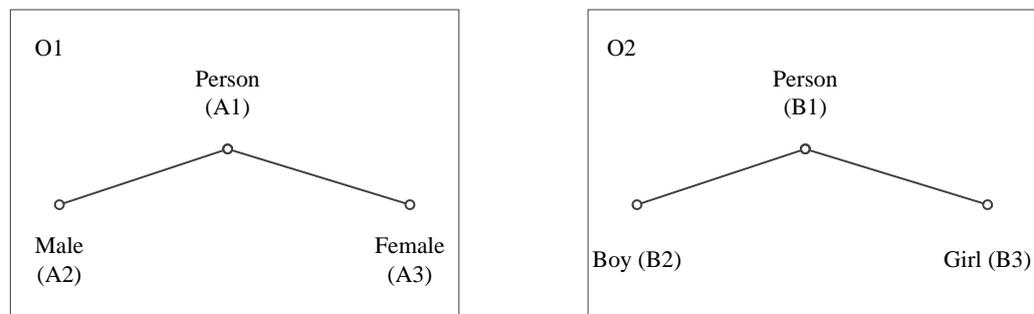


Figure 1-2: Motivational Example

There can be another example of the concepts: clerk and steno. These concepts can be used interchangeably in different ontologies. They can be considered to be semantically similar, but the syntax of these concepts has nothing in common. Therefore, an effective ontology integration process should incorporate the syntax, semantics and relationships. This research introduces an enhanced framework for ontology integration which measures the concept matching by performing syntactical, semantic and structural comparisons between concepts. This should improve the results of concept matching and provide a better solution to the problem of ontology integration.

## 1.2 Research Problem

How to design an efficient, scalable, flexible, and intelligent Semi-Automated Ontology Integration framework that could be applied across different domains?

### 1.3 Aim and Objectives

The primary aim of this research is to investigate the area of ontology matching and deliver a prototypical solution which provides an advancement in the area of ontology integration. This will most likely involve construction of a software platform to deliver a semi-automated way of integrating domain ontologies using a global ontology (Princeton.Edu, n.d.) with minimal user interaction.

The main objectives of the research are to:

- Identify and review approaches to integrate of existing ontologies.
- Analyse the state of the art and the methodologies used to enable ontology integration.
- Understand the particular issues with existing ontology integration methods and applications.
- Propose an enhanced integration framework based on an understanding of the issues.
- Taking a particular scenario (or scenarios) develop a software model which will demonstrate the effectiveness of the identified integration framework.
- Evaluate the efficiency and effectiveness of the implemented framework (Prototype).
- Critically evaluate achievements against the initial research objectives and suggest further work.

### 1.4 Methodology

The philosophy of this research of ontology integration is *positivism*, which leads to the development of a *deductive* approach towards the research. *Epistemologically*, the phenomenon behind ontology integration is to be studied and observed, which can be generalised to some standard rules. Approaches to achieve ontology integration will be evaluated and explored to identify weaknesses. From this evaluation an enhanced integration framework will be proposed and a solution linked to this will be developed as a software prototype. *Experiments* will be designed and conducted for testing the prototype. The results of these experiments will be analysed through *Quantitative* techniques.

Focus will be placed on the provision of a framework to support a dual ontology integration scenario, with the availability of two domain ontologies named as a reference ontologies and the development of a revised ontology. The reference ontologies will remain unchanged and be merged into the revised ontology (adding similar concepts from the reference ontologies to the revised ontology) in the process of ontology integration. This iterative process generates

the integrated ontology as output. The whole process requires pre-processing that involves the generation of taxonomy from the given ontology and the stemming of nodes (concept names) of the tree. The proposed framework uses a top down approach which takes a concept from the reference ontology, and performs concept matching in the revised ontology. Concept matching is based on concept names, concept definition, concept synonyms, a definition of the concept synonyms and the taxonomy of the concept. Concept definition, concept synonyms and their definition are enriched from WordNet (Princeton.Edu, n.d.). The proposed framework is an iterative process which is finished when all the concepts of the reference ontologies are merged into a revised ontology. Finally, the revised ontology will be considered as an integrated ontology.

The proposed framework will be implemented and a prototype will be developed. This prototype will be tested by comparing the proposed framework with existing models of ontology integration. Comparative analysis is conducted to identify the efficiency and effectiveness of the proposed framework.

## **1.5 Research Contribution**

This research has made the following contribution under the context of ontology integration.

- An enhanced framework for ontology integration is introduced in this research, based on the hybrid approach of ontology integration.
- A mixed (instance-based and schema-based) matching approach is developed to support the proposed framework and to achieve better results in ontology integration.
- A multi-level matching strategy is taken for the ontology integration in order to consider the taxonomical nature of ontologies.

## **1.6 Overview of the Thesis**

The remaining thesis is organized as follows. Chapter 2 provides the background knowledge of the problem. Chapter 3 presents the state of the art of different approaches for ontology integration. This is followed by Chapter 4 describing the dependencies, motivational examples, and design of the proposed framework for ontology integration. The overview of the developed prototype and implementation strategies are discussed in Chapter 5, followed by a performance evaluation of the proposed framework in terms of processing time (efficiency) and effectiveness. Chapter 6 discusses the conclusion and future work of the study.

## Chapter 2: What are Ontologies?

This chapter provides a background to the study and reviews other related work. Within this chapter section 1 gives an overview of knowledge management and section 2 introduces the philosophical background of ontologies. Section 3 discusses different types of ontologies. This is followed by section 4 which introduces the concept of the semantic web and explains how ontologies are working in semantic web. Finally, section 5 and section 6 review the design and population phase of ontology development. This chapter concludes by identifying the issues involved in the process of ontology building providing a platform for further exploration in the rest of this thesis.

### 2.1 Introduction

Human thinking and decision making is based on the things one may encounter in daily life. Most of the time, human perception is based on either what an individual may have experienced, heard, read, seen or learnt. On the basis of perception, appropriate decisions are made. The human mind categorises these perceptions as data, information, knowledge or understanding. There are two sources that the brain uses to build knowledge - data and information. Data are the facts and information refers to data which has been given some meaning through being processed in relation to a particular context. Knowledge is familiarity with the rules of the world and their application to the information that a human may learn through study, observations and personal experiences.

In order to illustrate this, if we take an example of temperature readings of 10°C, 8°C, 9°C, 11°C and 7°C over 5 summer days in Scarborough at 9am in the morning. 8°C (e.g. a single day's reading) would be an example of a piece of data. Information could be that on average over the five days the temperature was 9°C, e.g. the processing of the 5 pieces of data to determine the average. Knowledge could involve an individual in understanding temperature variations over time in order to predict the next five days temperature.

Knowledge management (KM) issues can centre on areas such as knowledge capture, knowledge storage and knowledge dissemination. Finding solutions for issues in these areas can help to play a vital role in problem solving. The role of knowledge within an enterprise is to generate suitable actions by providing a background to the problem, give an understanding of the possible actions, and to predict the possible outcomes.

Examining literature (Achterbergh and Vriens, 2002; O'leary, 2002) related to the knowledge lifecycle elicits four main lifecycle stages, these are:

- Generation – Where knowledge is created.
- Sharing – Where knowledge is transferred from one entity to another.
- Retention – Where knowledge is stored.
- Usage – Where knowledge is used by the entity according to a particular situation.

However, the knowledge lifecycle in knowledge based systems involving Artificial Intelligence (AI) may also include phases related to validation and integration (Birkinshaw and Sheeha, 2002). Knowledge capture, representation, processing and reuse are the key areas for knowledge based systems (Gruber, 1993a). Therefore, KM is very important to make knowledge based systems efficient and resourceful. The use of ontologies is one way of achieving this. In the context of Information Retrieval (IR), an ontology is defined as “the set of representational primitives, which model a domain of knowledge or discourse” (Özsu and Liu, 2009).

## **2.2 Philosophical Background of Ontology**

The Bailey's dictionary (1721) shows the first existence of the word “Ontology” in the English Language. It defined ontology as “an account of being in abstract” (Smith and Welty, 2001). The word ontology has its roots in the field of philosophy. Ontology is the philosophical study of the nature of being and existence (Gruber, 1993b; Özsu and Liu, 2009). Ontology is said to be an important branch of metaphysics: the study of most general features of reality (Sandholm, 2005). Generally, there is not a single meaning of any word in metaphysics; each object can have a different meaning dependent on different fields of interest. Similarly, meanings of the word “ontology” depend on what the ontology is for. In philosophy, an ontology is used as a theory of the nature of existence (Özsu and Liu, 2009). It tries to answer questions such as “what is existence?”, “What properties can explain existence?”, etc.

Among many of the philosophers, Aristotle was a pioneer for having a realist's view of the meaning of ontology. In his work on 'Categories', he defined different categories that represent most general kind of things. These categories are used to differentiate things and to define the attributes of things. Aristotle in 350 B.C. formulated a structure for grouping things and their attributes according to the following categories:

- Substance (e.g., man, horse)
- Quantity (e.g., two cubits long, three cubits long)
- Quality (e.g., white, grammatical)
- Relation (e.g., double, half, greater)
- Place (e.g., in the Lyceum, in the market-place)
- Time (e.g., yesterday, last year)
- Position (e.g., is lying, is sitting)
- State (e.g., shod, armed)
- Action (e.g., to lance, to cauterize)
- Affection (e.g., to be lanced, to be cauterized)

(Sandholm, 2005)

Aristotle also guided and suggested further possible research for the categorisation and classification of objects (Sandholm, 2005). On the basis of Aristotle's suggestions, Franz Brentano (Styltsvig, 2006) organized these categories as nodes of a tree as shown in Figure 2-1.

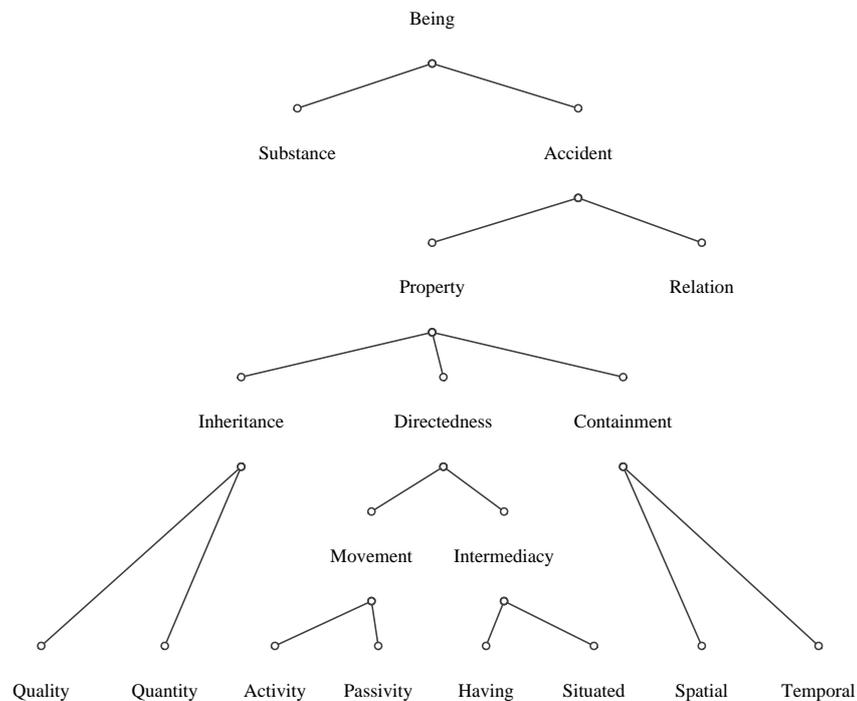


Figure 2-1: Brentano's Tree of Aristotle's Categories (Styltsvig, 2006)

In the philosophical context, an ontology is something which has to establish the reality by defining the concepts and their relationships with each other (Seco, 2005).

In the context of knowledge sharing ontology has been defined as “a specification of a conceptualization” (Gruber, 1992). Also within this context ontology is defined as “a description of the concepts and relationships that can exist for an agent or community of an agent” (Gruber, 1992). This helps in knowledge sharing and reuse. However, an ontology describes the semantics of available data in the field of computer science (Seco, 2005). Chandrasekaran (1999) has defined ontology as a representational vocabulary in a specific domain (Chandrasekaran et al., 1999). Ontology cannot define itself as a complete package of information, but it can carry instances and parts of information which may help in defining any particular domain (Sampson, 2005). In any communication process, ontologies can help and support the explanation of the actual meanings of messages between a sender and receiver (Wang and Hongshuai, 2010). At the present time, the goal of the research community is to make the semantic web a reality, ready to be applied to within industrial contexts.

Hierarchies and taxonomies are the most natural forms of knowledge representation. Perhaps this is why researchers in the field of AI have developed ontologies to assist in knowledge sharing and reuse (Fensel, 2001). Ontologies are gaining popularity, mainly because they promise: a shared and common understanding of concepts between entities (Fensel et al., 2001). Sometimes, the understanding of the meaning and the usage of the concepts within all possible contexts seems to be an abstract exercise. Many researchers and philosophers (Sandholm, 2005) have presented constructive arguments in relation to ontology production and implementation. These arguments provide a greater understanding in discovering the agreed meaning of the same concept between different entities. The field of computing can aid in developing systems which can perform the task of ontology development by impersonating the human way of acting and thinking.

### **2.3 Types of Ontologies**

The following sections detail the six different types of ontologies which are defined on the basis of information that ontologies can capture (Gómez-Pérez et al., 2004).

### **2.3.1 Top-Level Ontologies**

Top-Level ontologies are the most general ontologies and can also be called global ontologies. These ontologies work in parallel with domain specific ontologies to deliver an understanding of the meaning of general purpose concepts. WordNet is an example of a Top-Level ontology.

### **2.3.2 Domain Ontologies**

Domain ontologies contain domain specific knowledge from areas such as medicine, politics, etc. Common examples include the UMLS Meta-thesaurus defined by the U.S.

National Library of Medicine for the medical domain (Pisanelli et al., 1999; GómezPérez et al., 2004).

### **2.3.3 Task Ontologies**

Task specific ontologies are general purpose task ontologies, for example; a transactions or a diagnosis. These ontologies relate directly to top level ontologies to specify only information about tasks.

### **2.3.4 Domain Task Ontologies**

Domain task ontologies give the knowledge about specific tasks in relation to a particular domain by taking the information from task ontologies and domain ontologies, for example; bank transactions or MeSH, stands for Medical Subject Heading part of the UMLS Meta-thesaurus defined by the U.S. National Library of Medicine for the medical domain.

### **2.3.5 Method Ontologies**

Method ontologies basically define the relevancy of concepts and relationships by specifying the reasoning process.

### **2.3.6 Application Ontologies**

Application ontologies are defined to represent the knowledge in a specific application.

Ontologies provide the underlying infrastructure for information to be constructed and made available within the semantic web to enable complete and shareable machine understanding. Therefore, precise ontology development is essential for the success of the semantic web (Patel-Schneider and Fensel, 2002).

## 2.4 Semantic Web

The word „Semantics“ originated from a Greek word (semantikos), which means “significant meanings”. Therefore, the study of semantics is a scientific and philosophical study of meaning in natural and artificial languages (Lepore, 2011). Over the past ten years there has been and continues to be a tremendous growth in the World Wide Web (WWW). This growth has led to difficulties in the retrieval of desired information in a global context. This difficulty results in performance issues as users (including automated programs) have to sift through a variety of often irrelevant information. How this problem has arisen can be easily understood by observing the short history of the Web.

The WWW has taken a trip from Web 1.0, to Web 2.0, and now to Web 3.0 also known as the semantic web (W3C, n.d.). According to Tim-Berners Lee (1998), who is the creator of the WWW, “The dream behind the Web...[was]... of a common information space in which we communicate by sharing information” (Berners-Lee, 1998). Web 1.0 was an interpretation of this dream and could be considered as the “read-only web” (Berners-Lee, 1998). The first version of the WWW was a more static construction and provided limited user interaction and content contribution (Cormode and Krishnamurthy, 2008; Kinsella et al., 2008). The information was created by the creator/designer of the web pages and the users of Web 1.0 could only read this information. The non-dynamic nature of Web 1.0 was not only based on the kind of web pages used, but was also inherent within the framework and infrastructure through which the pages were served. Web 1.0 was basically slow wherein each web page was required to be refreshed to check for updates (Evans, 2006) and session information was not maintained very well. Shopping cart applications fall under the categories of Web 1.0.

