SIMILARITY IDENTIFICATION AND MEASUREMENT BETWEEN TWO WEB ONTOLOGIES

AMJAD FAROOQ
(2005-PhD-CS-03)

SUPERVISED BY
PROF. DR. ABAD ALI SHAH

(2009)

Department of Computer Science and Engineering
University of Engineering and Technology
Lahore, Pakistan
Similarity Identification and Measurement between Two Web Ontologies

Dissertation
Submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Computer Science
(2009)

AMJAD FAROOQ
(2005-PhD-CS-03)

Supervised By
Prof. Dr. Abad Ali Shah

Department of Computer Science and Engineering, University of Engineering & Technology, Lahore-Pakistan
Similarity Identification and Measurement between Two Web Ontologies

A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Computer Science

By

Amjad Farooq

Approved on: __________

Internal Examiner: ______________

Prof. Dr. Abad Ali Shah
Department of Computer Science & Engineering
U.E.T., Lahore, Pakistan

External Examiner: ______________

Prof. Dr. Irfan Younus
Pro-Vice Chancellor
HITEC University, Taxila

_________________________ ________________
Chairman Dean
Department of CS & E Faculty of Electrical Engineering
ABSTRACT

Nowadays, the web is the main collection of information resources but the retrieval of relevant and precise information from the web is a serious problem. The reason of this problem is the way web-contents are stored and formalized. With the realization of the ontology-based web, it is being tried to overcome this problem, but the ontologies may themselves suffer from heterogeneity when they are simultaneously used. Then, a need arises to identify and measure the similarity between concepts of those ontologies to overcome the said problem. Many investigators did work on this problem but still some issues need to be addressed. In this dissertation, a semi-automatic technique has been proposed for similarity identification and measurement between two web ontologies. The explicit semantics of concepts, in addition to their linguistic and taxonomic characteristics, is the key consideration and feature of the proposed technique. This proposed technique identifies all candidate pairs of similar concepts without omitting any similar-pair. The semantic relations and degree of similarity computed from new criteria, make the technique well-suited with the theme of semantic web. This technique can be used with different types of operations on ontology such as merging, mapping and aligning. We have implemented the proposed technique in Java, and validated it through different case studies. By comparing and analyzing the results of these case studies, a reasonable improvement in terms of completeness, correctness and overall quality of results, has been found.
ACKNOWLEDGEMENT

First of all, my thanks remain forever to ALLAH, Who guides me in every aspect of life. I am deeply thankful to my advisor Professor Dr. Abad Ali Shah, who has initiated me to research. He provided me full guidance during the research work. I am really grateful to him for his valuable comments and suggestions about research work. His attentive presence has enabled me punctual, consistent and persistent for doing research. I would like to say thank to Dr. Mohammad Ali Moud, the Chairman CSE Department, who has made me very comfort to do my research work. All this has become possible by the effort of Vice Chancellor who has provided a suitable research environment in the university. I am grateful to my colleagues for their healthy criticism and suggestions.

There is a special thanks to my parents for their unconditional support throughout my study. My wife also deserves an individual appreciation for her prayers and love.

I would also like to appreciate my student Mr. Kashif Bashir for his help in implementation needed to validate my work.

I acknowledge the financial support by the University of Engineering and Technology, Lahore and the Higher Education Commission of Pakistan.

Finally, again I am grateful to God, without His uncountable blessings my research work has been impossible.
Dedicated to my Well Wishers
# TABLE OF CONTENTS

## 1. INTRODUCTION

01

## 2. BACKGROUND AND LITERATURE REVIEW

06

2.1. Semantic Web VS Current Web

2.2. The Web Ontology

2.3. Engineering Ontologies for Semantic Web

2.4. Populating Web Ontologies

2.5. Integrating Web Ontologies

## 3. PROPOSED TECHNIQUE FOR SIMILARITY IDENTIFICATION AND MEASUREMENT

28

3.1 The Basis of Proposed Technique

3.1.1. Comparison of Concepts Rather Than Terms

3.1.2. Use of Domain-specific Semantics Rather Than Linguistic Semantics

3.1.3. Super-Concepts Based Contextual Similarity Measurement

3.1.4. Explicit Semantics Based Measurement of Degree of Similarity (DoS) and Semantic Relation (SR)

3.1.5. The Add-on of Both the DoS and SR with Each Pair of Similar Concepts

3.1.6. Matching Strategy

3.2 Definitions and Structure Diagram

3.3 IPS - Identifying Primary Similarity

3.4 ICS - Identifying Contextual Similarity
4. IMPLEMENTATION AND TESTING

4.1 Implementation

4.1.1. Main Module

4.1.2. Ontology Loading Module

4.1.3. Concepts Acquisition Module

4.1.4. Primary Similarity Measurement Module

4.1.5. Super-concepts Acquisition Module

4.1.6. Contextual Similarity Measurement Module

4.1.7. Roles Acquisition Module

4.1.8. Role-based Similarity Measurement Module

4.2. Testing

4.2.1. Case Study 1: Software Development Organization

4.2.1.1. List of Roles

4.2.1.2. List of Concepts of First Input Ontology

4.2.1.3. List of Concepts of Second Input Ontology

4.2.1.4. List of Pairs of Similar Concepts

4.3. Case Study 2: University – The Education Organization

4.3.1. List of Sample Concepts of Input Ontologies

4.3.2. Primary Similarity Identification and Measurement

4.3.3. Contextual Similarity Identification and Measurement

4.3.4 Role-based Similarity Identification and Measurement
4.4. Observations about Results of Different Test-scenarios .......................... 66
4.5. Results: Analysis and Discussion ........................................................................ 67
  4.5.1. Completeness ........................................................................................................ 68
  4.5.2. Correctness .......................................................................................................... 68
  4.5.3. Overall Quality of Result ...................................................................................... 68
  4.5.4. Test Cases for Evaluating Performance ................................................................. 69
    4.5.4.1. Evaluating Performance with First Test Case .................................................. 70
    4.5.4.2. Evaluating Performance with Second Test Case ............................................. 72
    4.5.4.3. Evaluating Performance with Third Test Case ............................................... 75
    4.5.4.4. Evaluating Performance with Fourth Test Case ............................................. 77
4.6. Discussion about Overall Performance ................................................................. 80
5. CONCLUSION AND FUTURE WORK ................................................................. 82
  5.1 Conclusion ............................................................................................................... 82
  5.2 Recommendations for Further Research ................................................................. 84
REFERENCES .............................................................................................................. 86
APPENDICES
I.  Ontology of a person’s family ................................................................................. 97
II. A Subset of Roles from Software Development Organization .................................. 100
III. Domain Vocabulary of a Software Development Organization ............................. 111
IV. datasoft.owl .......................................................................................................... 118
V.  ridos.owl .............................................................................................................. 125
LIST OF FIGURES

2.1 A layered model of semantic web ------------------------------------------ 11
2.2 A sample slice of person’s family ontology--------------------------------- 12
2.3 A slice of taxonomy of furniture ontology-A -------------------------------- 25
2.4 A slice of taxonomy of furniture ontology-B -------------------------------- 26
2.5 A slices of ontologies A & B of a university domain------------------------- 26
3.1 Concept vs. terms --------------------------------------------------------- 29
3.2: The use of concepts rather than terms in measurement of DoS------------- 29
3.3 The DoS through edit-distance based formula and through proposed formula-- 29
3.4 Similarity measurement strategy used in proposed technique----------------- 32
3.5 Degree of similarity measurement through existing techniques--------------- 33
3.6 Degree of similarity measurement through proposed technique

  along with new criterion (role-based) ---------------------------------------- 34
3.7 Similarity measurement strategy used in current techniques----------------- 34
3.8 Structure diagram of proposed technique------------------------------------ 40
3.9 A slice of pseudo code for acquisition of concepts from input ontologies -- 42
3.10 A slice of pseudo code for identifying primary similarity------------------- 43
3.11 Acquisition of parents of primarily similar concepts----------------------- 45
3.12 A slice of pseudo code for identifying contextual similarity--------------- 46
3.13 Acquisition of roles of contextually similar concepts----------------------- 49
3.14 A slice of pseudo code for identifying role-based similarity-------------- 51
4.1 A prototype for similarity identification and measurement------------------ 54
4.2 A subset of roles in software development organization ------------------------------------------------- 59
4.3 Completeness wise comparison of results with respect to first test case -------------------------- 71
4.4 Correctness wise comparison of results with respect to first test case ---------------------------- 71
4.5 Overall quality wise comparison of results with respect to first test case ---------------------- 72
4.6 Completeness wise comparison of results with respect to second test case -------------------- 73
4.7 Correctness wise comparison of results with respect to second test case ---------------------- 74
4.8 Overall quality wise comparison of results with respect to second test case ------------------- 74
4.9 Completeness wise comparison of results with respect to third test case -------------------- 76
4.10 Correctness wise comparison of results with respect to third test case --------------------- 76
4.11 Overall quality wise comparison of results with respect to third test case ------------------- 77
4.12 Completeness wise comparison of results with respect to fourth test case ------------------ 78
4.13 Correctness wise comparison of results with respect to fourth test case ------------------- 79
4.14 Overall quality wise comparison of results with respect to fourth test case ------------------ 79
4.15 Role-based similarity: expected results vs. produced results----------------------------- 80
LIST OF TABLES