Web 2.0 was envisaged by Tim O’Reilly as a way of representing the evolution of web based services into a read-write space. According to the Tim Berners-Lee’s point of view, the second version of the web introduced content contribution and user interaction with each other. It has changed the landscape of the web in a short time. Youtube, MySpace and Google are examples of web 2.0 (Evans, 2006; Anderson, 2007) which depend on user generated data. Web 2.0 has a framework which gives it openness, user control and standards along with new technologies and applications (Anderson, 2007; Futureexploration.Net, 2007). However, the introduction of Web 2.0 had generated a lot more data than what was ever expected. This data is needed to be moderated as there is a lot of exploitation in terms of loads of advertisements (Evans, 2006) and pop-ups. Another problem in Web 2.0 is inconsistent and non-generic tagging by users and mashups of applications (Gruber, 2006), especially within

the context of social networking websites which allow users to tag anything. From a security, scrutiny and ethical perspective there is a need for moderation of user generated content (whether by a site owner or by the community themselves). Tim Berners-Lee (2001) came up with the concept of the Semantic Web (Web 3.0) which is an extension of Web 2.0 providing a better definition and structure of data as a solution for the problems within the Web 2.0 context (Berners-Lee et al., 2001; Cormode and Krishnamurthy, 2008).

The goal of the Semantic Web is that of machine understand-ability leading to automated decision making. In order to accomplish this goal all the data on the WWW whether textual or multimedia should be semantically tagged with relevant metadata. The ability of machines to process web resources can be achieved through more specific explanation of resource content with a semantic mark-up called meta-data (Aufaure et al., 2008). The metadata helps the software agents to understand the data and make decisions by utilizing it (Amardeilh, 2009).

Concepts such as contextual advertisements, intelligent tagging, relatedness and relevance between different tags have all arisen in relation to the Semantic Web (Kinsella et al., 2008). The dream of the Semantic Web is to make global data, machine understandable and enable machine to take decisions on behalf of human beings. Semantic web technologies help to separate meanings of data, document content or application code, using technologies based on open standards (Balani, 2005). Stumme and others in (2006) defined four steps to understand the directions of activity in the Semantic Web: providing a common syntax for machine understandable statements, establishing common vocabularies, agreeing on a logical language and using the language for exchanging (Stumme et al., 2006).

The W3C has also stated that the Semantic Web is providing elements such as: the foundations of linking and embedding data; standards of defining and exchanging rules and data; presenting possibilities of enriching data with meanings (semantics); and defines standards of querying the data (W3C, n.d.). The Semantic web provides steps towards the maximum automation of knowledge life cycle processes i.e., knowledge representation, knowledge acquisition, adaption, reasoning, sharing and use (Achterbergh and Vriens, 2002; O'leary, 2002). The Semantic Web needs ontology (web ontology) in which the terms are described, to provide the common understanding of terms used in the meta-data for agents/applications.

### 2.4.1 Ontologies in Semantic Web

The layered framework of the Semantic Web (as shown in Figure 2-2) contains an Ontology layer sitting in the middle of the structure just above the Resource Description Framework (RDF). RDF is a standard model of data interchange which extends the linking structure of the Web (W3C, 2004). RDF helps web resources to annotate semantic information using a conceptual model (schema) (Nejdl et al., 2000).

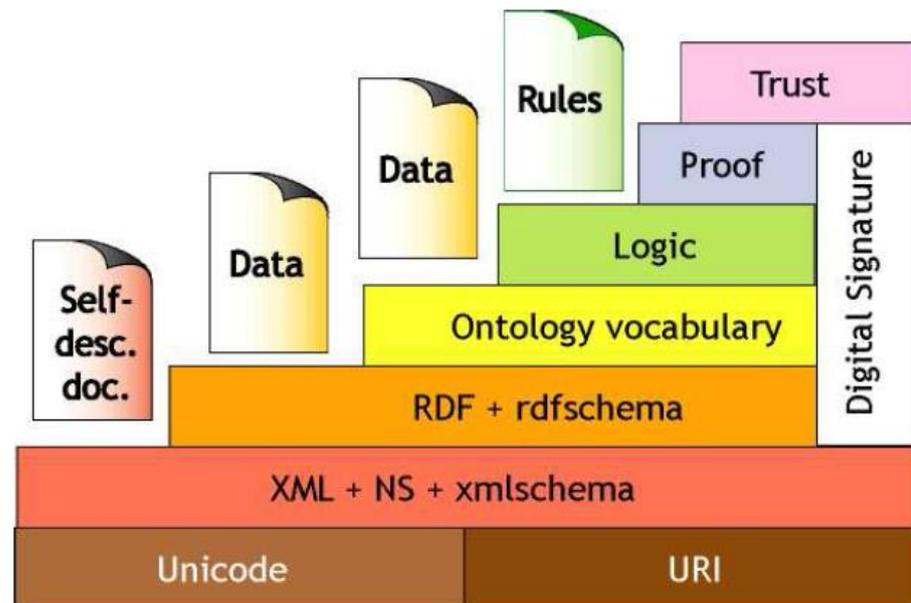


Figure 2-2: Layered Model of Semantic Web (Patel-Schneider and Fensel, 2002)

The RDF schema is used in an ontology by using different ontology definition languages like Web Ontology Language (OWL), Web Ontology Language for Description Logic (OWL-DL) and DARPA Agent Mark-up Language – Ontology Interface Layer (DAML – OIL) which are standardised by the W3C (Daml-Oil, 2001). The languages provide different syntax for the design of web ontologies but the main components of a web ontology described below are similar.

#### 2.4.1.1 Components of an Ontology

The following Figure contains an example of how an ontology can be constructed around a particular concept; in this case the concept is Vehicle. Descriptive content which follows outlines the main components of an ontology as linked in to this example case.

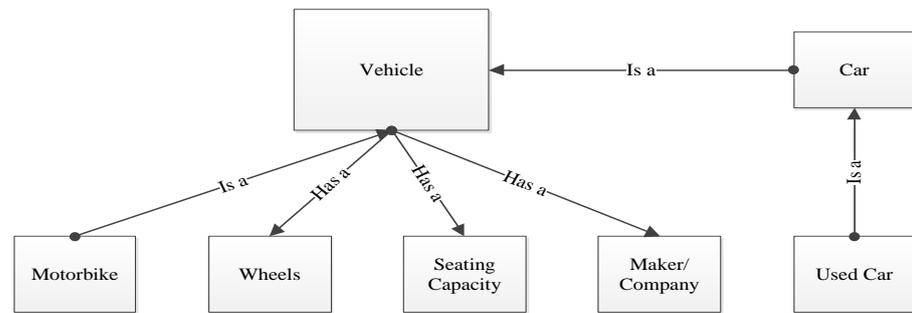


Figure 2-3: Description of Vehicle Ontology Components

#### 2.4.1.1.1 *Classes or Concepts*

Classes are the concepts of domain or task that are arranged in taxonomies. For example; Car, Wheels, Seating Capacity, etc. are the classes of Vehicle ontology.

#### 2.4.1.1.2 *Roles or Properties*

Role is a type of relationship between instances of concepts in the particular domain. For example; has-wheels, has-seating capacity and has-maker are role of Vehicle ontology.

#### 2.4.1.1.3 *Axioms*

The sentences that are always true represent axioms. For example, any car used over 2000 miles has to be below 50% of the value of a new car of the same model.

#### 2.4.1.1.4 *Individuals or Instances*

Anything that refers to specific classes of ontology is considered individual or is an instance. For example, Toyota can be an instance of Maker class of Vehicle.

The ontology development process includes understanding of the domain, identification of the required concept in that particular domain, and the relationship between concepts. It is followed by the identification of terms which best describe the concept and their relationships. The three approaches of building ontologies are (Gómez-Pérez et al., 2004):

- **Top-Down Approach** – starts with the most abstract concepts and then drills them down to more specific concepts. This approach provides good control over the level of detail, although there might be mistakes in choosing the top level abstract concept. This gives rise to instability of the model and requires a reworking of levels.

- **Bottom-Up Approach** – tries to identify the more specific concepts and then goes on to generalize them to more abstract concepts. According to (Gómez-Pérez et al., 2004) this approach requires a lot of effort to identify the relationships, which might lead to inconsistencies and rework.
- **Middle-Out Approach** – identifies the core terms and then specification and generalization is carried out in both directions. Researchers (Uschold and King, 1995) appreciated this approach due to its stable nature.

The current framework of development of ontologies is just like databases; first a schema is generated and populated. Fensel (2001) explained ontology research to be the database research of the 21st century (Fensel, 2001). Initially, HTML was the language of the WWW and it gave a standard to structure the documents. HTML was very simple which gave the WWW a very rapid growth. However, at the same time users required complex and advanced web applications. This provided initiative for the development of Active Server Pages (ASP), Java Server Pages (JSP), PHP: Hypertext Pre-processor (PHP), etc. EXtensible Mark-up Language (XML) was one of the extensions which let developers define arbitrary domain-specific and task-specific applications (Fensel et al., 2001). XML is a language which gives the serialized syntax for tree structures (Fensel et al., 2001). It requires a data model and framework for data and metadata on the WWW, such as RDF. This provides the foundations for interoperability of data and information on the WWW between different applications (W3C, 1999). It is based on a basic triple: object; property; and value. RDF is further extended within RDF Schema (RDFS), which provides the means to define vocabulary, structure and constraints about the metadata and other web resources (Broekstra et al., 2002). RDFS does not provide a complete representation of knowledge and it is not a knowledge representation language (Broekstra et al., 2002). RDFS needs some more extension although it models classes, properties and constraints in a web based context (Fensel et al., 2001; Broekstra et al., 2002) but, it requires some standardised language to define ontologies.

## 2.5 Designing an Ontology

Before the concept of the Semantic Web the ontologies are developed but less attention was paid towards the design phase of ontology (Reif et al., 2004; Cristani and Cuel, 2005). On the whole, methods try to map XML Schema onto Ontologies (Reif et al., 2004) or use the Object Oriented Model or Web Design Models (Plessers et al., 2005) and have not concentrated on the design of ontology (Gómez-Pérez and ManzanoMacho, 2003; Cristani and Cuel, 2005). There is not a standardised way to map schema. Therefore, direct mapping generates errors in

ontology design. In order to provide error free mapping for the design phase of ontology, many frameworks of mapping have been identified. Some of the most important frameworks related to ontology design are discussed below.

Ushlod and King (1995) proposed a method which starts by identifying the purpose of an ontology, followed by the building of the ontology in three steps: capturing the ontology, coding the ontology and integrating the ontology (Ushold and King, 1995).

Later in 1997, Fernandez, Gomez-Perez, and Juristo gave a life cycle of ontologies to build ontologies from scratch. They used the models of Software Engineering to describe ontology engineering in terms of concepts, relations and properties (Lopez et al., 1997).

Later, on the basis of Gruber's view (1995), Holsapple (2002) had defined ontology as "an explicit specification of an abstract, simplified view of world we desire to represent (Thomas, 1995; Holsapple and Joshi, 2002)" and mapped ontologies on knowledge based systems. Holsapple (2002) used a life cycle based on the specification of criterion, boundaries, conditions and standards for concepts, followed by formalization and production of the ontology based on an iterative process (Holsapple and Joshi, 2002). Pinto and Martins (2004) defined another iterative lifecycle focused on defining specifications for all the concepts, conceptualizing, formalizing, implementing and maintaining them (Pinto and Martins, 2004).

Farooq and Shah (2008) followed Holsapple's method for ontology development. They adopted a practical and simple approach for development activities along with software engineering standards (Farooq and Shah, 2008). Farooq, Shah and Asif (2007) proposed a technique for ontology design by using existing web application design methods (Farooq et al., 2007).

The current web is primarily for human users whereas the semantic web also includes machines as target users. Therefore, web engineering and the semantic web are different processes. In order to differentiate both processes, Kumar and others (2010) have defined the ontology development activities in five phases: Analysis, Design, Implementation, Integration & Testing and Documentation & Evaluation (Kumar et al., 2011). This life cycle is very much similar to that of web engineering but each phase is designed by keeping the ontologies in mind. In order to capture all the requirements to develop an ontology, a preliminary web ontology model should be prepared at the analysis phase. An RDF Model is prepared in line with the preliminary web ontology model. Ontology is codified in RDF and OWL language

during the implementation phase. The integration phase integrates the output of web engineering standards and ontology construction standards. Thereafter, testing, documentation and evaluation phases are carried out (Kumar et al., 2011).

## **2.6 Populating the Ontology**

Ontology design and population are two phases of ontology development as discussed above; however, ontology population is a very time consuming and tedious job. An automatic way of ontology population is a very active area of research (Celjuska and Vargas-Vera, 2004; Farooq, 2009; Hyun-Je et al., 2009; Xia and Wang, 2009). Due to the tedious nature of ontology population, most researchers have used either the direct mapping method of XML schema or already build software.

Reif, Jazayeri and Gall (2004) have introduced an approach to map XML Schema to ontologies, called Web Engineering for Semantic Web Applications (WEESA) (Reif et al., 2004). In the implementation phase the RDF description is based on XML content pages (Gerald et al., 2005; Reif and Gall, 2006). XML based mapping of ontologies has also been proposed by Farooq, Shah and Akram in 2006 (Farooq et al., 2006) and later, in 2007 another similar work was presented by them (Farooq, 2009). In 2008, Farooq also proposed a model of Ontology Development and defined a formalization phase for ontology coding and population (Farooq and Shah, 2008). Farooq proposed to use the available tools for ontology creation like Protégé-2000 (Farooq and Shah, 2008). Similarly, Zhang and Olango (2005) proposed a way of automatic population of ontology by using an open source tool Protégé (Zhang and Olango, 2005). They used Wikipedia pages and have extracted information by using XML tags, XSLT or Xquery (Valarakos et al., 2004).

## **2.7 Summary**

The review of different approaches to ontology design and population specifies that the manual design and population of ontologies requires significant resource in terms of time and labour. There are also some issues regarding the potential generation of redundancy of concepts and inconsistency between ontologies of the same or similar concepts. Ongoing research across multiple institutions continues to focus on reducing problems in the design and population of ontologies. The aim of a substantial amount of this research is focused on automation of the design and population of ontologies (Keet, 2004; Lee et al., 2006; Simon et al., 2006). The next chapter reviews existing approaches of ontology integration and represents how existing ontologies can be used/reused in the process of ontology integration.

## Chapter 3: Integration of Ontologies

This chapter presents background research in ontology integration and reviews the main approaches to ontology integration. Within this chapter the first three sections provide an overview of the techniques of ontology reuse, ontology mapping, and ontology merger which are the approaches to ontology integration. This is followed by Section 4, which reviews the concept matching techniques, which are considered as prerequisite for any approach to ontology integration. This chapter concludes by identifying the issues related to existing approaches in ontology integration.

### 3.1 Ontology Integration

Ontology integration is a process of building new ontologies by using available ontologies through an ontology development environment. Ontology integration can be broken down into the following three different scenarios, as defined by Pinto in 1977 (Farquhar et al., 1995).

- **Ontology Reuse** – Where already available ontologies are reused (Farquhar et al., 1995; Borst and Kkermans, 1997). For example; suppose there is a need to build an ontology of Public Transport that includes the information of Cities, Roads, Population distribution. This ontology can be built from scratch. However, this requires first analysis and design of the system in terms of cities, roads, etc. and then codification of the ontology. This involves a lot of time and can cause some of the different problems discussed in chapter 2. Another approach is to build the Public Transport ontology by reusing available ontologies within the particular domain, such as ontologies for Cities and Roads
- **Ontology Mapping** – Where the integrated ontology contains the rules which maps the concepts between ontologies which are to be integrated (Alasoud, 2009). For example; a company running many offices at different places having different operations e.g. show rooms, go-downs and service centres. The company head office requires an overarching ontology in order to take decisions. In this case an ontology would be generated from integration of all the branch ontologies. This would provide a unified view for head office to answer all the queries related to different branches.
- **Ontology Merger** – Where a new ontology is built by merging different ontologies (Alasoud, 2009). For example; there can be a need of finding a unified single ontology of all online medical journals. There might be ontologies having overlapping information about medical journals which requires the unification.

The following sections of this chapter provide a detailed overview of ontology reuse, ontology mapping and ontology merging.

### 3.1.1 Ontology Reuse

Within the context of ontology reuse, an existing ontology may be used as a platform from which to construct a new or extended ontology. The new or extended ontology would contain the existing ontology plus the addition of further concept(s). The original ontology would remain unchanged. This process is not considered as proper ontology integration, but a lot of developments have been made under the title of using or reusing ontologies since the survey done by Pinto in 1999 (Pinto, 1999). A software toolkit was developed for ontology development, reuse, and maintenance in the On-To-Knowledge Project (Fensel et al., 2000). Stumme and Mädche (2001) proposed a method of reuse and merging ontologies based on the merging of global ontologies, which were similar to global databases (Stumme and Maedche, 2001).

The purpose of ontology reuse is to build a new ontology through using existing ontologies as an alternative of building an ontology from scratch (Pinto, 1999). Figure 3-1 demonstrates the two existing ontologies A, B and the integrated ontology O, where reused ontologies A and B are part of O. It is generally considered that the domains of the existing ontologies A and B can be different from the domain of O. For example, Ontology “Road” and ontology “Hotel” can be reused to build the Ontology “Tourism”.