2.1 Relevant results (from top 20 results) of search some engines -------------------------- 07
2.2 Semantic web vs. current web------------------------------------------------------- 08
2.3 Examples of semantic web systems --------------------------------------------------- 10
2.4 A comparison of some techniques for similarity measurement between ontologies ---- 23
3.1 Lexical & linguistic similarity matchers vs. 1st level matcher ----------------------- 35
3.2 Contextual similarity matcher vs. 2nd level matcher ---------------------------------- 35
3.3 Third level or explicit semantic matcher--------------------------------------------- 35
3.4 Existing techniques vs. proposed technique------------------------------------------- 52
4.1 A sample slice of intellectual concepts form A ontology --------------------------- 60
4.2 A sample slice of intellectual concepts form B ontology------------------------------- 61
4.3 A sample list of similar concepts, including pairs having minimum degree of similarity--------------------------------------------- 61
4.4 A slice of role-based similar concepts with a threshold-value ---------------------- 62
4.5 Different scenarios for a pair of concepts to similarity identification ------------ 63
4.6 Results of first, second and third levels of similarities------------------------------- 65
4.7 Observations about results of different test-scenarios ----------------------------- 66
4.8 Test-cases for evaluating performance of different techniques ----------------------- 69
4.9 Results of different techniques with first test case ------------------------------ 70
4.10 Results of different techniques with second test case ---------------------------- 73
4.11 Results of different techniques with third test case ----------------------------- 75
4.12 Results of different techniques with fourth test case ---------------------------- 77
4.13 Test cases along with status of WordNet and Domain Vocabulary ------------------- 81
4.14 Comparison of results related to Heterogeneous Pairs of Similar Concepts (HPoSCs) 81
LIST OF PUBLICATIONS


LIST OF EQUATIONS

2.1 Degree of similarity based on an existing technique ........................................ 24
2.2 Value-range for degree of similarity ................................................................. 24
2.3 Semantic relation between two concepts ......................................................... 25
3.1 A set of terms to name a concept ................................................................. 36
3.2 A set of super-concepts of a concept ............................................................ 36
3.3 A set of sub-concepts of a concept ............................................................... 36
3.4 A set of siblings of a concept ........................................................................ 36
3.5 A set of roles of a concept .............................................................................. 36
3.6 Explicit semantics of a concept ................................................................. 36
3.7 Definition of primary similarity ....................................................................... 37
3.8 Definition of contextual similarity ............................................................... 37
3.9 Definition of role-based similarity ............................................................... 37
3.10 Role-based degree of similarity ................................................................. 38
3.11 Role-based semantic relation ................................................................. 38
3.12 Granularity of a concept .............................................................................. 39
3.13 Granularity-based degree of similarity ....................................................... 39
3.14 Granularity-based semantic relation .......................................................... 40
3.15 A set of concepts of first input ontology ...................................................... 42
3.16 A set of concepts of second input ontology ................................................. 42
3.17 A set of pairs of primarily similar concepts ............................................... 44
3.18 A set of super-concepts of primarily similar concepts of first ontology .. 44
3.19 A set of super-concepts of primarily similar concepts of second ontology .. 44
3.20 A set of pairs of contextually similar concepts -------------------------------- 47
3.21 A set of roles of contextually similar concepts of first input ontology --------- 47
3.22 A set of roles of contextually similar concepts of second input ontology ------- 47
3.23 A set of pairs of concepts possessing role-based similarity --------------------- 48
4.1 The measurement of completeness of result-------------------------------------- 68
4.2 The measurement of correctness of result---------------------------------------- 68
4.3 The measurement of overall quality of result----------------------------------- 68
Chapter 1

INTRODUCTION

The World Wide Web is a global information space. It includes information about almost every topic of the real world. Due to semantic heterogeneity and the lack of machine understandability of web contents, we use the keyword-based search engines to retrieve the desired information. The web pages containing any keyword of the search criteria are returned. The returned web pages are, then, further searched manually to get to the relevant and desired web pages. Also, certain relevant web pages may not be returned by the search engine due to the mismatch of keywords (of search criteria). It has been estimated that only 37 percent to 52 percent relevant results are retrieved and other retrieved results are irrelevant [9]. Theses statistics are unacceptable.

The concept of semantic web was envisioned by Lee [51], and it provides a promising solution to overcome the problems of current web. According to the theme of the semantic web, the web-content need to be structured, formalized and stored through ontologies to enable them machine-understandable. However, the ontologies may themselves suffer from semantic heterogeneity when they are simultaneously used in some operations such as alignment, merging, mapping and sharing and acquiring information [2, 3, 7, 24].

To handle this problem of ontological semantic heterogeneity, it is required to identify the similarity between ontologies. A lot of work has been done on similarity identification and measurement between ontologies [6, 16, 23, 26, 28-30, 33, 35, 57-61]. However, several
issues are still unsolved. In this dissertation, we concentrate at only a few issues which are listed below.

(i) The similarity identification and measurement process should incorporate the recommendations of World Wide Web Construction (W3C) related to the semantic web [8, 14, 21]. According to these recommendations the explicit semantics similarity between concepts of ontologies should be identified and measured. The explicit semantics of an intellectual concept is usually defined by the roles it keeps, in its respective domain. Most of the existing works, as cited easier, are about the measurement of similarity between two concepts based on their names, linguistic semantics, and the similarities of their taxonomic characteristics such as super-concepts, sub-concepts and sibling-concepts. However, no attention has been given on the explicit semantics based similarity measurement between concepts of ontologies.

(ii) The measurement of degree of similarity (DoS) based on Edit-distance formula, is not reliable because the DoS is measured based on terms rather than concepts represented by those terms [35, 58, 59, 85]. It may produce inaccurate results in various scenarios. That is, some pairs of similar concepts are declared dissimilar because of the heterogeneous terms used for the names those concepts. Similarly, some pairs of dissimilar concepts are declared similar because of the similarity of terms used for those concepts.

(iii) The existing criteria as reported in [5, 9, 27, 62] used for the identifying taxonomic similarity between concepts of two ontologies declare certain pairs of similar concepts as dissimilar due to the biasness of these criteria towards those concepts whose siblings-concepts, sub-concepts or direct super-concepts are not similar. This also suggests that there is a need of change in the measurement criteria of taxonomic similarity.
Most of the existing similarity measurement techniques compute only the DoS between concepts of two ontologies [4, 34, 41, 86]. The value of DoS remains between 0 and 1 which is inadequate to determine as to which concept is more generic or more specific than the other one. It has been considered as open research issue [56]. Similarly, some existing techniques compute only the Semantic Relation (SR) between two concepts [4, 55]. Although, SR shows that one concept is more generic, or more specific than the other concept, yet it does not give the level of generality. Therefore, each pair of similar concepts should be accompanied by their both measurements DoS and SR in order to take a better decision while performing the aligning, merging and mapping operations of ontologies.

The measurement of semantic similarity is a complex and inefficient in execution-wise [56]. We have empirically observed that the lexically dissimilar concepts are always taxonomically dissimilar. Similarly the taxonomically dissimilar concepts are always explicit semantically dissimilar. It means that there is no need to measure the taxonomic similarity between lexically dissimilar concepts and also there is no need to measure the explicit semantic similarity between taxonomically dissimilar concepts. This suggests that the measurement process should be in some layered fashion to make it more efficient.