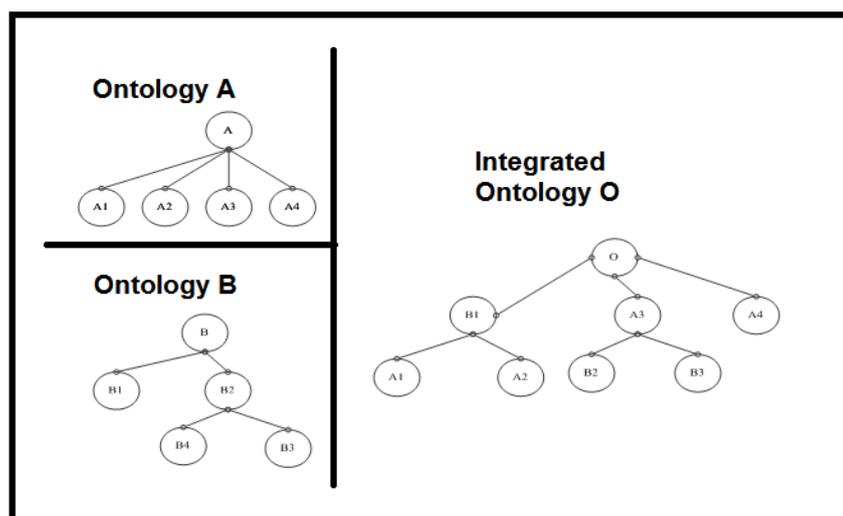


Figure 3-1: Ontology Reuse

In the process of integration with a reuse ontology approach, the concepts from existing ontologies may be taken according to the following four ways: without any change in definitions, with some modification in definitions, without any change in hierarchy (super-concept, sub-concept or sibling-concept relationships) or with some modification within the hierarchy (super-concept, sub-concept or sibling-concept relationships). For example; the domains of Cities and Roads are different but may be reused in the development of a Public Transport ontology. Figure 3-1 also represents that a concept B4 can be deleted or a new concept O can be added in the integrated ontology. Therefore, the selection of the existing ontologies should be closely matched to the requirements of the integrated ontology (for example, be in closely related domains) to achieve good results. Similarly, integrated ontologies should be clear and concise.

### 3.1.2 Ontology Mapping

Ontology Mapping is another approach which is used and proposed by many researchers (Kalfoglou and Schorlemmer, 2003; Noy, 2005) for ontology integration. In ontology mapping, the integrated ontology O contains the rules of mapping concepts between existing ontologies A and B. In the process of ontology mapping, generic rules are defined for the concepts contained within the integrated ontology O. Each concept from the existing ontology A is taken, compared and associated with the concepts of the existing ontology B (Ehrig and Staab, 2004). Ontology mapping basically provides an easy way to access and exchange information between ontologies by providing a common layer (Kalfoglou and Schorlemmer, 2003).

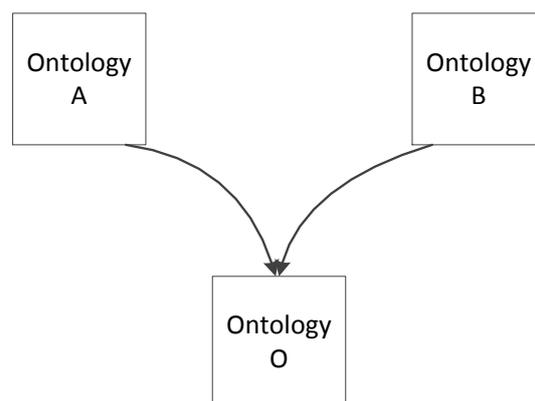


Figure 3-2: Ontology Mapping

Wache and some other researchers (2001) have categorised approaches to ontology mapping into the following three categories (Wache et al., 2001) as shown in Figure 3-2.

- **Single Ontology Approach** combines all the shared concepts of the existing ontologies in the integrated ontology.
- **Multiple Ontology Approach** maps the concepts of existing ontologies without building an integrated ontology.
- **Hybrid Ontology Approach** maps the concepts of existing ontologies, but a common vocabulary is also built to make the existing ontologies comparable. The common vocabulary contains the basic concepts of the domain.

### 3.1.3 Ontology Merging

Ontology merging is another approach to combine different ontologies to create a unified ontology (Pinto, 1999). In the process of ontology merging, a new ontology is built in the same domain and includes unified concepts, terms, definitions, etc. from existing ontologies. This process involves at least two existing ontologies A, B and the integrated ontology O as shown in Figure 3-3, where the domains of the existing and integrated ontologies are the same. The integrated ontology O is a more general integrated ontology created by using the gathered knowledge from the existing ontologies A and B of the same domain.

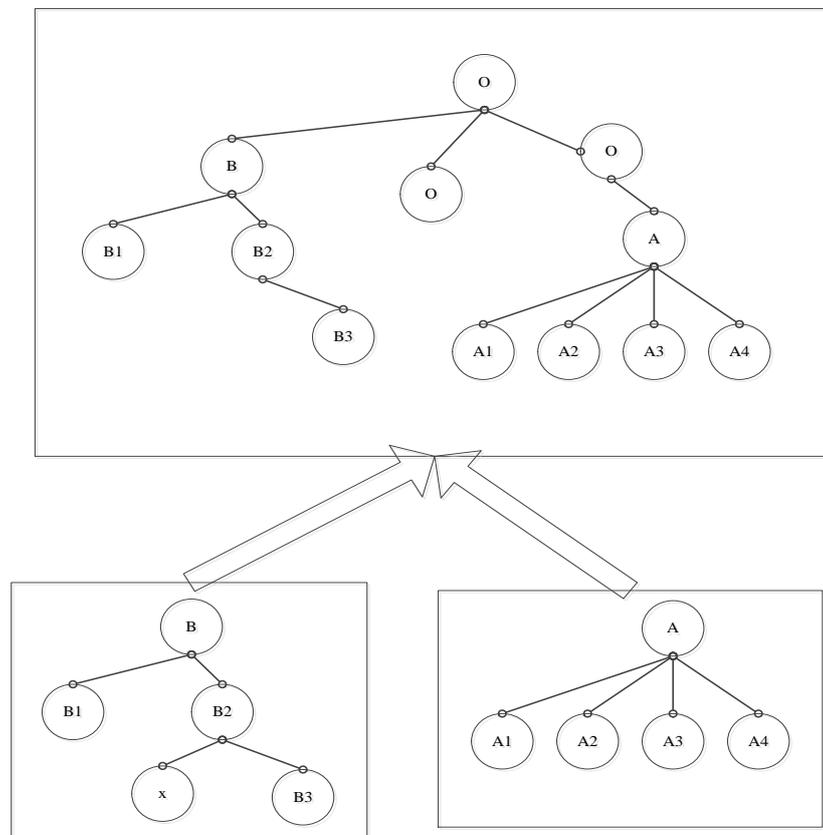


Figure 3-3: Ontology Integration

## 3.2 Concept Matching Techniques

At the foundation of ontology integration is the very important task of concept matching. This section reviews existing techniques to concepts matching. Several techniques are used to measure the similarity of concepts between two or more ontologies for concept matching. These concepts are generally matched on the basis of schema, instance or a hybrid (schema and instance) (Alasoud, 2009; Farooq, 2009; Farooq and Shah, 2010).

### 3.2.1 Schema-based Concept Matching Techniques

In schema based techniques similarity is measured at the structural level. Matching at the structural level can be performed in two ways:

- **Graph-based concept matching** transforms the existing ontologies into a labelled graph. This approach suggests that where you have nodes (x and y) considered to be similar across different ontologies, that neighbouring nodes to x and y, should also be somehow similar (definition, type or axiom etc.) (Euzenat et al., 2004).
- **Relationship-based (Taxonomy-based) concept matching** considers only the specialised relation. This technique proposes that if the nodes (x and y) are considered to be similar across different ontologies, and the nodes also have the same relationship such as an „is-a“ relationship, then the nodes and their super concepts can also be considered similar (Euzenat and Valtchev, 2004). The taxonomy-based approach of concept matching is the most common approach used, as ontologies are the taxonomised view of conceptualisation.

### 3.2.2 Schema-based Concept Matching Systems

The following are existing examples of systems using schema based techniques and tools for concept matching.

#### 3.2.2.1 Semantic Knowledge Articulation Tool (SKAT)

SKAT which is based on simple lexical and structural matching was introduced by Mitra and others in 1999 (Mitra et al., 1999). It takes input ontologies in the form of object graphs, and gives concept matching results through a semi-automatic process on the basis of rules (based on structure, terms and context, or explicitly provided by domain experts) (Mitra et al., 1999). A domain expert is required for the confirmation of the similar concepts.

### **3.2.2.2 ONtology compositiON (ONION)**

ONION, a graph oriented model ONION, is an extension of SKAT by Mitra (2000) (Mitra et al., 2000). It enables the integration of ontologies with the help of expert rules and external semantic dictionaries like WordNet (Mitra et al., 2000). It was a semi-automated system as experts were needed to approve, delete or modify the definitions and suggestions given by WordNet. It provides the comparison based on syntax in the first stage. It does not compare the concepts semantically if the concepts are syntactically matched.

### **3.2.2.3 PROMPT**

PROMPT algorithm formally known as SMART, uses a semi-automated approach to ontology merging and alignment. It is a general purpose system which guides the user about the ontology integration, suggests possible actions, determines conflicts and proposes solutions for these conflicts (Noy and Musen, 2000). PROMPT needs to integrate more heuristic rules to make it more efficient.

### **3.2.2.4 Anchor-PROMPT**

Anchor-PROMPT - an extension of PROMPT, automatically finds semantically similar terms in the ontologies, and merges them, while considering the paths connecting these concepts/terms. Anchor-PROMPT treats an ontology as a graph. The algorithm analyses the paths in the ontologies and determines which concepts frequently appear in similar positions on similar paths (Noy and Musen, 2001). This system does not work properly if the ontologies are built using different modelling techniques.

### **3.2.2.5 Context Matching (CTXMATCH) Algorithm**

CTXMATCH is an algorithm involving Scholastic Aptitude Test (SAT) based logical analysis for the ontology matching problem. It determines logical relationships between concepts and properties. It uses a WordNet based approach to understand linguistic and contextual information for labels (names of concepts) (Bouquet et al., 2003a). This algorithm is dependent on WordNet and the structure which is introduced in this algorithm for the ontologies.

### **3.2.2.6 CTXMATCH2**

CTXMATCH2 is an extension of work earlier explained, but it uses WordNet Description Logic and reasoning to represent structural meaning or any additional constraints (axioms) for domain knowledge (Bouquet et al., 2006).

### 3.2.2.7 American Society of Clinical Ontology (ASCO) Algorithm

ASCO is an algorithm for matching and comparing two ontologies based on all their concepts, relations or structure using WordNet or Euro WordNet (Le et al., 2004). This algorithm is established on RDF(S) formalism.

### 3.2.2.8 ASCO2

ASCO2 is the new version giving a similarity measure for comparing concepts of ontologies defined in OWL (Bach and Dieng-Kuntz, 2005).

## 3.3 Instance-based Concept Matching Techniques

The instance-based concept matching approach considers similarities between ontologies using the actual concepts instead of their relationships with other concepts. This technique is divided into the following methods:

- **String-based concept matching** considers the names of concepts in ontologies for matching. In this technique the similarity between the names of concepts is calculated on the basis of string matching. The five most commonly used methods of string matching are: (1) Edit-Distance – the distance of matching concepts from the root is calculated in order to reduce the cost of operation (Noy and Musen, 2000; Noy and Musen, 2001); (2) Normalisation – different string operations, for example; blank normalisation and punctuation elimination are performed before matching concepts to improve the results of concept matching; (3) String equality – a method that compares the strings and results as true or false; (4) Sub-string test – a matching of strings in which a comparison of substrings is also considered; and (5) Token based distance – a method that considers tokens (small word groupings) within the string of multiple words.
- **Language-based concept matching** is used to find out the relatedness of concepts where the concepts are taken as words in any national language. Natural Language Processing (NLP) techniques are used to identify the meaning of the words used for the concept, based on the linguistic relation between words e.g. synonyms. WordNet is an example implementation tool for use in language-based concept matching.
- **Constraint-based concept matching** considers the definition of concepts to identify the similarity between concepts, for example; type comparison, attribute comparison and domain comparison etc.

### **3.3.1 Instance-base Concept Matching Systems**

The below sections contain a description of important systems which make use of instance-based concept matching.

#### **3.3.1.1 TROPES Taxonomy building Tool (T-TREE)**

T-TREE is an extension of an existing system TROPES (an object based knowledge representational system). T-TREE constructs a taxonomy of objects using Tropes and integrates data analysis to automatically classify the object taxonomy. This system introduces the idea of bridges for the comparison and connection of object taxonomies (Euzenat, 1993).

#### **3.3.1.2 CAIMAN**

CAIMAN is a system to exchange documents between geographically dispersed people. Everyone has organized one's documents according to their own categorization scheme (ontology). CAIMAN performs ontology mapping, using machine learning techniques for text classification. It maps personal ontologies (categorization) onto the development of a community ontology (Lacher and Groh, 2001).

#### **3.3.1.3 Formal Concept Analysis MERGE (FCA-MERGE)**

FCA-MERGE is another method of merging ontologies which follows a bottom-up approach within the merging process. The FCA-MERGE method consists of three stages: 1- Linguistic Analysis and Context Generation; 2- Generation of the Pruned Concept Lattice; and 3- Generation of the new Ontology (Stumme and Maedche, 2001).

#### **3.3.1.4 GLUE**

GLUE ontology mapping system contains three steps: a Distribution Estimator, a Similarity Estimator, and a Relaxation Labeller. In the first step, it estimates the joint probability distribution then estimates the similarity between ontologies and their instances. GLUE finds the best matches in the last step (Doan et al., 2003).

### **3.4 Hybrid Concept Matching Techniques**

In this technique both schema-based and instance-based concept matching techniques are combined to integrate the ontologies.

### **3.4.1 Hybrid Concept Matching Systems**

The hybrid concept matching technique is considered more efficient and is the most widely used in current research on ontology Integration. Some of the most recent systems related to this technique are given below.

#### **3.4.1.1 Information Flow Theory (IF-Map)**

IF-Map is an automated ontology mapping technique based on channel theory, a semantic information flow theory. The system takes two local ontologies and a third agreed reference ontology. In this technique, first all the instances of local ontologies are assigned to the reference ontology and then IF-MAP takes all three ontologies to find alignment and mapping. If matching gives no results than IF-MAP uses the semantic and structure based methods (Kalfoglou and Schorlemmer, 2003).

#### **3.4.1.2 OWL Light Aligner**

The OWL Light Aligner system takes care of all strings, language, structure, etc. while aligning ontologies. The system considers ontologies as graphs and aggregates concepts on the basis of similarities (Loup, n.d.).

#### **3.4.1.3 Ontology MAP (oMAP)**

oMAP uses weights to discover the best candidates (entities) for mapping between ontologies. A number of matching techniques are used to find correspondence between the entities. These techniques are: a string similarity measure and learning methods which are used for instance data (Straccia and Troncy, 2005). Structural similarity is measured on the basis of information content. It measures the structural similarity between entities of two ontologies especially at granular levels of heterogeneity and gives best candidates for merging ontologies (Hariri, 2006). The electiveness of the machine learning part in oMAP could be improved using some other measures like KLdistance.

#### **3.4.1.4 Risk Minimization based Ontology Mapping (RiMOM)**

RiMOM a system used by integrating multiple techniques like: an edit-distance based strategy, statistical-learning based strategy, and three similarity-propagation based strategies. It starts by performing structural and label similarity, which helps in deciding strategy for the alignment of ontologies (Li et al., 2006). Later, RiMOM integrated some other alignment strategies like a path-similarity based strategy and background knowledge based strategy.

### 3.4.1.5 Semantic Matching (SM)

SM is another algorithm which takes ontologies as two graphs and maps the nodes of graphs on the basis of semantics. This algorithm performs mapping on the basis of semantics which are verified from WordNet (Giunchiglia et al., 2007).

## 3.5 Comparison of Existing Matching Tools

The following table provides a comparison between the above mentioned existing tools and techniques which can be used in ontology integration.

Techniques	Similarity Measuring Techniques	Linguistic Tools	Syntax Matching	Semantic Matching	Taxonomy Matching	Limitations
<b>SKAT (Mitra et al., 1999)</b>	Graph Based Slicing Approach	Online Dictionary Specified by Expert	Y	Y	Y	Requiring of Domain Expert
<b>ONION (Mitra et al., 2000)</b>	Graph Based Technique	Online Dictionary Specified by Expert	Y	Y	Y	Semantics Comparison only if Syntax does not Match
<b>PROMPT (Noy and Musen, 2000)</b>	Edit-distance	WordNet	Y	Y	Y	More Heuristic Rules Required
<b>AnchorPROMPT (Noy and Musen, 2001)</b>	Edit-distance	WordNet	Y	Y	Y	Same Ontology Modelling Techniques Required
<b>CTXMATCH (Bouquet et al., 2003a)</b>	SAT Based Logical Analysis	WordNet	Y	Y	Y	Proposed Structure Dependent

<b>CTXMATCH 2 (Bouquet et al., 2006)</b>	Description Logic Reasoners such as Pellet and FaCT	WordNet	Y	Y	Y	Semantic Similar Structure Required
<b>ASCO (Le et al., 2005)</b>	String Based Technique	WordNet EuroWordNet	Y	Y	Y	RDF(S) Formalism Dependent
<b>TROPES (Euzenat, 1993)</b>	Bridges (Interfaces)	N	Y	N	Y	Trees Structure Used
<b>CAIMAN (Lacher and Groh, 2001)</b>	Probability Measures, String Based Techniques	N	Y	N	Y	Designed for Document Repositories
<b>FCA– MERGE (Stumme and Maedche, 2001)</b>	Linguistic and Context Analysis and Regeneration of Ontology	SMES (Saarbrücken Message Extraction System)	Y	Y	Y	Protégé 2000 Dependent
<b>GLUE (Doan et al., 2003)</b>	Multiple Constraints (Neighbourhood, Name, Frequency, Union, etc.)	N	Y	N	Y	Constraints are used for Ontology Merging
<b>IF-Map (Kalfoglou and Schorlemmer, 2003)</b>	Based on Theory of Information Flow and FCA-MERGE	N	Y	N	Y	Working with same Organisation Ontologies
<b>OWL Light Aligner (Loup, n.d.)</b>	Using Strings, Language, Structure Methods	N	Y	N	Y	OWL Light Ontologies Required

<b>oMAP (Straccia and Troncy, 2005; Hariri, 2006)</b>	String Similarity Measure and Learning Method	N	Y	Y	Y	The Electiveness Of The Machine Learning Part Could Be Improved
<b>RiMOM (Li et al., 2006)</b>	Edit-distance Based Strategy and Statistical- learning Based Strategy	N	Y	N	Y	Time Cost Analysis Required
<b>SM (Giunchiglia et al., 2007)</b>	String Based Technique	WordNet	Y	Y	Y	Not Suitable for Iterative and Interactive System
<b>SIMTO (Farooq, 2009)</b>	String Based Technique, Edit Distance, DoS	WordNet	Y	Y	Y	Does not Consider the Lower Level Concept

Table 3-1: Syntax, Semantic and Taxonomy Comparison of Existing Techniques and their Limitations

### 3.6 Limitations of the State of the Art

The following points are identified as problems for ontology integration from the review of the state of the art in ontology integration, and from the analysis of existing frameworks. These points are also considered in the proposed framework to improve the results of ontology integration.

- The comparison of the concepts should be based on their definitions instead of terms used for the concepts.
- The semantics of the concepts should be domain independent.
- The semantic (contextual) similarity matching should also be based on the semantics of super concepts.
- The Semantic Relation (SR) of concepts should be considered and based on their explicit meanings.
- A layer based approach should be used for the concept matching.

- The next chapter will explain the identified points, motivational examples, propose a framework and further will describe how these points are considered in any prototype system to be developed.

## Chapter 4: Proposed Framework for Semantically Intelligent Ontology Integration

This chapter explains the limitations of the state of the art, provides a motivational example for ontology integration, and introduces the enhanced framework for ontology integration, which is based on providing solutions for the limitations identified in the previous chapter. Within this chapter, section 1 provides the details of the limitations of the literature review. This is followed by section 2, which gives an example explaining the significance of ontology integration. Then section 3 introduces the proposed framework. This chapter concludes by identifying the limitations involved in the development of the framework.

### 4.1 The Foundations of the Proposed Framework

This section gives details about the limitations of the state of the art in ontology integration and provides a foundation for the proposed framework.

#### 4.1.1 The Concept Comparison should be based on Semantics Instead of Terms Used

An ontology contains multiple concepts, and a concept can be represented using different terms or their synonyms in different ontologies of the same domain, such as shown in Figure 4-1. String based concept matching techniques use terms instead of meanings or definitions associated with concepts in order to measure the match between concepts. This technique for concept matching can produce incorrect results in some scenarios. For example; the terms „Comp Sci“ and „Computer Science“ in the following figure using the string based matching technique would not identify both concepts as similar. The similarity in this case focuses on semantics. Therefore, the semantics of concepts should be considered in the process of matching concepts to each other across ontologies.

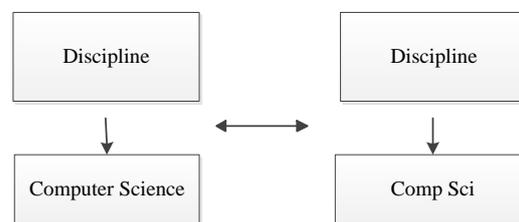


Figure 4-1: Same Concepts with Different Terms

This is a significant limitation of the state of the art and steps towards resolving this would involve calculation of matches between concepts using the meaning associated with the terms of concepts stated in the ontologies. The vocabulary to obtain the meaning of terms and their synonyms should be domain specific. WordNet can be used for concept similarity

identification, as WordNet contains the most widely used general vocabulary. There is however, a limitation that the use of WordNet can cause incorrect results according to previous analysis (Farooq, 2009; Farooq and Shah, 2010). However, this can be overcome through a semi-automated solution which relies on user interaction whilst integrating two ontologies to reduce the possibilities of incorrect results.

#### **4.1.2 The Semantics of the Concepts should be Domain Independent**

The semantic matching of concepts based on linguistics considers the terms used for the ontology concepts, and their synonyms for comparison. This means both concepts can be considered similar, if the term used for one concept is a synonym of the term of the other concept or vice versa. The existing techniques and proposed framework use WordNet (Noy and Musen, 2000; Noy and Musen, 2001; Bouquet et al., 2003a; Le et al., 2005; Bouquet et al., 2006) to get the synonyms of the terms, but WordNet does not provide the abbreviations, acronyms, or composite name words which may refer to the same term. An improved solution would resolve this problem through the addition of a user interaction layer in the integration process. In this process, the user will be asked to provide the synonyms, abbreviations, acronyms, or composite name words and the user will confirm. This will increase the efficiency of the concept matching process.

#### **4.1.3 The Semantic Similarity Matching should also be based on Semantics of Super Concepts**

The super, sub and sibling concepts of a concept in an ontology describe the context of that concept. A concept in an ontology may not have a sub concept or sibling concepts, but it always has a super concept. Therefore, a new concept must have a super concept and may have one or more sibling concepts while being added. This means, the semantics of the super concepts have to be considered in the matching process between concepts. This process would add value to ontology integration, in delivering a solution for enabling intelligent calculation of the context of the concept.

#### **4.1.4 The Semantic Relation (SR) of Concepts should be based on their Explicit Meanings**

Each concept will have relationships with its neighbouring concepts (super, sub, sibling). These relationships should also be taken into account when measuring similarity between concepts. When concepts are compared the relationships between these concepts and their neighbouring concepts, provide a mechanism for assessing the SR between concepts.

Consideration of the SR in ontology integration would help to reduce the contextual heterogeneity within an ontology.

#### **4.1.5 A Layer Based Approach should be used in Concept Matching**

A layered approach for concept matching provides a basis for a hybrid (instance based and schema based) approach to be developed. A layered approach can also help to reduce the complexity of the concept matching problem.

#### **4.1.6 Considerations Towards a Proposed Framework**

Taking into account sections 4.1.2 - 4.1.5 a proposed framework is detailed in section 4.3 below. This proposed framework provides the following solutions to help resolve the above problems.

- Three layered approach to concept matching taking into account syntax, semantics and context.
- Semi-automated interface enabling user confirmation of problematic cases (concept inconsistencies).
- Consideration of the semantics of super concepts to help to define the context of a concept.
- Consideration of the relationships between a concept and surrounding concepts (super, sub, sibling(s)) to help to define the context of a concept.

## **4.2 Motivational Example for Proposed Framework**

This section provides an example explaining the requirements of different concept matching approaches in ontology integration. The following diagram illustrates the related ontologies, A and B. A is an ontology from Organisation A and B is an Ontology from Organisation B. These ontologies involve multiple common concepts (e.g. City, Street, Item and Quantity) which could be considered to be similar.

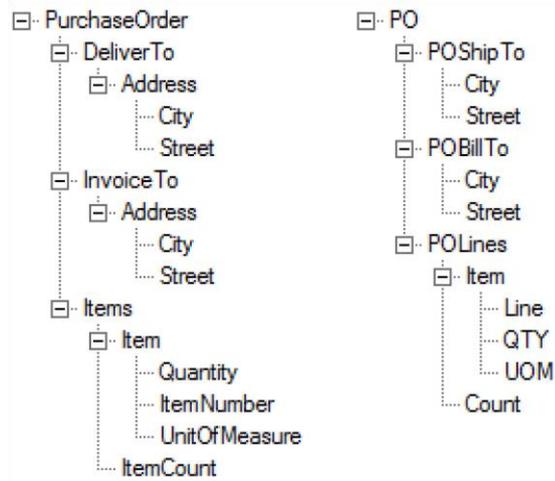


Figure 4-2: Sample Ontologies for Motivational Example for Proposed System

In order to develop an integrated ontology a hybrid approach for concept matching should be considered. For example; the term “City” can be matched in both ontologies (shown in Figure 4-2) by using a string based concept matching approach. The term “City” could be considered to be similar through the use of a string based matching approach. However, the meanings associated with the term “City” should also be considered to increase the efficiency of the matching process. For example; the concepts “DeliverTo” and “POShipTo” are not syntactically similar but they are semantically similar as both terms refers to the “Shipping Address”. Therefore, the meanings behind concepts can play a vital role in the concept matching process.

Similarly, the synonyms and abbreviations of the terms should also be evaluated in concept matching. For example, the terms “Quantity” and “Qty” shown in the above figure can be declared similar if the abbreviations of the term “Quantity” are included as part of the matching process.

On the other hand, the context of a concept should also be taken into account within the matching process. The meaning of concepts considered to be similar (in a syntactical matching process) may be different, if the concepts have a different hierarchy and different contexts in their parent ontologies. For example; the “City” concept appears twice in each of the above mentioned ontologies, but it has different hierarchy and meaning. In the same way, SR is another aspect of concept matching in ontology integration. The concept “Address” will

be ignored in ontology integration if the SR is not taken into account within the matching process.

### 4.3 An Enhanced Framework for Ontology Integration

In the previous chapter, different ontology matching approaches and tools have been reviewed to examine the concept matching strategies used for ontology integration. This section explains the proposed framework which will deliver an enhanced approach to ontology integration.

#### 4.3.1 A Block Diagram of Proposed Framework

The Figure 4-3 illustrates a block diagram which explains the flow of the proposed framework. The proposed framework takes two ontologies (A, B) named as “Reference Ontologies” (input ontologies) and converts them into a tree structure using the schema of the ontologies. Each reference ontology tree contains all the concepts of the reference ontology. Each concept in the tree includes the term used for the concept, the meaning of the term and the relationship with other neighbouring concepts (super-concept, sub-concept and sibling-concepts).

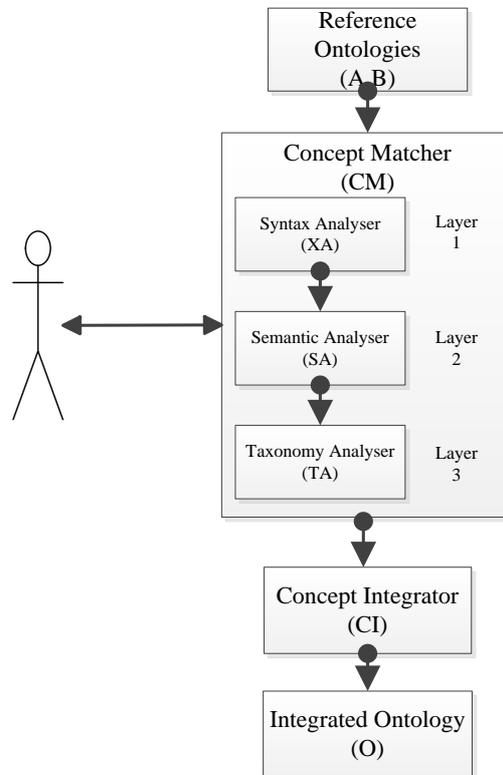


Figure 4-3: Block Diagram of Proposed Framework

After converting the reference ontologies into a tree format, both trees are forwarded to the Concept Matcher (CM) for the identification of matching concepts. The CM provides a layer based approach to measuring the similarity between concepts across the reference ontologies. The CM compare the concepts into three different levels (Syntax, Semantic and taxonomy explained in Section 4.3.2) to identify the similarity between them. The CM may involve the user in a process of confirming similarity or not, in complex cases. Complex cases may include concepts: that are not identified similar semantically; that have the same term but a different definition; or do not have any matching concepts.

The CM transfers concepts which have been identified as matching to the Concept Integrator (CI). The CI integrates the matching concept in order to build the integrated ontology. The processes of identification of matching concepts and the integration of these will continue until all the concepts from both reference ontologies are merged into the integrated ontology.

Finally, the CI provides an integrated ontology as output. The integrated ontology will contain all the matching concepts of the reference ontologies identified by CM or verified from the user.

### 4.3.2 Layer Based Approach of Concept Matching

The CM performs the task of the identification of concept matching through three layer of activity. Each layer executes a specific task. The layers are named Syntax Analyser (XA), Semantic Analyser (SA) and Taxonomy Analyser (TA). The functionality of XA, SA and TA is explained below.

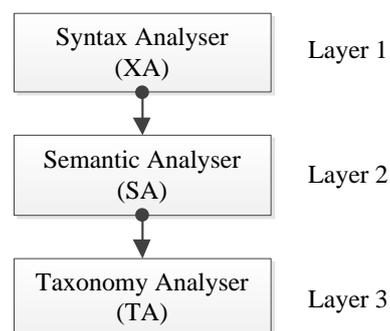


Figure 4-4: Layers of Concept Matching Approach

The XA, SA and TA use the terms, definitions of the terms, synonyms of the terms and definitions of the synonyms in the process of matching. The definitions of the terms and their synonyms are collected from WordNet. The first section of the next chapter explains why the

proposed framework is built using the WordNet resource, by providing a comparison of WordNet with other general purpose ontologies.

#### 4.3.2.1 Syntax Analyser (XA)

The XA compares the concepts for matching by comparing the strings of the terms or by finding a first term from the list of synonyms of the second term, if the strings are not matched (same). For example; the XA declares the concept Male similar from two ontologies by comparing the terms used for the concept in two ways:

- The same term Male is used in the given ontologies for the Male concept.
- The term Male is used in the first ontology and the term Boy is used in the second ontology for the Male concept. In this scenario, XA finds the term boy in the list of synonyms of the term Male.

The XA marks the concepts as similar or not-similar syntactically and passes these to the SA. The process of XA can also be explained as:

$T1 = T2$   
where  
 $T1 \in C1, T2 \in C2, O1 \neq O2, \text{SynSet}(\text{String}(T1)) \neq \emptyset, \text{SynSet}(\text{String}(T2)) \neq \emptyset$   
if  
 $(\text{String}(T1) = \text{String}(T2))$  or  $(\text{String}(T1) \in \text{SynSet}(\text{String}(T2)))$  or  $(\text{String}(T2) \in \text{SynSet}(\text{String}(T1)))$

which means that the term T1 of concept C1 from ontology O1 will be equal to the term T2 of concept C2 from ontology O2, if the string of T1 is equal to the string of T2, or the string of T1 is part of the list of synonyms of the string of T2 (wherein the list of synonyms T2 should not be empty), or the string of T2 is part of the list of synonyms of the string of T1 (wherein the list of synonyms T1 should not be empty).

#### 4.3.2.2 Semantic Analyser (SA)

The SA takes marked concepts from the XA whether they are similar or not-similar. In other words, the SA is dependent on the XA as SA takes process data from the XA for next level processing. The SA uses the definitions of the terms of the concepts and their synonyms to enhance the concept matching process. The understanding of the meaning of the terms in the particular domain is very important in the concept matching process. The SA first compares the definitions of both terms (syntactically similar or not-similar). Both concepts will be declared similar semantically if the SA has found the meaning (definition) of the terms to be

similar. The next step of SA executes if the terms meanings are different. In this case, the SA takes the definitions of the synonyms for the first term and compares with the definition of the second term. For example, the terms “Clerk” and “Steno” are not syntactically similar. However, the SA would find both terms to be similar because the meaning is equivalent. The XA would mark the concepts as similar or not-similar semantically and would pass them to the TA.