Based on above motivations, we feel that there should be an integrated technique, based on innovative vision of semantic web, to identify and measure the lexical, taxonomic and explicit semantic similarities between concepts of ontologies.

In this dissertation, we propose a new technique for identifying and measuring similarity between concepts of two ontologies to achieve the following objectives:
(i). Identifying all pair of similar concepts with out omitting or overlooking any candidate pair of similar concepts.

(ii). Identifying and measuring the explicit semantic similarity between intellectual concepts of ontologies.

(iii). Improving the similarity measurement process to make it efficient, complete and realistic.

The rest of dissertation is structured as follows: Chapter 2 presents background of the area of research and the relevant literature. The research domain includes semantic web, ontologies, reuse of ontologies and similarity identification and measurement between ontologies. Reviewed literature presented in the same chapter, has been divided into three sub-sections: First section includes the brief description of semantic web, technologies involved, its standards and its current state of art. A comparison is tried between current web and semantic web. The second includes literature overview about structure, development and populating of ontologies. Different methodologies for developing ontologies are overviewed. Their similarities, differences and shortcomings are specially taken into consideration. Similarly, different methodologies used for populating ontologies are also reviewed in same section. The third section is about the reuse of ontologies in different operations such as merging, aligning, mapping, querying and even engineering new ontologies. When multiple ontologies are reused simultaneously, heterogeneities usually arise and then to handle the situation, the similarity between their respective concepts needs to be identified. A literature surveyed about similarity identification between concepts of ontologies is presented in order to determine the current state of art of similarity identification between concepts of ontologies. Chapter 3 presents the proposed technique for identifying similarities between
concepts of ontologies. The brief introduction of technique is followed by its structure diagram, and then its phases are described. The working of each phase has also been listed in algorithmic form. Chapter 4 presents the validation of proposed technique, manually, and through implementation. The results and the discussions about results are also presented in same chapter. Chapter 5 concludes the thesis along with some recommendation for future work.
Chapter 2

BACKGROUND AND LITERATURE REVIEW

The web information resources are growing explosively in number and volume but to retrieve relevant information from these resources are becoming more difficult and time-consuming. The way of storing web information resources is considered a root cause of irrelevancy in information retrieved because they are not stored in machine-understandable organization. It has been estimated [9] that nearly 48 percent to 63 percent irrelevant results are retrieved. That is, the relevant results are in the rage of 37 percent to 52 percent, as shown in Table 2.1, which are far from the accuracy. An extension of current web known as semantic web, has been envisioned to organize the web content through ontologies in order to enable them machine understandable. To support the theme of semantic web, there is a crucial need of techniques for designing, development, populating and integrating ontologies.

2.1 Semantic Web VS Current Web

Semantic web may be compared with non-semantic web within several parameters such as content, conceptual perception, scope, environment and resource-utilization.

(a) Content: Semantic web encompasses actual content along with its formal semantics. Here, the formal semantics are machine understandable content, generated in logic-based languages such as Web Ontology Language (OWL) recommended by W3C. Through formal semantics of content, computers can make inferences about data i.e., to understand what the
Table 2.1: Relevant results (from top 20 results) of search some engines

<table>
<thead>
<tr>
<th>Search Engine</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yahoo</td>
<td>0.52</td>
</tr>
<tr>
<td>Google</td>
<td>0.48</td>
</tr>
<tr>
<td>MSN</td>
<td>0.37</td>
</tr>
<tr>
<td>Ask</td>
<td>0.44</td>
</tr>
<tr>
<td>Seekport</td>
<td>0.37</td>
</tr>
</tbody>
</table>

data resource is and how it relates to other data. In today’s web there is no formal semantics of existing contents. These content are machine-readable but not machine understandable.

(b) **Conceptual Perception:** Current web is just like a book having multiple hyperlinked documents. In book scenario, index of keywords are presented in each book but the contexts in which those keywords are used, are missing in the indexes. That is, there are no formal semantic of keywords in indexes. To check which one is relevant, we have to read the corresponding pages of that book. Same is the case with current web. In semantic web this limitation will be eliminated via ontologies where data is given with well-defined meanings, understandable by machines.

(c) **Scope:** Through literature survey, it has been determined that inaccessible part of the web is about five hundred times larger than accessible one [53]. It is estimated that there are billion pages of information available on the web, and only a few of them can be reached via traditional search engines. In semantic web formal semantics of data are available via ontologies, and the ontologies are the essential component of semantic web accessible to semantic search engines.

(d) **Environment:** Semantic web is the web of ontologies having data with formal meanings. This is in contrast to current web which contains virtually boundless information in the form of documents. The semantic web, on the other hand, is about having data as well as
Table 2.2: Semantic web vs. current web

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Factors</th>
<th>(Non-Semantic) Web</th>
<th>Semantic Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Conceptual Perception</td>
<td>large hyperlinked book</td>
<td>Large interlinked database</td>
</tr>
<tr>
<td>2.</td>
<td>Content</td>
<td>No formal meanings</td>
<td>Formally defined</td>
</tr>
<tr>
<td>3.</td>
<td>Scope</td>
<td>Limited – Probably invisible web excluded</td>
<td>Boundless – Probably invisible web included</td>
</tr>
<tr>
<td>4.</td>
<td>Environment</td>
<td>Web of documents</td>
<td>Web of ontologies, data and documents</td>
</tr>
<tr>
<td>5.</td>
<td>Resources Utilization</td>
<td>Minimum-Normal</td>
<td>Maximum</td>
</tr>
<tr>
<td>6.</td>
<td>Inference/Reasoning capability</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>7.</td>
<td>Knowledge Management applications sport</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>8.</td>
<td>Information searching, accessing, extracting,</td>
<td>Difficult and time-consuming task</td>
<td>Easy and Efficient</td>
</tr>
<tr>
<td>9.</td>
<td>Timeliness, accuracy, transparency of information</td>
<td>Less</td>
<td>More</td>
</tr>
<tr>
<td>10.</td>
<td>Semantic heterogeneity</td>
<td>More</td>
<td>Less</td>
</tr>
<tr>
<td>11.</td>
<td>Ingredients</td>
<td>Content - Presentation</td>
<td>Content, presentation &amp; Formal Semantics</td>
</tr>
<tr>
<td>12.</td>
<td>Text simplification and clarification</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

documents that machines can process, transform, assemble, and even act on data in useful ways.

(e) **Resources Utilization:** There are a lot of web resources that may be very useful in our everyday activities. In current web it is difficult to locate them; because they are not annotated properly by the metadata understandable by machines. In semantic web there will be a network of related resources. It will be very easy to locate and to use them in semantic web world. Similarly there are some other criteria factors for comparison between current web and semantic web. For example, information searching, accessing, extracting,
interpreting and processing on semantic web will be more easy and efficient; Semantic web will have inference or reasoning capability; network or communication cost will be reduced in the presence of semantic web for the reason of relevant results; and many more - some are listed in the Table 2.2.

According to theme of semantic web, if the explicit semantics of web resources are put together with their linguistic semantics and are represented in some logic-based languages then we can handle the limitations of current web. To support this theme of semantic web, W3C recommended some standards [88] such as RDF (Resource Description Framework), RDFS (RDF Schema), OWL (Web Ontology Language), SPARQL (a query language for RDF) and GRDDL (Gleaning Resource Descriptions from Dialects of Languages). RDF is used for data model of semantic web application. Resources are represented through URIs, connected through labeled edges which are also represented through URIs. RDF is represented through a language called RDFS. There is another more powerful language so-called OWL to represent RDF model. The query language such as SPARQL can be used to querying RDF model. Now the semantic web vision has become a reality [89, 90]. Several semantic web systems have been developed. A subset of these systems is given in Table 2.3. A huge number of ontology based web documents have been published.

2.2 The Web Ontology

Ontology formally provides structural knowledge of a domain and its data in the machine-understandable way in some W3C [1] recommended technologies such as RDFS [87] and OWL [64] for ontology formalization. It is an essential component of any semantic web application. Basically, it is the central component in overall layered architecture of the
<table>
<thead>
<tr>
<th>SW Systems</th>
<th>County</th>
<th>Activity area</th>
<th>Application area of SWT</th>
<th>SWT used</th>
<th>SW technology benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Digital Music Archive (DMA) for NRK using SW techniques</td>
<td>Norway</td>
<td>Broadcasting</td>
<td>IS, CD, &amp; DI</td>
<td>1,2,3 &amp; IHV</td>
<td>IS, INR, S&amp;RD</td>
</tr>
<tr>
<td>A Linked Open Data Resource List Management Tool for UndergradSds</td>
<td>UK</td>
<td>ELT, and publishing</td>
<td>CD, CM, DI, and SA</td>
<td>RDF, 3, 1, SKOS &amp; PV</td>
<td>(ECR), P, reduced time to market, and S&amp;RD</td>
</tr>
<tr>
<td>A Semantic Web Content Repository for Clinical Research</td>
<td>US</td>
<td>HC and PI</td>
<td>DI</td>
<td>1, 2, 4, 5, and PV</td>
<td>automation, IM, and IS</td>
</tr>
<tr>
<td>An Intelligent Search Engine for Online Services for Public Admin.</td>
<td>Spain</td>
<td>PI and eG</td>
<td>portal and IS</td>
<td>1 and IHV</td>
<td>ECR, INR, and IS</td>
</tr>
<tr>
<td>An Ontology of Cantabria’s Cultural Heritage</td>
<td>Spain</td>
<td>PI and museum</td>
<td>portal and DI</td>
<td>1 and IHV</td>
<td>ECR and IM</td>
</tr>
<tr>
<td>Composing Safer Drug Regimens for the Individual Patient using SWT</td>
<td>US</td>
<td>HC</td>
<td>DI and IS</td>
<td>1, 2, PV, and IHV</td>
<td>S&amp;RD, open model, IS, and DCG</td>
</tr>
<tr>
<td>CRUZAR — An application of semantic matchmaking for eTourism in the city of Zaragoza</td>
<td>Spain</td>
<td>PI and eTourism</td>
<td>portal and DI</td>
<td>1, 3, Rules, PV, &amp; IHV</td>
<td>faceted navigation, P, and (S&amp;R D)</td>
</tr>
<tr>
<td>Enhancement and Integration of Corporate Social Software Using SW</td>
<td>France</td>
<td>Util and energy</td>
<td>DI, CM, and SN</td>
<td>1, 3, and PV</td>
<td>IS, S&amp;RD, and INR</td>
</tr>
<tr>
<td>Enhancing Content Search Using SW</td>
<td>US</td>
<td>IT industry</td>
<td>portal and IS</td>
<td>1</td>
<td>IS and S&amp;RD</td>
</tr>
<tr>
<td>Geographic Referencing Framework</td>
<td>UK</td>
<td>PI, GIS &amp; eG</td>
<td>DI</td>
<td>1, 2 &amp; IHV</td>
<td>S&amp;RD and automation</td>
</tr>
<tr>
<td>Improving the Reliability of Internet Search Results Using Search Thresher</td>
<td>Ireland</td>
<td>Web accessibility</td>
<td>IS</td>
<td>1</td>
<td>IS</td>
</tr>
<tr>
<td>Improving Web Search Using Metadata</td>
<td>Spain</td>
<td>Search</td>
<td>portal, IS, CD, and customization</td>
<td>1, 2, RDF, 4, PV, and IHV</td>
<td>P, open model, and S&amp;RD</td>
</tr>
<tr>
<td>KDE 4.0 Semantic Desktop Search and Tagging</td>
<td>Germany</td>
<td>Semantic desktop</td>
<td>DI, CD, IS, service integration, and SA</td>
<td>1,3,2, and RDFS++</td>
<td>IS, IM, and open model</td>
</tr>
<tr>
<td>POPS — NASA’s Expertise Location Service Powered by SW Technologies</td>
<td>US</td>
<td>PI</td>
<td>DI and SN</td>
<td>1 and 3</td>
<td>faceted navigation, S&amp;RD, and ECR</td>
</tr>
<tr>
<td>Prioritization of Biological Targets for Drug Discovery</td>
<td>Spain</td>
<td>Financial</td>
<td>CD</td>
<td>1 and IHV</td>
<td>IS, S&amp;RD, IM, and ECR</td>
</tr>
<tr>
<td>Real Time Suggestion of Related Ideas in the Financial Industry</td>
<td>US</td>
<td>life sciences</td>
<td>DI and portal</td>
<td>1, 3, 2, 2DL, PV, and IHV</td>
<td>IS, S&amp;RD, open model, and IS</td>
</tr>
<tr>
<td>S Cut Desc to improve discovery</td>
<td>UK</td>
<td>Telecom</td>
<td>CD and IS</td>
<td>1, PV &amp; IHV</td>
<td>S&amp;RD, open model, and IS</td>
</tr>
<tr>
<td>Semantic MDR and IR for National Archives</td>
<td>Korea</td>
<td>PI</td>
<td>portal, CD, and SA</td>
<td>1, 2, Rules, PV, and IHV</td>
<td>IS</td>
</tr>
<tr>
<td>Semantic Tags</td>
<td>Serbia</td>
<td>IT industry</td>
<td>SA, SN, and DI</td>
<td>1, 3 and PV</td>
<td>INR, IS, S&amp;RD, and DCG</td>
</tr>
<tr>
<td>SWT for Public Health Awareness</td>
<td>US</td>
<td>HC, PL&amp; eG</td>
<td>DI</td>
<td>1,2, PV&amp;IHV</td>
<td>S&amp;RD and IM</td>
</tr>
<tr>
<td>Semantic-based Search and Query System for the Traditional Chinese Medicine Community</td>
<td>China</td>
<td>PI and HC</td>
<td>DI, IS, and schema mapping</td>
<td>2, 3, and PV</td>
<td>S&amp;RD and IS</td>
</tr>
<tr>
<td>The SW for the Agricultural Domain, Semantic Navigation of Food, Nutrition and Agriculture Journal</td>
<td>Italy</td>
<td>PI and eG</td>
<td>Portal, IS, SA, and CD</td>
<td>1, 2, SKOS, PV, and IHV</td>
<td>IS</td>
</tr>
<tr>
<td>The swoRDFish Metadata Initiative: Better, Faster, Smarter Web Content</td>
<td>US</td>
<td>IT industry</td>
<td>portal and DI</td>
<td>1 and IHV</td>
<td>DCG and ECR</td>
</tr>
<tr>
<td>Twine</td>
<td>US</td>
<td>IT industry</td>
<td>SA, SN, and DI</td>
<td>RDF, 1, 2, 3</td>
<td>INR, IS, S&amp;RD, and DCG</td>
</tr>
<tr>
<td>Use of SWT in Natural language interface to Business Applications</td>
<td>India</td>
<td>IT industry</td>
<td>natural language interface</td>
<td>1, 2, 5, 3, and IHV</td>
<td>IM and ECR</td>
</tr>
</tbody>
</table>

**Different abbreviations and integers used in above table are:**

- **SWT** (Semantic Web Technologies), **IHV** (In-House Vocabularies), **IS** (Improved Search), **IM** (Incremental Modeling), **CD** (Content Discovery), **DI** (Data Integration), **INR** (Identify New Relationships), **S&RD** (Share and Reuse Data), **PV** (Public Vocabularies), **ECR** (Explicit Content Relationships), **SA** (Semantic Annotation), **SN** (Social Networks), **GIS** (Geographic Information System), **ELT** (Education, Learning Technology), **CM** (Content Management), **DCG** (Dynamic Content Generation), **P** (Personalization), **PI** (Public Institution), **HC** (Health Care), **eG** (eGovernment), **1** (RDFS), **2** (OWL), **3** (SPARQL), **4** (GRDDL), **5** (Rules, Rules (N3))
Figure 2.1: A layered model of semantic web (barrowed from [12]).