The functionality of the SA can be described mathematically as:

$T1 = T2$

where

$T1 \in C1, T2 \in C2, O1 \neq O2, \text{SynSet}(\text{String}(T1)) \neq \emptyset, \text{SynSet}(\text{String}(T2)) \neq \emptyset, \text{Def}(\text{String}(T1)) \neq \emptyset, \text{Def}(\text{String}(T2)) \neq \emptyset$

if

$\{(\text{String}(T1) = \text{String}(T2)) \text{ or } (\text{String}(T1) \in \text{SynSet}(\text{String}(T2)) \text{ or } (\text{String}(T2) \in \text{SynSet}(\text{String}(T1)))\} \text{ and } \{(\text{Def}(\text{String}(T1)) = \text{Def}(\text{String}(T2)))\}$

which means that two concepts will be declared similar semantically if the string of T1 is equal to the string of T2, or the string of T1 is part of the list of synonyms of the string of T2, or the string of T2 is part of the list of synonyms of the string of T1. However, this involves a compulsory condition ( $\text{Def}(\text{String}(T1)) = \text{Def}(\text{String}(T2))$ ) which verifies the similarity of the definition associated with each term.

The SA confirms that the term T1 of concept C1 belongs to ontology O1, the term T2 of concept C2 belongs to ontology O2, the list of synonyms T1 should not be empty, the list of synonyms T2 should not be empty, the term T1 should have a definition, the term T2 should have a definition, and both ontologies are not equal, before starting the process of concept matching.

#### **4.3.2.3 Taxonomy Analyser (TA)**

The TA is dependent on the SA in the concept matching process. The TA takes the semantically similar concepts and compares them based on the hierarchical information associated with them. The TA gathers all the super terms up to the root term of the terms being compared. The TA also compares the super terms and their definitions to confirm the hierarchy of the matching concepts before passing the information regarding the matching concepts to the CI. For example; the term Male is a sub-concept of the Person term in the first ontology and the term Boy is a sub-concept of the Person term in the second ontology. The terms Male and Boy are semantically similar and both terms have a semantically equal super

concept. Therefore, the integrated term should be placed under the term Person in an integrated ontology.

#### 4.3.2.4 The Super, Sub and Sibling Concept Identification for TA and CI

The following equations relate to the identification of super concepts. The equations explain the process of how the TA identifies super concepts for semantic comparison. The TA collects all super concepts by using the structural information of the concept and then compares them based on their semantics. The TA declares the match as not similar if it finds any semantic dissimilarity between concepts across ontologies. The TA uses a Top-Down Approach (root to nodes in tree structure) for the checking of structural similarity.

##### Siblings Concept Identification of T1 and T2

where

$T1 \in C1, T2 \in C2, O1 \neq O2$

if

$(Dis(T1) = Dis(T2))$  and  $(Super(T1) = Super(T2))$

---

##### Super Concept Identification of T1 and T2

where

$T1 \in C1, T2 \in C2, X1 \in C1, X2 \in C2, O1 \neq O2, X = \text{Super Concept to Map}$

if

$X1 = T1 \rightarrow Inherit(Super(X1))$

$X2 = T2 \rightarrow Inherit(Super(X2))$

$X = (X1 = X2)$

---

##### Sub Concept Identification of T1 and T2

where

$T1 \in C1, T2 \in C2, X1 \in C1, X2 \in C2, O1 \neq O2, X = \text{Sub Concept to Map}$

if

$X1 = X1 \rightarrow Inherit(Super(T1))$

$X2 = X2 \rightarrow Inherit(Super(T2))$

$X = (X1 = X2)$

The above equations are also used by the CI in order to map similar concepts of the reference ontologies into an integrated ontology. The CI identifies the super, sub and sibling concepts and maps them into the integrated ontology if they are similar. In this process, the CI checks the distance between two terms and then compares the super concepts of those two terms. Both terms will be siblings, if their distance is equal and their super concepts will be the same. On the other hand, the first concept will be a sub-concept of the second concept if the term of the first concept will be inherited from the term of the second concept. Similarly, the first concept will be the super-concept of the second concept if the term of the second concept will be inherited from the term of the first concept.

The next chapter will provide reasoning for the use of WordNet as a global vocabulary for the definitions of the terms, synonyms of the terms, and definitions of the synonyms. It will also explain the algorithms involved in the process of this proposed ontology integration framework and the limitations of the proposed framework implementation.

## **Chapter 5: Implementation**

This chapter provides an overview of the implementation processes followed in order to construct the ontology integration framework described in Chapter 4. The chapter starts with an examination of appropriate vocabulary platforms on which to base programmatic reasoning. This is followed by presentation of the algorithms and pseudo code used to construct different modules of the proposed framework. Finally the chapter ends with a discussion of the limitations of the implementation of the proposed framework.

### **5.1 The Comparison of WordNet with other Vocabulary Resources**

This section gives a brief overview of the most popular vocabulary resources and their comparison with WordNet. It also explains the importance of and reasons for using WordNet as the global vocabulary platform for the proposed framework.

#### **5.1.1 Cyc**

Cyc has been developed as a research project since 1984 (Curtis et al., 2006; Cycorp, n.d). Cyc is a large scale general purpose knowledge based system (Seco, 2005) which uses CycL. CycL is a knowledge representation schema which represents higher level relationships between concepts through the use of predicate calculus (Curtis et al., 2006). The inference engine of Cyc combines general theorem which provides a formal representation of facts, rules and heuristics of objects and events of daily life. The defined facts, rules and heuristics provide reasoning about the objects and events. Cyc is not only developed as an extensive taxonomy of concepts and terms, but also has a large, self-reflective vocabulary for describing the most common definitional needs (Reed and Lenat, 2002). However, the glossaries (definitions) of Cyc are long and rambling (Seco, 2005).

#### **5.1.2 WordNet**

WordNet is a semantic lexicon for the English language developed at the Cognitive

Science Laboratory of Princeton University (Suchanek et al., 2007; Princeton.Edu, n.d.). WordNet concepts have different relations (i.e. is-a) with each other. WordNet is planned to model the human glossary and psycholinguistic findings have also been taken into account in its design phase (Suchanek et al., 2007). WordNet keeps track of the context of situations in which words are being used, which provides help in defining semantically similar words as synonyms. WordNet also provides the taxonomic relations between words (i.e. Super, sub and

sibling relationship of words). WordNet is the most widely used lexicon by the community of language processing (Seco, 2005).

### **5.1.3 HowNet**

HowNet is a common-sense knowledge base which gives the inter-conceptual relationships and inter-attribute relationships of concepts. HowNet relationships are provided in lexicons of Chinese and equivalent English (Dong et al., 2010). HowNet does not provide a human-oriented textual glossary for every concept. However, it combines sememes (a basic representation of meaning) for the composition of semantic representation of every concept which does not provide better taxonomic results (Veale, 2005). HowNet is unique in its four peculiarities: (1) use of sememes, (2) definition in a structuralized language, (3) Self-sufficiency and (4) Language independence (Dong et al., 2010).

### **5.1.4 Preference of WordNet for the Proposed Framework**

The above explanation provides a brief overview of resources which can be used as vocabulary for language processing. The discussed resources are the most popular resources for language processing (Seco, 2005; Styltsvig, 2006). Each resource has its own advantages and drawbacks. All the resources provide information about the semantics of individual concepts, however, the choice of resource will often vary dependent on the specifics of the requirements of individual research projects.

The proposed framework is largely based on semantic comparison of concepts. It also uses the synonyms of the terms used for the concepts. In a comparison of Cyc, WordNet and HowNet, WordNet is the most substantial lexicon to provide synonyms based on the semantics of concepts. The proposed framework also takes into account glossaries of the concept terms and glossaries of the synonyms of the concept terms. In this scenario, WordNet is preferred as compared to Cyc and HowNet, because it provides glossaries that are closer to those of human perception.

The better approach for the evaluation of taxonomic relations is another important factor that makes WordNet different from Cyc or HowNet. WordNet is preferred as the core vocabulary for development of the proposed framework, as it is extensively used in language processing and fulfils all the requirements of the proposed framework effectively.

## 5.2 Implementation

This section gives a brief description of the proposed framework and a detailed overview of the implementation of the proposed framework. A prototype is developed to validate the defined objectives of this thesis and to check the efficiency of the algorithms. The algorithms used in different modules of the proposed framework are also included in this section.

### 5.2.1 Brief Overview of the Proposed Framework

The following Figure 5-1 provides the structure of the proposed framework, the framework itself is described in full in the previous section 4.3. The Figure 5-1 explains that the proposed framework takes two reference ontologies A and B and passes them to the XA. The XA parses the two ontologies and compares the concepts syntactically. The SA obtains the concepts from the XA which are to be compared, and evaluates them semantically. The SA communicates with the user for confirmation of similarity matches in problematic cases. The TA provides the structural comparison of the concepts that are transferred from the SA, if the concepts are semantically similar. Finally, the CI integrates all the similar concepts and produces the integrated ontology.

Finally, the CI integrates all the similar concepts and produces the integrated ontology.

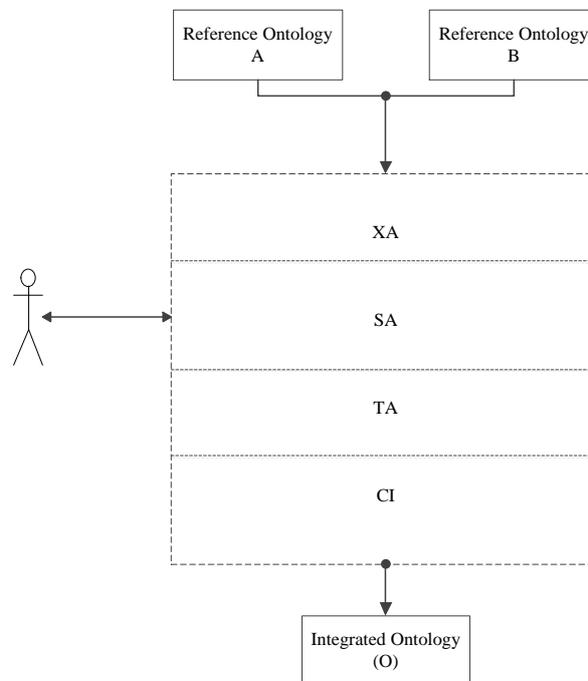


Figure 5-1: Structural Diagram of Proposed Framework

## 5.2.2 Implementation Tools and Languages

The proposed framework was implemented in C#.Net with the help of Visual Studio 2010 – Microsoft’s visual programming environment. Schemas for reference ontologies were used in order to parse the ontologies. After parsing, concepts were loaded into a tree structure where each node represents a concept. Each node of the tree contains the term, definition of the term and the taxonomic (super node, sub node and sibling nodes) information of the node. The prototype also provides a Graphic User Interface (GUI) that can be used to build ontologies using WordNet. The GUI for the ontology building used the WordNet web service (Princeton.Edu, n.d.) to obtain definitions of the terms of concepts. The user can also write the definition if WordNet does not have the definition of any term. The prototype also provides a GUI to the user for confirmation of concept matches in problematic cases. The implemented GUIs for the processes mentioned above are shown in the Figures 5-2, 5-3 and 5-4. In the Figure 5-2, First Ontology and Second Ontology are Reference Ontologies and Integrated Ontology is an Output Ontology. The Integrated Ontologies integrates the similar concepts in the resulted ontology. For example; the concepts “DeliverTo” in the First Ontology and “POShipTo” in the Second Ontology are not syntactically similar but they are semantically similar as both terms refers to the “Shipping Address”. As a result, both concepts are integrated as shown in Integrated ontology.