The semantic web, as shown in Figure 2.1. In ontologies, the information-resources are connected in such a way that each is uniquely identified by URI and new information can be drawn through a reasoning process. Basically, the ontology is a special type of network of web-resources. A conceptual view of ontology can be in RDF-graph or in triples form. Consider an example of a person’s family ontology as shown in Figure 2.2. Now by definition of ontology, it looks a special kind of a network of information resources. The only information explicitly given in said ontology are isFatherOf, isMotherOf, isBrotherOf, isSisterOf, but we can drive a several types of new information such as isSonOf(3,1) (i.e. Humza is the son of Farooq), isHusbandOF(8,9), isWifeOf(9,8), isGrandFatherOf(8,3), isGrandmotherOf(9,3) and many more can easily be derived. It can be implemented in some logic-based language such as OWL and conceptually it can be seen as triples or RDF-graph (for detail please see Appendix-I).
Figure 2.2: A sample slice of person’s family ontology
2.3 Engineering Ontologies for Semantic Web

We have reviewed the literature concerning the adaptation of existing web development methodologies for semantic web applications. The findings disclose that not much work has been done in this direction. XML web engineering methodologies adaptations are proposed in the form of WEESA [38]. It generates semantic annotations by defining a mapping between the XML schemas and existing ontologies. No proper guidelines are there for designing ontology. WEESA cannot directly design or develop ontologies. It focused on mapping of XML schema concepts onto concepts of existing ontology. In [37], the authors have proposed an extension of the Web Site Design Method. In this approach object chunk entities are mapped to concepts in the ontology. OOHDM has extended in the light of the semantic web technologies [36]. Its primitives for conceptual and navigation models have been described as DAML classes and RDFS have been used for domain vocabulary. The Hera [39] methodology has been extended for adaptive web-based application engineering. It uses semantic web standards for models representation.

Existing web engineering methodologies as mentioned above, have adapted the use of ontology in their development process, but their main focus is mapping and annotation, using existing ontologies. Design process of new ontologies during semantic web application development is remained lack of focus. No proper guidance is there on how to design ontology for semantic web applications.

There are some other methodologies for ontology development as surveyed in [48, 49, 50]. Mostly these methodologies focus on specification and formalization of ontology and do not concentrate on its design phase. These methodologies are based on natural language
processing (NLP) and machine learning techniques. Orientation of these methodologies is web-agents facilitation rather than web-contents formalization. The work on ontology development was boosted when the idea of Semantic Web was envisioned.

In KBSI IDEF5 methodology [42], data about domain is collected and analyzed. Then ‘built and fixed’ strategy is used to create ontology. Ushold and King [43] proposed an ontology development methodology. In this methodology, after identifying purpose of ontology, it is captured and then coded. In MethOntology [44] after preparing ontology specification, knowledge is acquired and analyzed to determine domain terms such as concepts, relations and properties and then formalization is started. After that, evaluation and documentation is performed.

In [45] a methodology, based on collaborative approach has been proposed. In its first phase, design criteria for the ontology, specified boundary conditions for the ontology and set of standards for evaluating ontology, are defined. In second phase, the initial version of ontology is produced, and then through iterative process the desired ontology is obtained. Software ‘Built and Fixed’ approach is followed, which leads to heavy development and maintenance cost.

Helena and João [46] proposed a methodology for ontology construction in 2004. This method divides ontology construction activities in different steps such as specification, conceptualization, formalization, implementation and maintenance. Knowledge acquisition, evaluation and documentation are performed during each phase. There are some other approaches investigating the transformation of a relational model into an ontological model. In these approaches, the ontology is developed from database schemas. These approaches
mainly use reverse engineering theory. In [47], a methodology is outlined for constructing ontology from conceptual database schemas using a mapping process. The process is carried out by taking into consideration the logical database model of target system.

Most of methodologies as overviewed above are based on the ‘built and fixed’ approach. In that, first initial version of ontology is built and improved iteratively until domain requirements are satisfied. In this way the basic principles of software engineering are not followed properly. These methodologies mainly focus on data during developing process rather than focus on descriptive knowledge. These methodologies mainly work on specification and implementation phases and design phase lacks in proper attention. Moreover the design and implementation phases of these methodologies are difficult to identify and separate.

We have proposed methodologies [75, 80] based on software engineering principles, to design web ontology separately and during semantic web application development.

2.4 Populating Web Ontologies

Web-ontology automatic population is one of the research issues nowadays. There are two stages of web-ontology developments; one is its (schema) creation and second is its population. Although populating web-ontology is not a complicated task, but it is very time consuming, enormous and laborious when it is performed manually. For the success of Semantic Web there is a need of a proper solution for populating ontologies. In current web, mostly the data is available in XML data files, and there is extremely large number of these data resources, containing data in terabytes. To upgrade such data-intensive web applications
into semantic web applications, there is need of some proper methodologies for automatic populating of ontologies.

Different approaches have been proposed for ontology populating. Some of them are based on natural language processing and machine learning techniques [63] [67]. Junte Zhang and Proscovia Olango [66] presented a novel approach for making and populating ontologies. According to this method domain knowledge about the ontology is collected and domain ontology is constructed by using open source tool called Protégé. This domain ontology is transformed to equivalent RDF file and this RDF file is manipulated manually to populate the ontology skeleton created by the Protégé. XSLT or Xquery have been used to extract the relevant information from Wikipedia pages into Perl’s regular expressions and then ontology instances are generated using those expressions. Semantic heterogeneities and inconsistency problems were raised while exporting Wikipedia pages to XML format and those remained unsolved.

Web-ontology creation and populating guidelines are provided while developing new semantic web applications using WSDM [68] [69]. Concepts in the ontology are mapped to object chunks manually at conceptual level. This conceptual mapping is used to generate actual semantic web page at implementation level. A similar approached is used in WEESA [72]. This is an adaptation of XML-base web engineering. Web-ontology concepts are mapped to schema elements of XML document. This mapping is defined for each page. Then ontology is populated via tool [71]. In [76] a methodology is proposed for extracting data from web documents for ontology population. This methodology consists of three steps. The first step consists of extracting information in the form of sentences and paragraphs. Web
documents are selected by using some search engines or we could select them manually. This information is understood by the system semantically and syntactically and it also understands the relations between the terms of text using some rhetorical structures. For efficient representation of extracted information, XML is used due to its flexibility and abilities for handling data. We proposed ontology populating methodology [73] to populate ontology from data stored in XML data files. This methodology may help in transforming an existing non-semantic web application into semantic web application by populating its web-ontology semi-automatically through a set of transformation algorithms by reducing the time consuming task of ontology population. In [77], a similar work is presented. The proposed methodologies take a web-ontology schema and XML-document as input and produce a populated ontology as output.

2.5 Integrating Web Ontologies

When multiple ontologies are simultaneously used in some integration operations such as merging, mapping and aligning then they may suffer from different types of heterogeneities such as semantic heterogeneity and non-semantic or syntactic heterogeneity [2, 3].

The syntactic heterogeneity is due to the use of different languages for the formalization of ontologies e.g., one ontology is formalized in OWL [64] and the other is formalized in DAML [65] language. The semantic heterogeneity consists of terminological, conceptual and contextual heterogeneities. The terminological heterogeneity occurs when different terms are used to represent same concepts or same term is used to represent different concepts. Conceptual heterogeneity between two concepts may occur due to
granularities of concepts i.e., when one is sub-concept or super-concept of the other or both are overlapped. Similarly, two concepts are explicit-semantic heterogeneous when they have different roles or functionalities in the similar domain.

In ontology model, the taxonomic and non-taxonomic relations of a concept represent its roles and super/sub concepts respectively. Since, in a certain domain an intellectual concept is explicitly defined in term of roles that it keeps, therefore we may call the roles of a concept as its explicit semantics. However, explicit semantics of a non-intellectual concept may be derived from its granularities and its physical attributes because of typical nature of those domains, e.g. in ontology of a furniture domain, concepts like chair, table and desk have no intellectual properties, these concepts have only taxonomic (i.e. parent, child, sibling) and elementary (i.e. color, type, etc.) characteristics.