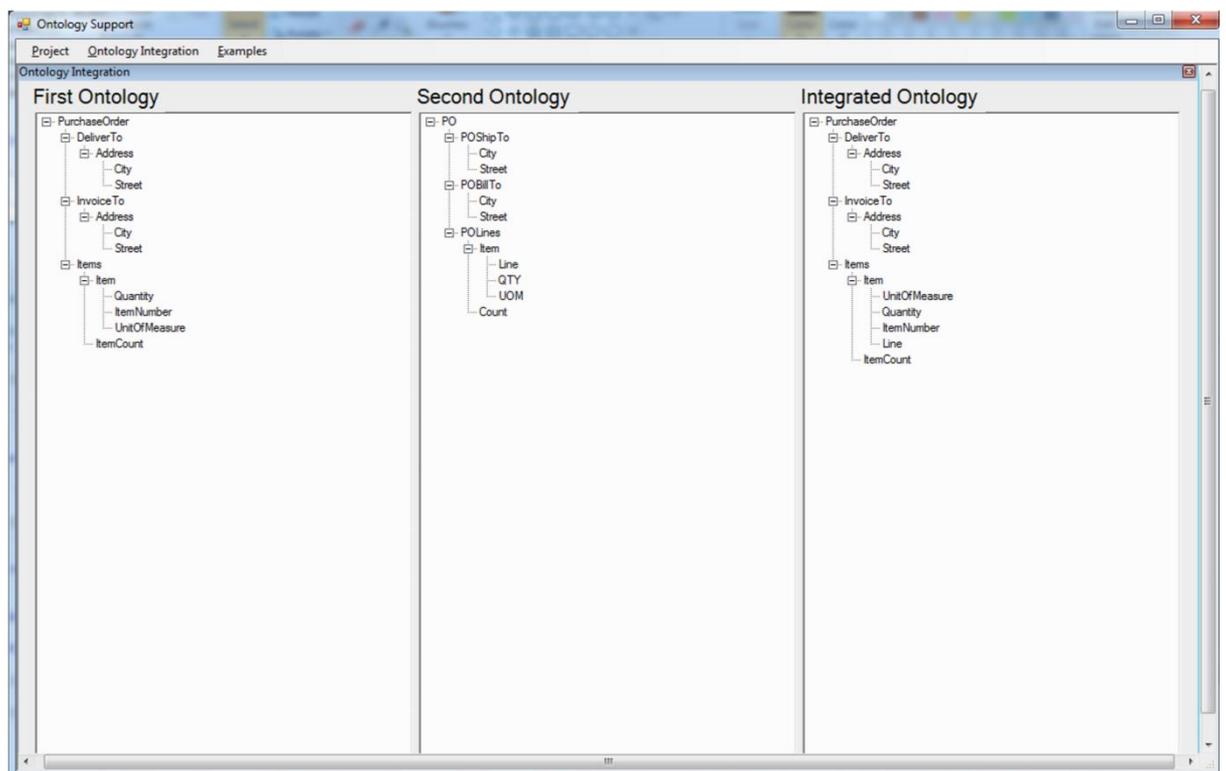


Figure 5-2: GUI of Final Output of Reference Ontologies and Integrated Ontology

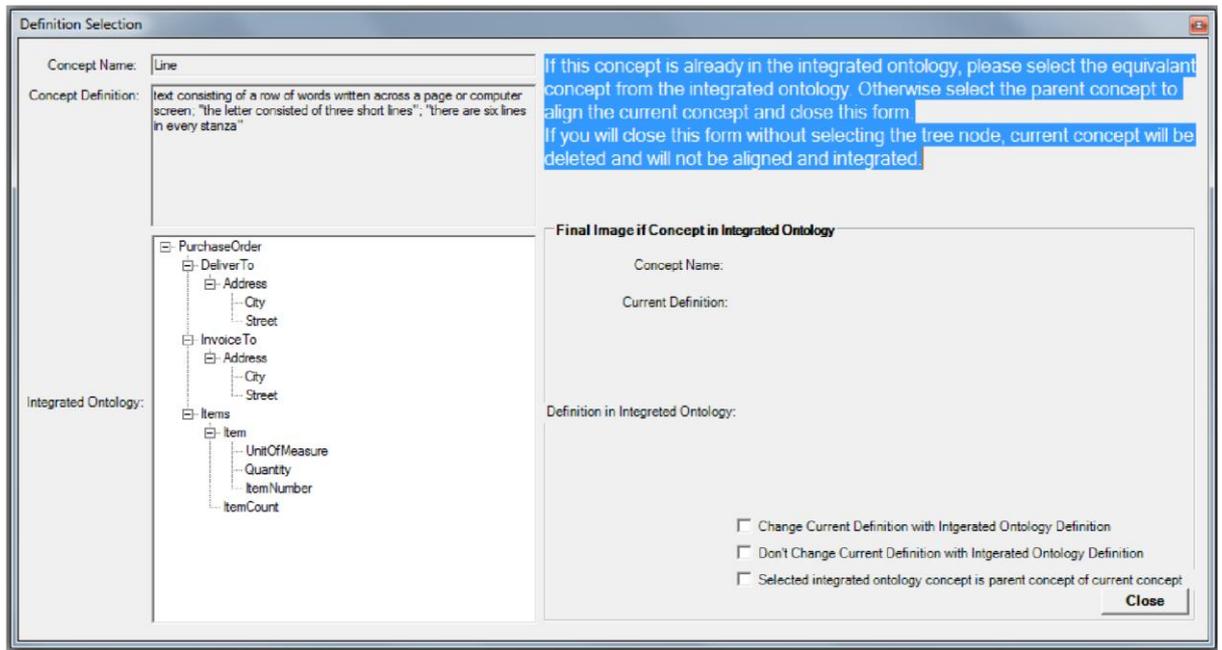


Figure 5-3: GUI for User Interaction

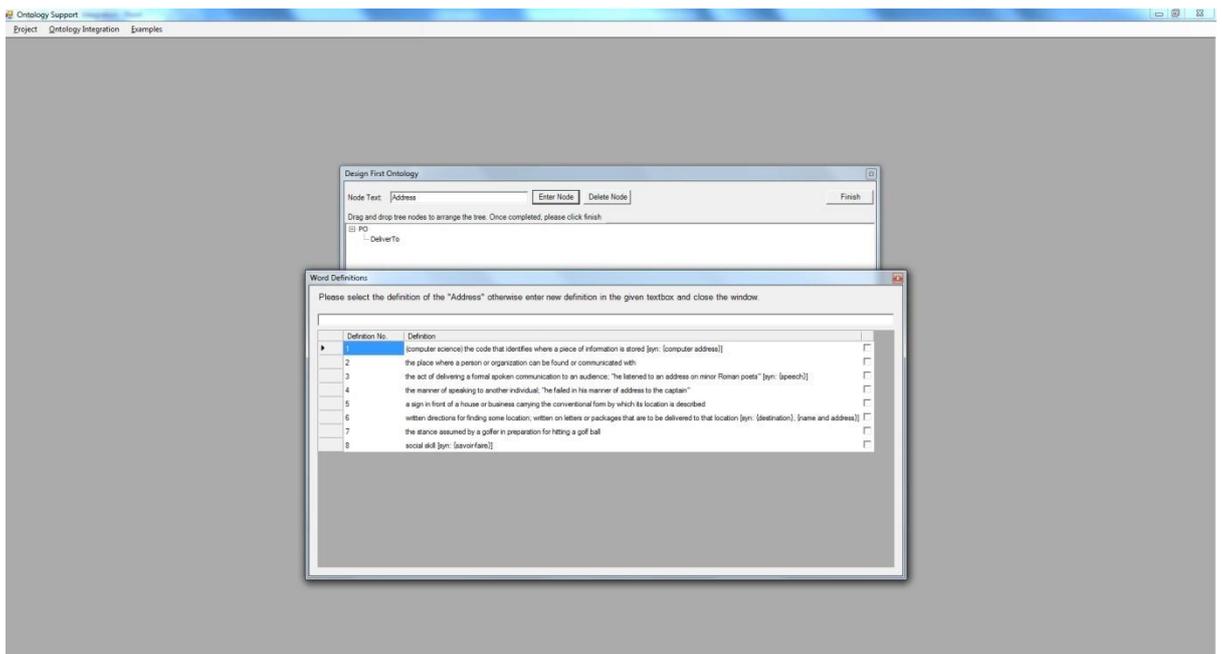


Figure 5-4: GUI of Ontology Building

### 5.2.3 Pseudo Codes and Algorithms

The pseudo codes of different modules and algorithms used for the implementation of the proposed framework are as follows:

### 5.2.3.1 Pseudo Code of Main Module

The main module takes two reference ontologies, converts them into a tree structure, extracts all the nodes (concepts), and starts comparison based on syntactic, semantic and structural properties.

#### Pseudo Code:

```
void mainModule (void) {  
  
Tree ontologyA = ontologyLoadModule (PathOfTheOntologyASchema);  
  
Tree ontologyB = ontologyLoadModule (PathOfTheOntologyBSchema);  
  
Tree ontologyO = new Tree();  
  
List <TreeNode> OntologyANodes = loadOntologyAIntoTreeA (ontologyA);  
  
List <TreeNode> OntologyBNodes = loadOntologyBIntoTreeB (ontologyB);  
  
List <Concept> matchingConcepts = conceptMatcher (ontologyA, ontologyB);  
  
.  
..
```

### 5.2.3.2 Pseudo Code of Ontology Load Module

The main module uses this module in order to load the reference ontologies.

#### Pseudo Code:

```
XMLFile ontologyLoadModule (String PathOfTheOntologySchema) {  
  
    XMLFile ontology = null; ontology = loadOntology (PathOfTheOntologySchema);  
    return ontology;  
  
}
```

### 5.2.3.3 Pseudo Code of Tree Load Module

This module is used in the main module for the conversion of the loaded reference ontologies into a tree structure.

#### Pseudo Code:

```
Tree loadOntologyIntoTree (XMLFile file) {
```

```

Tree tree = extractConcepts (file);

return tree;

}

```

#### 5.2.3.4 Pseudo Code of CM Module

The CM obtains the concepts which are to be compared from the main module. It contains three sub-modules for the comparison of the concepts: XA, SA and TA. The algorithms for these modules are given below.

#### Pseudo Code:

```

bool conceptMatcher (Concept ontologyAConcept, Concept ontologyBConcept) {

    bool result = false;

    bool syntacticallyMatchConcepts = false;

    bool semanticallyMatchConcepts = false;

    bool taxonomicallyMatchConcepts = false;

    //Algorithm of the Syntax Analyser is given in the section

    syntacticallyMatchConcepts = syntaxAnalyser (ontologyAConcept,
ontologyBConcept);

    //Algorithm of the Semantic Analyser is given in the section

    semanticallyMatchConcepts = semanticAnalyser (ontologyAConcept,
ontologyBConcept);

    if (syntacticallyMatchConcepts and semanticallyMatchConcepts)

    {

        //Algorithm of the Taxonomy Analyser is given in the section

        taxonomicallyMatchConcepts = taxonomicAnalyser (ontologyAConcept,
ontologyBConcept);

    }

    if (syntacticallyMatchConcepts and semanticallyMatchConcepts and
taxonomicallyMatchConcepts)

    {

```

```

        result = true;
    }

Return result;
}

```

### 5.2.3.5 Pseudo Code of CI Module

The CI integrates similar concepts into an integrated ontology.

#### Pseudo Code:

```

Tree integratedOntology conceptIntegrator (Concept ontologyAMatchingConcept, Concept
ontologyBMatchingConcept) {
    if (conceptMatcher) {
        ontologyO = ontologyAMatchingConcept ;
    }
}

```

### 5.2.3.6 Algorithm for the XA

This section contains the algorithm used by the XA for the syntactic comparison between concepts.

#### Algorithm:

##### Input:

- Term A from Ontology O1
- Term B from Ontology O2

##### Output:

True/False

##### Begin

Temp = False

SynSetB List = getSynSet(Term B)

SynSetDefB List = getSynSetDef(Term B)

If String(Term A) equals to String(Term B) or SynSetB contains String(Term A)

Then Temp = True

End If

Return Temp

**End Sub**

**Function** getSynSet (String term) Returns ListOfStrings

ListOfStrings = WordNetWebService.getSynonyms(term)

Return ListOfStrings

**End Function**

**Function** getSynSetDef (String term) Returns ListOfStrings

ListOfStrings = WordNetWebService.getDefinitions(term)

**Return ListOfStrings**

**End Function**

**End Main**

#### **5.2.3.7 Algorithm for the SA**

The following is an algorithm used by the SA for the semantic comparison between concepts.

**Algorithm:**

**Input:**

- Syntax analyser will pass the list of Term A, Term B, SynSetDef(Term B)

**Output:**

True/False

**Begin**

Temp = False

If String(Def(Term A)) equals to String(Def(Term B))

Then Temp = True

```
End If  
Else If SynSetDef(B) contains String(DefTerm A))  
Then Temp = True  
End Else If  
Return Temp
```

**End Sub**

**End Main**

### 5.2.3.8 Algorithm for the TA

The TA uses the following algorithm for the structural comparison of concepts.

**Algorithm:**

**Input:**

- The output of the semantic analyser should be True

**Output:**

True/False

**Begin**

Temp = True

ListOfSuperTermsOfTermA = C#.NetTreeFunction(Root, Term A)

ListOfSuperTermsOfTermB = C#.NetTreeFunction(Root, Term B)

For each term in ListOfSuperTermsOfTermA

If (Def(ListOfSuperTermsOfTermA(Index) not equals to

Def(ListOfSuperTermsOfTermA(Index)) and

ListOfDef(Synset(ListOfSuperTermsOfTermA(Index)) do not contains

Def(ListOfSuperTermsOfTermA(Index)

Temp = False

Break For

End If

End For

**Return** Temp

**End Sub**

**End Main**

### **5.3 Implementation Limitations**

In the implementation of the proposed framework, WordNet is used as a general purpose vocabulary which is a limitation of the proposed framework. WordNet does not contain the definition of the terms in all possible domains. Therefore, WordNet may not be able to provide the definition of a word in a particular domain. WordNet also does not deal with terms based on multiple words. It provides the definitions of general purpose terms (based on single words only). Similarly, WordNet does not support the abbreviation of the terms as their synonyms which can produce incorrect results in the process of ontology matching. Although, the semi-automatic nature of the proposed framework considers this problem and obtains a definition from the user in these cases. However, the addition of domain specific vocabularies (in conjunction with WordNet) could be used to increase the efficiency of the proposed framework.

The implemented framework obtains only two ontologies for the comparison. The input of more than two ontologies can be an extension of the proposed framework. In the same way, the proposed framework accepts the input ontologies of the same domain. The consideration of multiple domain ontologies as reference ontologies can enhance the proposed framework in order to apply it into multiple domain case studies.

## **Chapter 6: Experiments and Results of the Implemented Proposed Framework**

This section explains the experimental test bed established to evaluate the proposed framework and its prototype implementation. Two case scenarios are presented to be integrated within the context of the experimental test bed with the results of this process then analysed and discussed.

### **6.1 Methodology**

An enhanced ontology integration is proposed during the course of this chapter after reviewing and analysing the state of the art methodologies for ontology integration and existing applications for the same purpose. The methodology of the comparative study is followed in the current thesis in order to evaluate the effectiveness and the efficiency of the proposed framework. The proposed framework is implemented to evaluate it. The implementation process along with the algorithms used are explained in detail in Chapter 5. The evaluation process of the proposed ontology integration framework is based on the Concept Matching (CM) process. The effectiveness and the efficiency of the framework are also based on the concept matching. The functionality of the CI is also dependent on the CM. In other words, the CM is at the heart of the proposed framework.

In order to evaluate the performance of the proposed framework, the implemented concept matching techniques are compared with SIMTO (a technique of concept matching identification) as proposed by Farooq in 2010 and extended in Farooq and Shah, 2010. During the evaluation phase of SIMTO and post evaluation SIMTO has been compared with other concept matching solutions. SIMTO compares concepts based on their roles as defined in the ontology. SIMTO does not support granular level concept matching.

The following Figures 6-1 and 6-2 present the tree structures of the pair of ontologies which are reproduced for the testing of the proposed system. Two ontologies were used in the evaluation of SIMTO (CSUET and LCWU) in the SIMTO test case scenario 2 and two ontologies of software development organisations were used in SIMTO test case scenario 1. The same two pairs of ontologies are considered within the scope of evaluation of the proposed system.

The reasons behind the choice of SIMTO for the evaluation process are that the SIMTO is an established research where a dual ontology integration approach is used similar to that of proposed. During the evaluation phase SIMTO was evaluated through its comparison with nine other established researches. SIMTO was evaluated using the same quantitative matrices as suggested by proposed evaluation. Therefore, the use of SIMTO as reference research will help and still does not sacrifice the quality of evaluation process.

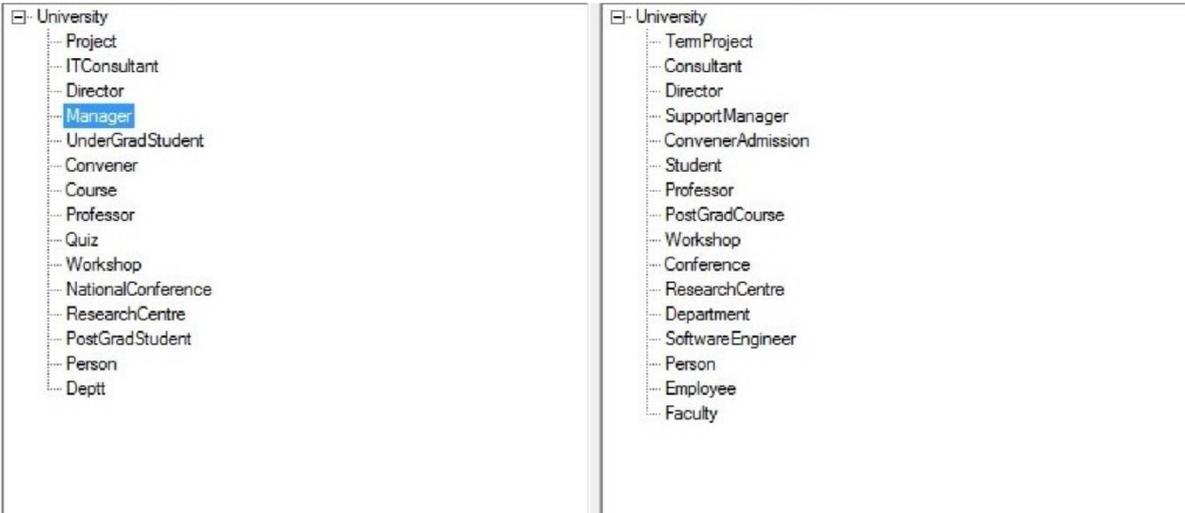


Figure 6-1: Sample Reference Ontologies for Test Case Scenario 1



Figure 6-2: Sample Reference Ontologies for Test Case Scenario 2

The root of each tree presents the domain of the ontology and each node presents a concept involved in the ontology.

## **6.2 Experiments**

Based up on the methodology for the evaluation of the proposed framework defined earlier, an experimental prototype is designed and implemented as explained in chapter 5. This experimental prototype is tested and evaluated using two test cases as explained earlier. These test cases are evaluated based upon identification of the syntactically and semantically similar concept in the reference ontologies. This identification of the similar concepts also takes care of the respective hierarchical structure of the identified concept in ontologies.

The correctness and completeness are two quantitative values used to gauge the efficiency and effectiveness of the proposed system. Same formulas are used for both the test scenario even by Farooq. This gives a good understanding of the effectiveness of the system, as a range of values for correctness and completeness of the experimental prototype are available for both scenarios (Proposed, SIMTO and 9 others). The details are presented in sections for respective tests.

$$\text{CORRECTNESS} = \text{CORRECT PAIRS FOUND} / \text{CORRECT PAIRS EXPECTED}$$

$$\text{COMPLETENESS} = \text{CORRECT PAIRS FOUND} / \text{TOTAL PAIRS FOUND}$$

### **6.2.1 Test Case Scenario 1 (as defined in Farooq, 2010)**

Figure 6-1 describes test case scenario 1 where two reference ontologies describing University organisational structures are presented. Figure 6-3 depicts the SIMTO results in a tree structure as described in Table 4.6.

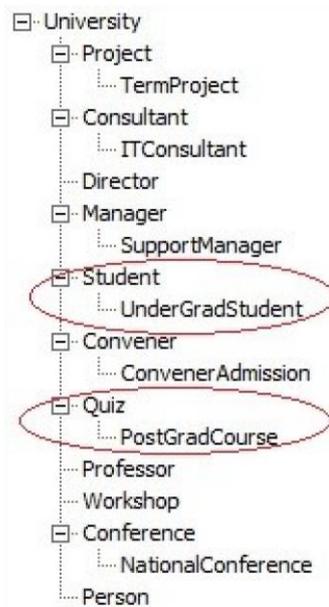


Figure 6-3: A Tree Structure of the SIMTO Results (Re-interpretation of Table 4.6, P65 - Farooq, 2009)

### 6.2.1.1 Test Case Scenario 1 Analysis and Observations

The results produced from SIMTO shown in Figure 6-3 are analysed, observed and discussed in this section. The following are the observed cases which were not considered in SIMTO.

SIMTO had not considered the two concepts similar if the term of the first concept is a synonym of the term used for the second concept. For example, the terms Deptt and Department of the reference ontologies (as shown in Figure 6-1) are semantically similar; however, earlier no similarity is drawn between these concepts.

SIMTO had not found the semantic relationship between the concepts of Student and PostGradStudent (encircled in Figure 6-1 and 6-3). The concept PostGradStudent should come under the concept of Student as a sibling of UnderGradStudent. In other words, the Student concept should be a super-concept of PostGradStudent.

SIMTO had not considered the semantics of the concepts Quiz and PostGradCourse (encircled in Figure 6-3). SIMTO had merged the concept PostGradCourse as a sub-concept of the concept Quiz in the integration process. However, the concept PostGradCourse should come under the concept of Course as a sub-concept based on semantic comparison.

The above mentioned issues are taken into account in the proposed framework of ontology integration. The Figure 6-4 produced from the implemented prototype, shows the proposed

framework considered, and rectifies the issues mentioned earlier. The corrections are encircled as shown in Figure 6-4.

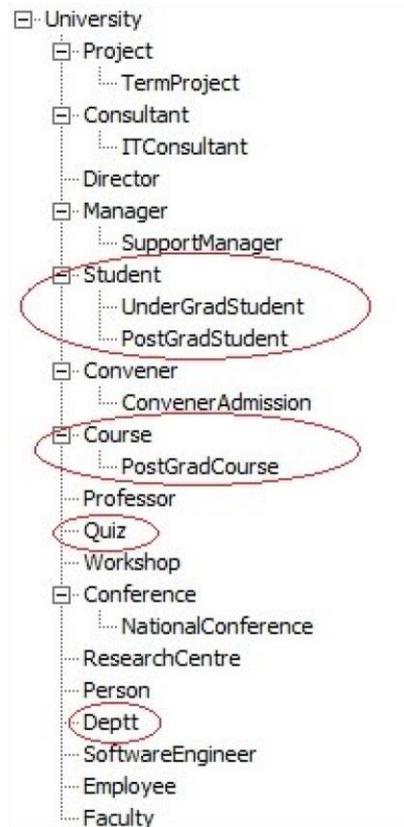


Figure 6-4: Integrated Ontology for Test Case Scenario 1, Generated by using Proposed Framework

Similarly, Figure 6-5 gives a comparison between the pairs of similar concepts returned and the correct pairs found. The blue and red bars in the graph represents the pairs of similar concepts returned and correct pairs found respectively. In this graph, the values of the proposed framework are calculated using the formulae given in the equation of 4.1, 4.2 and 4.3 in (Farooq, 2009). Whereas the values of SIMTO and other existing techniques are taken from the Table 4.9 (Farooq, 2009).

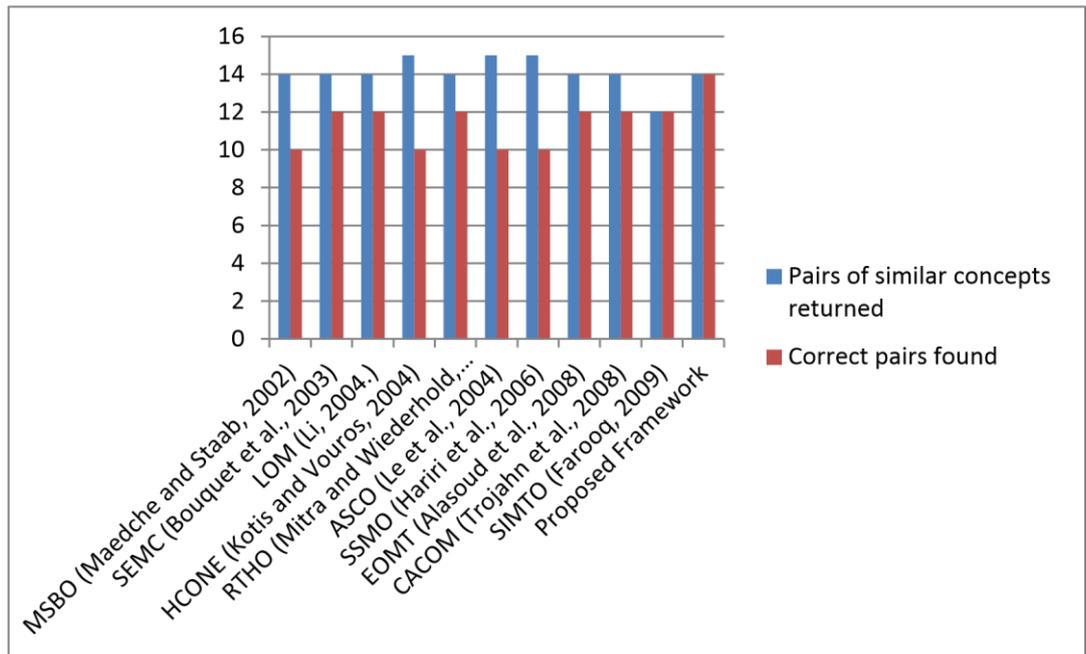


Figure 6-5: Performance Comparison of Test Case Scenario 1 between Proposed Framework and Existing Techniques

The expected returned pair of concepts of SIMTO and the proposed framework was 12 and 14 respectively. Although, SIMTO returned pairs of similar concepts and the correct pairs found are equal as the proposed framework. However, the expected return pair of concepts was different in SIMTO and the proposed framework.

### 6.2.2 Test Case Scenario 2 (as defined in Farooq, 2010)

Figure 6-2 illustrates the two reference ontologies of two software development organisations as test case scenario 2. The results of SIMTO are represented in the Table 6-1.

<b>Match Cases</b>	<b>Ontology A Concept</b>	<b>Ontology B Concept</b>	<b>Similar Roles</b>
1	SoftwareEngineer	Programmer	DesignAlgorithms, DesignStructure, ImplementDatabase, ImplementGUI
2	Programmer	Programmer	DesignWebPages, ImplementDatabase, ImplementAlgorithm, ImplementReports, ImplementGUI, ImplementStructure
3	Programmer	Coder	DesignWebPages, ImplementDatabase, ImplementAlgorithm, ImplementReports, ImplementGUI, ImplementStructure
4	Designer	Designer	DesignGraphics, DesignWebPages
5	Designer	SoftwareArchitect	DesignReports, DesignInputScreens, DesignGraphics, DesignWebPages
6	Analyst	Analyst	AnalyseSoftwareRequirements, AnalyseHardwareRequirements, AnalyseFunctionalRequirements
7	SQAEngineer	SQAEngineer	TestFunctionalRequirements, TestNonFunctionalRequirements
8	DBA	SoftwareEngineer	DesignDatabase

9	DBA	Programmer	DesignDatabase, ImplementDatabase
10	DBA	Coder	ImplementDatabase
11	DBA	DBA	ImplementDatabase, TuneDatabase, BackupDatabase
12	TechnicalWriter	TechnicalWriter	WriteRequirementsSpecification, WriteDesignDocuments, WriteCodeDocument
13	ProjectManager	ProjectManager	CostManagement, ResourceManagement
14	ProcessManager	ProjectManager	DefineStandardOperatingProcedures

**Table 6-1: A Tabular Form of the SIMTO Results (Re-interpretation of Table 4.3, P61 - Farooq, 2009)**

The above table represents the pairs of similar concepts identified by SIMTO based on their role.

### 6.2.2.1 Test Case Scenario 2 Analysis and Observations

In the analysis of the SIMTO's results of test case scenario 2 shown in Table 6-1, some problems are identified and listed. SIMTO had not considered the listed problems during the concept matching process.

SIMTO had declared the matches 1, 3, 5, 8, 9 and 10 similar as shown in Table 6-1. It is observed that all these cases are not semantically similar. In order to integrate the ontologies, the mentioned results will produce incorrect results in terms of hierarchy.

SIMTO had considered the similarity between concepts based on their roles as mentioned earlier. During the comparison of child concepts, SIMTO had not considered their super concepts. For example, SIMTO had identified the concepts DBA and Programmer based on their sub-concepts: DesignDatabase and ImplementDatabase, but DBA and Programmer both are not semantically same.

The proposed framework of ontology integration has considered the mentioned problems. Figure 6-6 shows the integrated ontology built from the implemented prototype. The proposed framework takes care of syntax, semantic and structural relationships in the process of integration as shown in the integrated ontology (Figure 6-6) of the test case scenario 2.



Figure 6-6: Integrated Ontology for Test Case Scenario 2, Generated by using Proposed Framework

Similarly, the Figure 6-7 provides the comparison between the pairs of similar concepts returned and the correct pairs found, which are presented through blue and red bars respectively. In this graph, the values of the proposed framework are calculated using the formulae given in the equation of 4.1, 4.2 and 4.3 in (Farooq, 2009). Whereas the values of

SIMTO and other existing techniques are taken from the Table 4.9 (Farooq, 2009). (Maedche and Staab, 2002; Bouquet et al., 2003a; Bouquet et al., 2003b; Kotis and Vouros, 2004; Le et al., 2004; Li, 2004; Hariri, 2006; Alasoud et al., 2008; Trojahn et al., 2008).

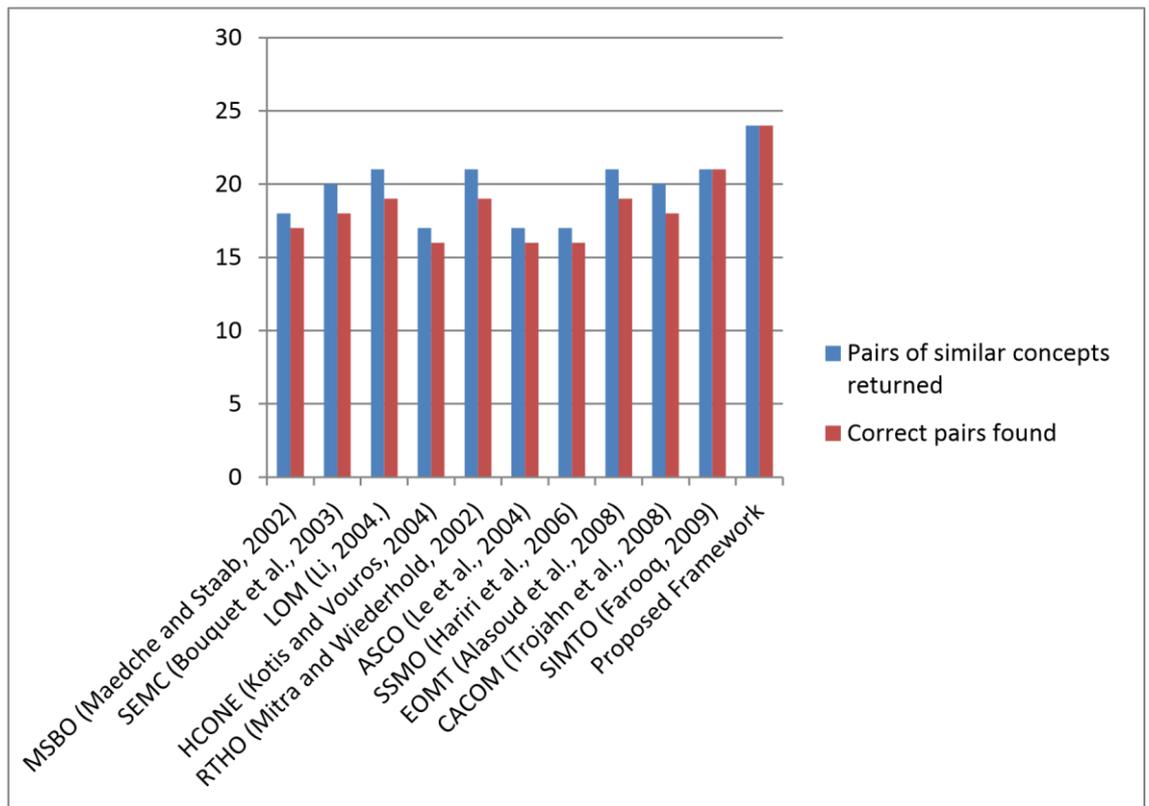


Figure 6-7: Performance Comparison of Test Case Scenario 2 between Proposed Framework and Existing Techniques

Although, SIMTO returned pairs of similar concepts and the correct pairs found are equal to the proposed framework. However, the expected return pair of concepts was different in SIMTO and the proposed framework. The expected return pair of concepts of SIMTO and proposed framework was 21 and 24 respectively.

## **Chapter 7: Conclusion and Future Work**

The chapter provides the conclusion and the future work of the current research.

### **7.1 Conclusion**

The primary aim of this research as specified in chapter 1 was to investigate the area of ontology integration and to provide an advanced prototypical solution in the field of ontology integration. This advanced prototypical solution is presented in chapter 4.

The course of current research started by identifying and reviewing the existing tools and techniques used by other researchers for different concept matching approaches in the concept matching process. All these reviewed techniques were presented and reviewed in chapter 3.

During the analysis of the state of the art, the area of ontology matching has showed that each approach has its own advantages and shortcomings, and the decision about selection of ontology matching approach is totally based upon the scenario of ontology matching and the requirements of ontology matching. The proposed framework of ontology integration supports and uses a hybrid approach of concept matching in order to combine the advantages of the different concept matching techniques. The use of the hybrid approach of concept matching improves the efficiency and more effective for concept matching. The proposed concept matching technique mostly relies on the semantics of the concepts. The concept matching process in the proposed techniques also involves the user confirmation for the complicated cases in terms of their semantics and hierarchy. User involvement does not decrease the efficiency of the concept matching process; however, it improves the quality of the result and decrease the chances of the unwanted or incorrect results.

The comparison of the concepts also involves structural comparison between concepts which enhances the results of the concept matching. The structural comparison helps in taking decision regarding the similarity of the concepts, as the structure contains the information related to the semantic relationship of concepts. A prototype is developed for the implementation of the proposed framework with the use of WordNet as a semantic Vocabulary.

As described in the objectives, there is a need to take some particular scenarios in order to develop the software model and evaluate the system for the effectiveness of the presented

integration framework. Two pairs of reference ontologies are taken from Farooq's PhD thesis (Farooq, 2009) and experiments are conducted in order to analyse and compare the results with existing techniques. The reasons behind the selection of Farooq's experiments are explained in chapter 6.

The current system is developed based upon Farooq's proposed scenarios. This gave a reason to evaluate the efficiency and effectiveness of the current system against SIMTO, a technique of ontology integration proposed and evaluated by Farooq. It is observed and studied that Farooq has evaluated his technique against 9 other existing techniques, therefore his system is used as benchmark.

However, during the evaluation stage the WordNet vocabulary is used as a reference vocabulary. The use of WordNet as a vocabulary can be a limitation of the proposed framework in some scenarios. For example; WordNet does not provides the meaning of the medical terms and the abbreviations associated with different terms. The probability of the incorrect results in the related scenarios. In the same way, the proposed framework is evaluated against the input ontologies of the same domains. Although, there are limitations which are identified, yet these limitations are mainly identified based upon the test scenarios used. It is very much possible that these limitations can improve by using some other ontologies for the test case scenarios.

The proposed framework is an extension towards ontology integration. The proposed approach of the comparing concepts based on their semantics and structure along with their syntax is a better way of concept comparison as identified during the evaluation. This approach may increase the processing time but it is very effective and efficient in order to achieve the required results in ontology integration.

## **7.2 Future Work**

This section provides the potential future work that may help in improving the proposed framework of ontology integration. As discussed earlier, concept matching is the most significant phase in ontology integration. In the testing and analysis phase of the proposed framework, it is observed that the improvement in the concept matching process of the proposed framework may enhance the results of ontology integration. The possible improvements are the following:

The proposed framework can be enhanced by increasing the number of reference ontologies for their integration.

- Domain specific vocabularies should be used with WordNet for the semantic comparison. This will help in implementing the proposed framework across multiple domains.
- String based comparison between concepts can be ignored in order to increase the running time of the concept matching algorithms involved in the proposed framework.
- There is a need to define some standards for ontology development. For example, the standardised domain specific vocabularies should be used in ontology building. This may reduce the problem of heterogeneity between ontologies and the complexity of the ontology integration process.

## References

- Achterbergh, J. and Vriens, D. (2002) Managing Viable Knowledge. *Systems Research and Behavioral Science*, Vol. 19, 3.
- Alasoud, A., Haarslev, V. and Shiri, N. (2008) An Effective Ontology Matching Technique. In: *The 17th International Symposium on Methodologies for Intelligent Systems (ISMIS'08)*. pp. 585-590.
- Alasoud, A. K. (2009) *A Multi-Matching Technique for Combining Similarity Measures in Ontology Integration*. Degree of Doctor of Philosophy (Computer Science), Concordia University.
- Aleksovski, Z., Kate, W. T. and Harmelen, F. V. (2006a) Exploiting the structure of background knowledge used in ontology matching. In: *Proceedings of International Workshop on Ontology Matching collocated with the 5th International Semantic Web Conference*. USA. pp. 13-24.
- Aleksovski, Z., et al. (2006b) Matching Unstructured Vocabularies using a Background Ontology. In: *Proceedings of 15th International Conference on Knowledge Engineering and Knowledge Management Managing Knowledge in a World of Networks*. Czech Republic. pp. 182-197.
- Amardeilh, F. (2009) Semantic Annotation and Ontology Population. In: IGI Global.
- Anderson, P. (2007) What is Web 2.0? Ideas, technologies and implications for education. *JISC Technology and Standards Watch*.
- Aufaure, M.-A., et al. (2008) Metadata and Ontology-Based Semantic Web Mining. *Data Warehousing and Mining Concepts Methodologies Tools and Applications*, Vol. 33, pp. 531-3556.
- Bach, T. L. and Dieng-Kuntz, R. (2005) Measuring Similarity of Elements in OWL DL Ontologies. In: *Contexts and Ontologies AAAI Workshop*. Springer.
- Balani, N. (2005) The future of the Web is Semantic: Ontologies form the backbone of a whole new way to understand online data [WWW]. Available from:  
<http://www.ibm.com/developerworks/web/library/wa-semweb/> [Accessed 15 August 2011].
- Batet, M., Sánchez, D. and Valls, A. (2010) An ontology-based measure to compute semantic similarity in biomedicine. *Journal of Biomedical Informatics*, Vol. 44, 1, pp. 118-125.
- Bendeck, F. (2008) *WSM-P Workflow Semantic Matching Platform*. PhD, University of Trier.
- Berners-Lee, T. (1998) The World Wide Web: A very short personal history [WWW]. Available from: <http://www.w3.org/People/Berners-Lee/ShortHistory.html> [Accessed 19 November 2011].
- Berners-Lee, T., Hendler, J. and Lassila, O. (2001) The Semantic Web. *Scientific American Magazine* May, 2001.