Nowadays, the similarity based on explicit semantics of concepts has become vital in order to realize the vision of semantic web but a very little attention has been given on this dimension of concepts while measuring their similarity. In [28], the background knowledge of domain has been used via ontology to determine similarity between concepts of two ontologies, especially for those concepts which are not lexically and structurally similar. A similar work was presented in [29], and it has been evaluated by matching a medical ontology to another while using comprehensive medical domain ontology as background knowledge. The key consideration of this technique is the consultation of third ontology i.e. the comprehensive domain ontology for handling the missing non-taxonomic or logical relations between concepts of input ontologies. This technique is well suited for those ontologies having very poor taxonomic and non-taxonomic relations between concepts.
Several techniques have been proposed for measuring similarity between concepts of ontologies for aligning them [52-54]. On the basis of similarity-measuring criteria, these techniques are mostly categorized into schema-based and instance-based techniques. In schema-based techniques, similarity between concepts is measured at structure-level while ignoring their actual data, whereas in instance-level techniques, similarity is measured by taking actual data into consideration. In structural aligning, the taxonomic characteristics of concepts are mostly considered. The two concepts are rendered taxonomically similar [5, 9, 27, 62] if (i) their direct super-concepts are similar; (ii) their sibling-concepts are similar; (iii) their direct sub-concepts are similar; (iv) their descendant-concepts are similar; (v) their leaf-concepts are similar and vi) concepts, in the paths from the root to those concepts, are similar. Irrespective of the structural aligning technique used, we observed [81], that certain pairs of similar concepts are categorized dissimilar because of bias of above mentioned criteria towards those concepts whose siblings-concepts, sub-concepts or direct super-concepts are not similar.

There are some other approaches for measuring semantic similarities between concepts of XML schemas, database schemas and some graph-like structures [54-60]. In these schemas, the explicit meanings of concepts are determined either from their respective attributes or from their hierarchical positions. The meanings of concepts in terms of their interactions with other concepts are not explicitly defined in these schemas. Therefore these approaches seem to be inappropriate for measuring the similarities between concepts of ontologies schemas.
Alasoud et al., proposed technique for ontology matching [52]. It has three phases. It uses Levenshtein Distance [20] and WordNet [22] techniques in first phase. A matrix with binary values, is the output of first phase e.g. if there are three concepts in A ontology and 4 concepts in B ontology then the size of matrix is 3 X 4., if \( r_{ij} = 1 \) means that \( a_i \) and \( b_j \) are similar, whereas \( r_{ij} = 0 \) indicates that \( a_i, b_j \) are not similar. For \( a_i \) there may be multiple corresponding \( b_j \)s that are similar and only one among them is short listed on the bases of high score computed in the third phase from the neighbors of those concepts search in the second phase. The algorithm does not properly differentiate between concepts and its data properties and object properties. Data types and constraints are ignored while measuring similarity between data properties. Due to use of WordNet, Levenshtein Distance formula of degree of similarity, the completeness and the correctness of the result is comparative low.

Secondly, the pairs of similar concepts are not accompanied by their semantic relations. Whole-part relationships are only considered while the role-based and taxonomic characteristics are overlooked.

In order to detect and retrieve relevant ontologies Alexander Maedche, and Steffen Staab [61], proposed a set of similarity measures for ontologies. The lexical and conceptual aspects of concepts of ontologies are considered. In lexical level measures, the terms used to name concepts are compared and their similarity is computed using well known method known as edit distance [20] and they proposed a lexical metric for similarity computing which is equal to \( \text{MAX} (0, \text{MIN} (|L_i|, |L_j|) - \text{ed} (L_i, L_j) / \text{MIN} (|L_i|, |L_j|)) \), where \( L_i \) and \( L_j \) are two lexical entities whose similarity is being computed. The metric value varies in between 0 and 1, where 0 means both are dissimilar and 1 means both terms are exactly same. The \( \text{ed} \) is a function that returns an integer which is equal to number of insertions, deletions or
substitutions to transform one lexical term into other. At conceptual level, the similarity is computed from the similarities of their respective super-entities. Two entities are similar if their direct super-entities in their respective taxonomies are similar or all super-entities of first entity are similar to super-entities of second entity used in comparison.

Trojahn et al., [30] proposed a technique for composite ontology mapping. Different existing matchers have been collectively used in this technique. The technique has been automated through agent-based scenarios. For lexical similarity measuring, they use the string-based measures and to examine the linguistic semantics of terms, the WordNet has been used. The structural similarity between two terms has been computed based on the similarities of their respective super and sub concepts. The overall degree of similarity has been computed from the lexical, linguistic and structure similarities of terms.

In [34], syntactic and semantic matchers are used to compute similarity, and final decision is made by the user. The syntactic matcher uses string-based techniques, known as edit distance and n-gram to measure the degree of similarity between two input terms. For semantic comparison, a thesaurus is searched for synonyms of input terms and again the comparison is made. During semantic matching, the depth of concepts from their common super-concepts in their respective taxonomies, are also considered. The overall degree of similarity is computed from the results of syntactic matcher and semantic matcher.

In HCOME [15] approach of ontology merging, the ontology is defined as (i) a set of terms used to represent concepts, their relationships and data-properties; (ii) and the axioms for interpretation of terms. Using WordNet and semantic index method, the highly ranked sense of each term is located and identified. For each term, all generic and specific terms are
also retrieved from wordNet and then semantic relation between two terms, based on this information is identified. Finally, the merging decision based on the semantic relation, is made.

There are some other ontology merging, mapping and align tools [16-19, 23-27]. Each of them uses different matching technique to measure the similarity between concepts of ontologies. These toots use string-based techniques such as edit distance and n-gram to measure the degree of similarity between terms used for representing concepts. Some of them use WordNet [22] to get linguistic information such as synonyms and hyponyms while measuring similarity and then the structural information of terms are further used to compute the overall their degree of similarity. We surveyed briefly just to examine whether they used the domain specific vocabulary while measuring the primary similarity and then their non-taxonomic interactions, so-called role-based interaction especially for intellectual concepts of ontologies.

From Table 2.4, we have observed that some approaches consider the synonyms provided by the WordNet [22] but their main consideration are the terms or the synonyms of terms rather than concepts represented by them and secondly, most of the tools consider the taxonomic characteristics of concept i.e., their relations with parents and children but none of them have focused on their non-taxonomic relations..

The string-based techniques, such as (i) edit-distance (ii) prefix (iii) suffix and (iv) n-gram, are not well suited for identifying similarity between concepts of web-ontologies because they may produce inaccurate results in various scenarios. These techniques compare
Table 2.4: A comparison of some techniques for similarity measurement between ontologies

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Name-based similarity</th>
<th>Linguistic-based similarity</th>
<th>Taxonomic-based similarity</th>
<th>NonTax.-based similarity</th>
<th>DOS</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAOM [34]</td>
<td>Edit-distance, n-gram</td>
<td>Thesaurus</td>
<td>Parents</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>MSBO [61]</td>
<td>Edit-distance</td>
<td>N</td>
<td>Parents</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>SEMC[19]</td>
<td>Edit-distance</td>
<td>WordNet</td>
<td>Parents, Children</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>LOM [23]</td>
<td>Edit-distance</td>
<td>WordNet</td>
<td>Parents</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>ASCO [16]</td>
<td>String-based</td>
<td>N</td>
<td>Parents, Children</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Chimaera [17]</td>
<td>Edit-distance</td>
<td>N</td>
<td>Parents, Children</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>OntoMap [18]</td>
<td>Edit-distance</td>
<td>N</td>
<td>Parents, Children</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>SSMO [26]</td>
<td>String-based</td>
<td>N</td>
<td>Parents, Children</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>SAMBO [27]</td>
<td>Edit-distance</td>
<td>N</td>
<td>Parents, Children</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>EOMT [52]</td>
<td>Edit-distance</td>
<td>WordNet</td>
<td>Parents</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>SM [55]</td>
<td>String-based</td>
<td>WordNet</td>
<td>Parents</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Cupid [35]</td>
<td>Edit-distance</td>
<td>DV</td>
<td>Parents, Children</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>
terms used for concepts rather than concepts themselves. Edit distance [20] is mostly used for measuring the similarity between two terms. In this edit distance method, the similarity is measured based on the number of insertions, deletions and the substitutions to transform one term into other. The degree of similarity (DoS) between two concepts based on their terms can be measured using a metric, as given below:

\[
\text{DoS} = \max(0, \frac{\min(\text{LengthOf}(a), \text{LengthOf}(b)) - \text{IDS}(a,b)}{\min(\text{LengthOf}(a), \text{LengthOf}(b))}) \in [0,1]
\] (2.1)

In Equation 2.1, the \(\text{IDS}\) is a function that returns integer-value, equal to number of insertion, deletion or substitutions to transform term \(a\) into \(b\) or vice versa. In some scenarios, the above metric produces incorrect results e.g. the degree of similarity between terms \(\text{Deptt}\) and \(\text{Department}\) of respective ontologies \(A\) and \(B\) is 0.25 i.e. these are partially similar with respect to this metric, but both the terms represent the same concept. Similar are the cases of other techniques ii, iii and iv as list above. The WordNet [22] has been recommended to check the semantics of different terms, but it does not provide a domain-specific scene of certain terms. Secondly, these techniques measure similarities in terms of similarity coefficient instead of semantic relation that may exist between a pair of concepts.