- Birkinshaw, J. and Sheeha, T. (2002) Managing the Knowledge Life Cycle. *MITSloan Management Review*, Vol. 24, 1.
- Borst, P. and Kkermans, H. A. (1997) Engineering ontologies. *Int. J. Human – Computer Studies*, Vol. 46, pp. 365 – 406.
- Bouquet, P., et al. (2003a) A SAT-based algorithm for context matching. In: *Proceedings of the 4th international and interdisciplinary conference on Modeling and using context*. Stanford, CA, USA. Springer-Verlag.
- Bouquet, P., Serafini, L. and Zanobini, S. (2003b) Semantic Coordination: A New Approach and an Application. *ISWC 2003, LNCS 2870*, pp. 130-145.
- Bouquet, P., et al. (2006) Bootstrapping semantics on the web: meaning elicitation from schemas. In: *Proceedings of the 15th international conference on World Wide Web*. Edinburgh, Scotland. ACM.
- Broekstra, J., et al. (2002) Enabling knowledge representation on the Web by extending RDF Schema. *Computer Networks*, Vol. 39, 5, pp. 609-634.
- Cassia, T., et al. (2008) A cooperative approach for composite ontology mapping. In: Stefano, S. (ed.) *Journal on data semantics X*. Springer-Verlag. pp. 237-263.
- Celjuska, D. and Vargas-Vera, D. M. (2004) Ontosophie: A Semi-Automatic System for Ontology Population from Text.
- Chandrasekaran, B., Josephson, J. R. and Benjamins, V. R. (1999) What Are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems*, Vol. 14, 1, pp. 20-26.
- Cormode, G. and Krishnamurthy, B. (2008) Key differences between Web 1.0 and Web 2.0. *Cormode*, Vol. 13, 6.
- Cristani, M. and Cuel, R. (2005) A survey on ontology creation methodologies. *International Journal on Semantic Web & Information Systems*, Vol. 1, 2, pp. 49-69.
- Curtis, J., Cabral, J. and Baxter, D. (2006) On the Application of the Cyc Ontology to Word Sense Disambiguation. In: *The 19th International Florida Artificial Intelligence Research Society Conference*.
- Cycorp (n.d) Cycorp [WWW]. Available from: <http://www.cyc.com/> [Accessed 16 October 2011].
- Daml-Oil (2001) DAML+OIL (March 2001) [WWW]. Available from: <http://www.daml.org/2001/03/daml+oil-index> [Accessed 12 December 2011].
- Doan, A., et al. (2003) Ontology Matching: A Machine Learning Approach. In: *Handbook on Ontologies in Information Systems*. pp. 397-416.
- Dong, Z., Dong, Q. and Hao, C. (2010) HowNet and its computation of meaning. In:

- Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*. Beijing, China. Association for Computational Linguistics.
- Ehrig, M. and Staab, S. (2004) QOM -- Quick Ontology Mapping. In: *3rd International Semantic Web Conference (ISWC2004)*. Hiroshima, Japan.
- Euzenat, J. (1993) Brief overview of T-TREE: the TROPES Taxonomy building Tool. In: *4th ASIS SIG/CR Classification Research Workshop*.
- Knowledge web NoE (2004) *State of the art on ontology alignment*.
- Euzenat, J. and Valtchev, P. (2004) Similarity-based ontology alignment in OWL-lite. In: *15th European Conference on Artificial Intelligence (ECAI)*. Valencia (ES). pp. 333337.
- Evans, D. M. (2006) *The Evolution of the Web - From Web 1.0 to Web 4.0*. School of Systems Engineering, University of Reading.
- Farooq, A. (2009) *Similarity Identification and Measurement between Two Web Ontologies*. Doctor of Philosophy (PhD), University of Engineering & Technology.
- Farooq, A. and Shah, A. (2008) Ontology Development Methodology for Semantic Web Systems. *Pakistan Journal of Life and Social Sciences*, Vol. 6, 1, pp. 50-58.
- Farooq, A. and Shah, A. (2010) Similarity Identification and Measurement between Ontologies. *Journal of American Science*, Vol. 6, 4, pp. 67-85.
- Farooq, A., Shah, A. and Akram., Q. (2006) A Technique for Generating Semantic Web Documents. In: *Proceedings of The 2006 International Conference on Information & Knowledge Engineering (IKE'06)*. Las Vegas, Nevada, USA. CSREA Press, pp. 26-29.
- Farooq, A., Shah, A. and Asif, K. H. (2007) Design of ontology in Semantic Web engineering process. In: *High Capacity Optical Networks and Enabling Technologies*. pp. 1-6.
- Farquhar, A., et al. (1995) Collaborative Ontology Construction for Information Integration.
- Fensel, D. (2001) *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. 2nd edition. Berlin, Springer-Verlag.
- Fensel, D., et al. (2001) OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems*, Vol. 16, 2, pp. 38-45.
- Fensel, D., et al. (2000) On-to-knowledge: Ontology-based tools for knowledge management. In: *Proceedings of the eBusiness and eWork*. Citeseer, pp. 18-20.
- Futureexploration.Net (2007) *WEB 2.0 Framework*. Future Exploration Network.

- Gerald, R., Harald, G. and Mehdi, J. (2005) WEESA: Web engineering for semantic Web applications. In: *Proceedings of the 14th international conference on World Wide Web*. Chiba, Japan. ACM.
- Giunchiglia, F., Yatskevich, M. and Shvaiko, P. (2007) Semantic Matching: Algorithms and Implementation. *Journal On Data Semantics*.
- Gómez-Pérez, A., Fernández-López, M. and Corcho, O. (2004) *Ontological Engineering: With Examples from the Areas of Knowledge Management, E-commerce and the Semantic Web*. Springer.
- IST Project IST-2000 (2003) *OntoWeb D.1.5 A survey of ontology learning methods and techniques*.
- Gruber, T. (1992) What is an Ontology? [WWW]. Available from: <http://www.wksi.stanford.edu/kst/what-is-an-ontology.html> [Accessed 5 April 2011].
- Gruber, T. (2006) Where the Social Web Meets the Semantic Web. In: *Tenth IEEE International Symposium on Wearable Computers, ISWC 2006*. Montreux, Switzerland. Springer, p. 994.
- Gruber, T. R. (1993a) Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies*, Vol. 43, pp. 907-928.
- Gruber, T. R. (1993b) A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, Vol. 5, 2, pp. 199-220.
- Hariri, B. B. (2006) A new Structural Similarity Measure for Ontology Alignment. *Computer Engineering*, pp. 36-24.
- Holsapple, C. W. and Joshi, K. D. (2002) A collaborative Approach to Ontology Design. *Communications of the ACM*, Vol. 45, 2, pp. 42-47.
- Hyun-Je, S., Seong-Bae, P. and Se-Young, P. (2009) An automatic ontology population with a machine learning technique from semi-structured documents. In: *Information and Automation, 2009. ICIA '09*. pp. 534-539.
- Kalfoglou, Y. and Schorlemmer, M. (2003) IF-Map: an ontology mapping method based on Information Flow theory. *Journal on Data Semantics* Vol. 1, 1, pp. 98-127.
- Keet, C. M. M. (2004) *Aspects of ontology integration*. Literature research & background information for the PhD proposal entitled: Bottom-up development of ontologies and ontology integration in the subject domain of ecology, Napier University.
- Khodaei, A., Abolhassani, H. and Hariri, B. B. (2006) A new Structural Similarity Measure for Ontology Alignment. In: *Proceedings of the 2006 International Conference on Semantic Web & Web Services*. USA. pp. 36-42.

- Kinsella, S., et al. (2008) From Web 1.0 to Web 2.0 and back -: how did your grandma use to tag? In: *Proceedings of the 10th ACM workshop on Web information and data management*. Napa Valley, California, USA. ACM.
- Kotis, K. and Vouros, G. A. (2004) The HCONE Approach to Ontology Merging. In: *The First European Semantic Web Symposium*. Springer, pp. 137-151.
- Kumar, S., et al. (2011) A Semantic Web Engineering Model.
- Lacher, M. S. and Groh, G. (2001) Facilitating the Exchange of Explicit Knowledge through Ontology Mappings. In: *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference*. AAAI Press.
- Le, B. T., Dieng-Kuntz, R. and Gandon, F. (2004) ON ONTOLOGY MATCHING PROBLEMS for building a corporate Semantic Web in a multi-communities organization. In: *Proceedings of the 6th International Conference on Enterprise Information Systems*.
- Le, B. T., Dieng-Kuntz, R. and Gandon, F. (2005) ON ONTOLOGY MATCHING PROBLEMS for building a corporate Semantic Web in a multi-communities organization. In: *Proceeding of the Sixth International Conference on Enterprise Information Systems*. Porto - Portugal.
- Lee, Y., Supekar, K. and Geller, J. (2006) Ontology integration: Experience with medical terminologies. *Computers in Biology and Medicine*, Vol. 36, 7, pp. 893-919.
- Lepore, E. (2011) semantics. In: *Encyclopedia Britannica Online*, Encyclopædia Britannica Inc., 2012.
- Li, J. (2004) LOM: A Lexicon-based Ontology Mapping Tool In: *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS)*.
- Li, Y., et al. (2006) Result of Ontology Alignment with RiMOM at OAEI'06. In: *Ontology Alignment Evaluation Initiative*.
- Lopez, M. F., Perez, A. G. and Juristo, N. (1997) METHONTOLOGY: from Ontological Art towards Ontological Engineering. In: *The AAAI97 Spring Symposium*. pp. 33-40.
- Loup, D. (n.d.) Ontology Alignment [WWW]. Available from:  
<http://www.iro.umontreal.ca/~owlola/alignment.html> [Accessed 15 December 2011].
- Maedche, A. and Staab, S. (2002) Measuring Similarity between Ontologies. *Knowledge Engineering and Knowledge Management Ontologies and the Semantic Web*, Vol. 2473, pp. 251-263.
- Michael, U. (2003) Where are the semantics in the semantic web? *AI Mag.*, Vol. 24, 3, pp. 2536.

- Mitra, P., Wiederhold, G. and Jannink, J. (1999) Semi-automatic Integration of Knowledge Sources. In: *Proceedings of Fusion 99* Sunnyvale, USA. pp. 572-581.
- Department of Information Engineering and Computer Science, University of Trento, Italy (2000) *A Graph-Oriented Model for Articulation of Ontology - DISI*. Italy.
- Nejdl, W., Wolpers, M. and Capelle, C. (2000) The RDF Schema Specification Revisited. In: *Modellierung*.
- The third Summer School on Ontological Engineering and the Semantic Web. (2005) *Ontology Mapping and Alignment*.
- Noy, N. F. and Musen, M. A. (2000) PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*. AAAI Press.
- Noy, N. F. and Musen, M. A. (2001) Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In: *Proceedings Of The Workshop On Ontologies And Information Sharing At The International Joint Conference On Artificial Intelligence*.
- O'leary, D. E. (2002) Knowledge management across the enterprise resource planning systems life cycle. *International Journal of Accounting Information Systems*, Vol. 3, 2, pp. 99-110.
- Özsu, M. T. and Liu, L. (2009) *Encyclopedia of Database Systems*. 1 edition. New York, USA, Springer.
- Patel-Schneider, P. F. and Fensel, D. (2002) Layering the Semantic Web: Problems and Directions. In: *Proceedings of the First International Semantic Web Conference on The Semantic Web*. Sardinia, Italy. Springer-Verlag.
- Pedersen, T., et al. (2007) Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics*, Vol. 40, 3, pp. 288-299.
- Pinto, H. S. (1999) Some Issues on Ontology Integration. *Praxis* Vol. 18, pp. 1-12.
- Pinto, H. S. and Martins, J. P. (2004) Ontologies: How can They be Built? *Knowl. Inf. Syst.*, Vol. 6, 4, pp. 441-464.
- Pisanelli, D. M., Gangemi, A. and Steve, G. (1999) A Medical Ontology Library That Integrates the UMLS MetathesaurusTM. In: *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*. Springer-Verlag.
- Plessers, P., Casteleyn, S. and Troyer, O. D. (2005) Semantic Web Development with WSDM.

- Princeton.Edu (n.d.) WordNet [WWW]. Available from: <http://wordnet.princeton.edu/> [Accessed 11 January 2012].
- Cycorp, Inc (2002) *Mapping Ontologies into Cyc*.
- Reif, G. and Gall, H. (2006) Using WEESA to Semantically Annotate Cocoon Web Applications. In: *1st Semantic Authoring and Annotation Workshop*. Geogria, USA. pp. 5-9.
- Reif, G., Jazayeri, M. and Gall, H. (2004) Towards Semantic Web Engineering: WEESA - Mapping XML Schema to Ontologies. In: *In Workshop on Application Design, Development and Implementation Issues in the Semantic Web at the 13th International World Wide Web Conference*. p. 105.
- Sampson, J. (2005) Ontology Alignment in Agent Systems: Current and future challenges. In: *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce(CIMCA-IAWTIC'05)*. Vienna, Austria.
- Royal Institute of Technology (2005) *The Philosophy of the Grid: Ontology Theory - From Aristotle to Self-Managed IT Resources*. Sandholm, T. Stockholm, Sweden.
- Seco, N. A. L. (2005) *Computational Models of Similarity in Lexical Ontologies*. Masters in Computer Science, University College Dublin.
- Shah, A. F. A. A. (2010) Similarity Identification and Measurement between Ontologies. *Journal of American Science*, Vol. 6, 4, pp. 67-85.
- Simon, J., et al. (2006) Formal ontology for natural language processing and the integration of biomedical databases. *International Journal of Medical Informatics (2006)*, Vol. 45, 34, pp. 224-231.
- Smith, B. and Welty, C. (2001) FOIS introduction: Ontology--towards a new synthesis. In: *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*. Ogunquit, Maine, USA. ACM.
- Straccia, U. and Troncy, R. (2005) oMAP: Combining classifiers for aligning automatically OWL ontologies. *Web Information Systems Engineering--WISE 2005*, Vol. 3806, pp. 133-147.
- Stumme, G., Hotho, A. and Berendt, B. (2006) Semantic Web Mining: State of the art and future directions. *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 4, 2, pp. 124-143.
- Stumme, G. and Maedche, A. (2001) FCA-MERGE: Bottom-Up Merging of Ontologies.
- Styltsvig, H. B. (2006) *Ontology-based Information Retrieval*. PhD, Roskilde University.
- Suchanek, F. M., Kasneci, G. and Weikum, G. (2007) YAGO: A Core of Semantic

- Knowledge Unifying WordNet and Wikipedia. In: *The 16th international conference on World Wide Web*. Banff, Alberta, Canada. ACM.
- Thomas, R. G. (1995) Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, Vol. 43, 5-6, pp. 907-928.
- Trojahn, C., et al. (2008) A Cooperative Approach for Composite Ontology Mapping. . *Journal on Data Semantics X*, pp. 237-263.
- Uschold, M. and King, M. (1995) Towards a Methodology for Building Ontologies. In: *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95*.
- Valarakos, A., et al. (2004) Enhancing Ontological Knowledge Through Ontology Population and Enrichment. In: *Engineering Knowledge in the Age of the Semantic Web*. Berlin, Springer. pp. 144-156.
- Veale, T. (2005) Analogy generation with HowNet. In: *Proceedings of the 19th international joint conference on Artificial intelligence*. Edinburgh, Scotland. Morgan Kaufmann Publishers Inc.
- Vernadat, F. B. (2007) Interoperable enterprise systems: Principles, concepts, and methods. *Annual Reviews in Control*, Vol. 31, 1, pp. 137-145.
- W3C (1999) Resource Description Framework (RDF) Model and Syntax Specification [WWW]. Available from: [Accessed 14 November 2011].
- W3C (2004) Resource Description Framework (RDF) [WWW]. Available from: <http://www.w3.org/RDF/> [Accessed 14 September 2011].
- W3C (n.d.) Semantic Web [WWW]. Available from: <http://www.w3.org/standards/semanticweb/> [Accessed 29 March 2011].
- Wache, H., et al. (2001) Ontology-Based Integration of Information - A Survey of Existing Approaches. In: *IJCAI--01 Workshop: Ontologies and Information Sharing (2001)*. pp. 108-117.
- Wang, L. and Hongshuai, Z. (2010) Ontology for Communication in Distributed Multi-agent System. In: *Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science*. Hong Kong, China.
- Xia, M. and Wang, J. A. (2009) Temporal Text Extraction and Automated Time-OWL Population. In: *ASEE Southeast Section Conference*.
- Zhang, J. and Olango, P. (2005) Populating an Ontology Using Wikipedia's Taxonomy To Know More. *Information Science*.