Individually, the degree of similarity does not mark a concept which one is more generic or more specific than the other in a pair of similar concepts and similarly the semantic relation individually does not reflect the level of generality. The \(\text{DoS}\) and the semantic relation (SR) between concepts are represented by the expressions as given below:

\[
\text{DoS} \in [0, 1]
\] (2.2)
SR ∈ \{=, \neq, \subseteq, \supseteq, X\} \quad (2.3)

In Equation 2.3, the values =, \neq, \subseteq, \supseteq and X refer to equivalent, not equivalent, more general, more specific, and unknown relations respectively. The DoS, as shown in Equation 2.2, varies between 0 and 1. The 0 means both concepts are entirely dissimilar, while 1 means they are exactly same. From DoS it is difficult to say which one concept is more generic or more specific? Similarly, \subseteq and \supseteq semantic relations between two concepts do not specify as to how much one concept is more generic or how much one concept is more specific than the other one.

The structural similarity measurement criteria [5, 9, 27, 62], as discussed before, declare certain pairs of similar concepts as dissimilar because of the biasness of these criteria towards those concepts whose siblings-concepts, sub-concepts or direct super-concepts are not similar. For example, in Figure 2.5, although the pairs (A:Department, B:Department), (A:Conference, B:Conference) and (A:FullProfessor, B:FullProfessor) are similar but these are declared dissimilar because of violation of above criteria. Similarly, the certain pairs of

![Figure 2.3: A slice of taxonomy of furniture ontology-A](image_url)
dissimilar concepts are categorized as similar because of satisfying the above criteria. For example, the pair \((A:Table, B:Chair)\) as shown in Figures 2.3 & 2.4, is declared similar because of the satisfaction of said criteria although this pair of concepts is a dissimilar pair.

With respect to ontology, a concept is defined as a class of objects or individuals with some common elementary, taxonomic and non-taxonomic characteristics. A concept has a certain
name and some synonyms. Usually, it is known by its taxonomic characteristics (i.e., parents, children and siblings), and the roles (i.e., non-taxonomic characteristics) it keeps, in a certain domain in addition to its name and synonyms. The similarity between intellectual concepts is usually based on their functionalities or roles instead of their terminological or taxonomic characteristics but a very little attention has been given on this dimension of concepts while measuring their degree of similarity and semantic relations. Since an intellectual concept is explicitly defined and known by the roles it keeps instead of by its parents, children or siblings. Some concepts may have no intellectual properties e.g. in ontology of a furniture domain, concepts like chair, table and desk have only taxonomic (i.e. parent, child, sibling) and elementary (i.e. color, type, etc.) characteristics. For such situation the granularities of concepts should be used for measuring their DoS and SR whereas the roles should be used in place of granularities for measuring DoS and SR between intellectual concepts.

We have proposed a semi-automatic technique [74], for measuring the similarity between concepts of two ontologies based on their explicit meanings in addition to their super-concepts. The proposed technique is based on layered strategy. In first layer, the primary similarity is identified via domain vocabulary. The contextual similarity of primarily similar concepts, identified in first layer, is then measured in the second layer. In the third layer, the similarity based on explicit semantics is identified and measured. The proposed technique determines all candidate pairs of similar concepts without omitting any candidate similar-pair. The degree of similarity, based on explicit meanings of concepts is measured which makes the technique more reliable and well-suited for vision of semantic web. The technique works at conceptual-level of ontologies and this technique may be used in different operations of ontology such as merging, mapping, integrating in addition to aligning operation.
Chapter 3

PROPOSED TECHNIQUE FOR SIMILARITY IDENTIFICATION AND MEASUREMENT

In previous chapter, we have reviewed different ontology matching techniques to examine their matching criteria and strategies used for similarity measurement. Based on rigorous analysis, we propose a technique for similarity identification and measurement between ontologies in the context of semantic web. Each phase of proposed technique is formally defined in algorithmic form. A feature-based comparison between proposed technique and existing techniques is also made at the end of the chapter.

3.1 The Basis of Proposed Technique

- The comparison of concepts rather than terms used for them
- The use of domain-specific semantics of concepts rather than their linguistic semantics
- The super-concepts based contextual similarity measurement while relaxing the similarities between their respective sub or sibling concepts
- The explicit semantic based measurement of degree of similarity (DoS) and semantic relation (SR)
- The add-on of both the DoS and SR with each pair of similar concepts
- The use of layered matching strategy

3.1.1 Comparison of Concepts Rather Than Terms

A concept is represented by a set of terms, including its synonyms such as shown in Figure 3.1. The existing techniques, as summarized in Table 2.4, use edit-distance based formula
to compute **DoS** between two concepts. In the edit-distance based **DoS**, the terms are used in the measurement process of **DoS** rather than concepts represented by those terms.

![Figure 3.1: Concept vs. terms](image1)

![Figure 3.2: The use of concepts rather than terms in measurement of **DoS**](image2)

![Figure 3.3: The **DoS** through edit-distance based formula and through proposed formula](image3)

In some scenarios, the edit-distance based **DoS** produces incorrect results e.g. according to edit-distance based **DoS**, the **dept** and **department** (from ontologies **o1** and **o2**) are
dissimilar but actually, these terms represent the same concept. Similarly, some pairs of
dissimilar concepts are declared as similar pairs such as (Software Design, Software
Designer), (System Analyst, System Analysis) because the edit-distance based DoS between
concepts of these pairs are 0.86 and, 0.85 respectively. Also, these techniques measure
similarity in terms of similarity coefficient and do not measure SR that may exist between
a pair of concepts.

In proposed technique, the measurement of DoS is performed on the concepts themselves
represented by the given terms in ontologies. The measurement process of DoS is
accomplished through domain vocabulary (DV), as shown in Figures 3.2 and 3.3. In
proposed technique we have referred to it as primary similarity, as discussed in the coming
section. The DoS and SR are computed from the granularities of concepts.

3.1.2 Use of Domain-specific Semantics Rather Than Linguistic Semantics

In linguistic-semantic based matching, the concepts and their respective synonyms are
examined. That is, if one concept is a synonym of other concept or vice versa, then both
concepts are considered as equivalent concepts. The current techniques use WordNet [22] to
fetch the synonyms of concepts. However a domain may have some abbreviated, acronyms
or composite named concepts which are not found in WordNet. In proposed technique we
use domain specific vocabulary in place of WordNet to get better results of linguistic
semantic matching.

3.1.3 Super-Concepts Based Contextual Similarity Measurement

The context of a concept is usually known by its Super, Sub and Sibling (3S) concepts in the
respective ontology. Usually, a concept may or may not have sub or sibling concepts but it
always has some parents. This means that while identifying contextual similarity between
two concepts, the similarity between their respective super concepts should be considered only. We have empirically observed that while measuring contextual similarity between two concepts, if the similarities of 3S concepts are taken into consideration then some pairs of similar concepts may be declared dissimilar. This is because of dissimilarity of their respective sub concepts or sibling concepts. Furthermore, while measuring contextual similarity between two concepts, the similarity between their respective immediate super-concepts is not mandatory. In proposed technique, we have taken into consideration the similarity of their super-concepts while relaxing the similarities of sub and sibling concepts.

3.1.4 Explicit Semantic Based Measurement of Degree of Similarity (DoS) and Semantic Relation (SR)

The explicit semantic similarity measurement is the key basis of proposed technique. According to theme of Semantic Web, the issues of current web can be overcome by formalizing explicit semantics of web-contents via ontologies. However, the ontologies may themselves suffer from explicit semantic heterogeneity when their lexically and contextually similar concepts have different or overlapped explicit semantics. In order to resolve such type of heterogeneity, the similarity measurement based on explicit-semantics, has become essential.

3.1.5 The Add-On of both the Dos and SR with Each Pair of Similar Concepts

The proposed technique compute both the DoS and SR between concepts, As mentioned earlier, the value of DoS between two concepts remains in the range of 0 and 1. The DoS is inadequate to determine which concept is more generic or more specific than the other concept? Similarly, the semantic relations such as $\supseteq$ and $\subseteq$ between two similar concepts.
show that one concept is more generic or more specific to the other concept. However, it does not reflect the DoS between the two concepts. Therefore, each pair of similar concepts should be accompanied with both DoS and SR in order to take better decision while aligning, merging and mapping operations of the ontologies.

### 3.1.6 Matching Strategy

We have empirically observed that within a certain domain, the lexically dissimilar concepts are always contextually dissimilar. Similarly the contextually dissimilar concepts are always explicit semantically dissimilar. That is, there is no need to measure the contextual similarity between lexically dissimilar concepts. And, there is no need to measure the explicit-semantics similarity between contextually dissimilar concepts. Secondly, the direct measurement of contextual similarity without measuring the lexical similarity may produce

![Diagram](image)

Figure 3.4: Similarity measurement strategy used in proposed technique
inaccurate result. This suggests that, if the similarity measurement is performed in some integrated and layered fashion as shown in Figure 3.4, the measurement process becomes more efficient. The DoS measurements between a pair of concepts, through existing techniques and through proposed technique, are shown on Figures 3.5 and 3.6 respectively. Most of the existing techniques follow the individual matching. The individual matching strategy reduces the efficiency of overall similarity computing process because of the maximum input for all matchers as shown in Figure 3.7. For example, there are $N$ numbers of candidate pairs whose similarities are to be measured. In individual matching strategy, as shown in Figure 3.4, each matcher gets same and the maximum input i.e. $N$, whereas, in integrated and layered strategy, as shown in Figure 3.4, the input of second and third matchers are $N_1$ (where $N_1 < N$) and $N_2$ (where $N_2 < N_1$) number of pairs respectively. That is, the input of 2\textsuperscript{nd} Matcher of proposed
Figure 3.6: Degree of similarity measurement through proposed technique along with new criterion (role-based)

The technique is less than the input to the second matcher of existing techniques and same is the case with third matchers of proposed and existing techniques. Furthermore, the 1st level matcher used in proposed technique, identifies similarity between input terms, based on the actual concepts represented by those terms whereas the lexical matcher, used in existing techniques, measures similarity through string-based approaches. The logical comparison between different matchers of exiting techniques and proposed techniques is given in Tables 3.1, 3.2 and 3.3 respectively.

Figure 3.7: Similarity measurement strategy used in current techniques
Table 3.1: Lexical & linguistic similarity matchers vs 1st level matcher

<table>
<thead>
<tr>
<th>Lexical Matcher &amp; Linguistic Matcher (M₁ &amp; M₂)</th>
<th>1st Level Matcher (M¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Lexical matcher measures similarity between terms by using string-based techniques.</td>
<td>• It measures similarity between concepts through customized domain-specific vocabulary, including synonyms of concepts.</td>
</tr>
<tr>
<td>• Linguistic matcher measures similarity between terms and linguistic semantics of terms i.e., synonyms of the terms.</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: Contextual similarity matcher vs 2nd level matcher

<table>
<thead>
<tr>
<th>Contextual Matcher (M₃, M₄ &amp; M₅)</th>
<th>2nd Level Matcher (M²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• It measures similarity between two concepts based on the similarity of their super concepts and the similarity of sub-concepts and the similarity of their sibling-concepts</td>
<td>• Measure similarity between two concepts based on the similarity of their super-concepts; whereas, the similarity of sub-concepts and sibling concepts have been relaxed while making contextual similarity judgment, because the sub and sibling are optional.</td>
</tr>
</tbody>
</table>

Table 3.3: Third level or explicit semantic matcher

<table>
<thead>
<tr>
<th>Existing Matcher</th>
<th>3rd Level Matcher (M³)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Not Found</td>
<td>• It measures similarity between two concepts based on their explicit semantics</td>
</tr>
</tbody>
</table>

In this chapter, for identifying similarity between concepts of two ontologies, we propose an integrated technique based on the parameters as described above. The proposed technique
has three phases as shown in Figure 3.8: (a) IPS - Identifying Primary Similarity (b) ICS - Identifying Contextual Similarity (c) IRS - Identifying Role-based Similarity. There are some prerequisite tasks such as: (i) acquisition of concepts, (ii) acquisition of super-concepts of primarily similar concepts and (ii) acquisition of roles of contextually similar concepts are the prerequisites of these three phases respectively.

3.2 Definitions and Structure Diagram

*Concept:* With respect to ontology, a concept is defined as a class of objects or individuals with some common elementary, taxonomic and non-taxonomic characteristics. A concept has a certain name with some synonyms. At schema-level of ontology a concept is formally defined as a 5-tuple i.e. \(<T, P, C, S, R>\); the T, P, C, S and R are formally defined below:

\[
T = \{\text{term}_i \mid 1 \leq i \leq N_1\} \\
P = \{\text{parent}_i \mid 1 \leq i \leq N_2\} \\
C = \{\text{child}_i \mid 1 \leq i \leq N_3\} \\
S = \{\text{Sibling}_i \mid 1 \leq i \leq N_4\} \\
R = \{\text{role}_i \mid 1 \leq i \leq N_5\}
\]

(3.1) (3.2) (3.3) (3.4) (3.5)

*Explicit Semantic:* A concept has two types of semantics, linguistic and explicit semantic. The name and synonyms of a concept represents its linguistic semantic. The linguistic semantic may also be called an implicit or built-in semantic whereas the explicit semantic of a concept is defined in terms of its roles or responsibilities that it keeps in a certain domain. That is, the explicit semantic are domain-dependent. Let the concept \(C\) has certain roles \(r_1, r_2, \ldots, r_n\) in a domain \(D\), and then the explicit semantic of \(C\) is formally defined as:

\[
\text{ES}_{C^D} = \{r_1, r_2, \ldots, r_n\}
\]

(3.6)
Primary Similarity: We may call it the 1st level similarity. Two concepts are primarily similar if and only if either (i) their names belong to T (as defined in Eq. 3.1) of concept $C_i$ where $1 \leq i \leq N$; or first-name belongs to T and second-name belongs to P (as defined in Eq. 3.2) or versa where T and P both belong to same $C_i$. The primary similarity, denoted as $\approx_1$, can be formally written as below:

\[ a \approx_1 b \text{ Iff } ((a \land b) \in T(c_i) \lor ((a \in T(c_i) \land b \in P(c_i))) \lor (b \in T(c_i) \land a \in P(c_i))) \]  

(3.7)

In Equation 3.7, $T(c_i)$ and $P(c_i)$ are sets of terms of $C_i$ concept of target domain, whereas, concepts $a$ and $b$ belong to ontologies A and B respectively.

Contextual Similarity: We may call it the 2nd level similarity. Two concepts are contextually similar if and only if they possess primary similarity and have one or more common concepts in their respective list of super-concepts. It can be formally written as:

\[ a \approx_2 b \text{ Iff } ((\text{condition given in Eq. (3.7) is true}) \land (P_a \cap P_b) \neq \emptyset) \]  

(3.8)

In Equation 3.8, $P_a$ and $P_b$ are respective sets of parents as defined in Equation 3.2, of concepts $a$ and $b$, whereas, concepts $a$ and $b$ respectively belong to ontologies A and B.

Role-based similarity: We may call it the 3rd level of similarity or explicit-semantics based similarity and it is especially used for identifying similarity between intellectual concepts. Two concepts possess role-based similarity if and only if they possess contextual similarity and they have one or more common roles in their respective list of roles.

\[ a \approx_3 b \text{ Iff } ((\text{condition given in Eq. (3.8) is true}) \land (R_a \cap R_b \neq \emptyset)) \]  

(3.9)
In Equation 3.9, $R_a$ and $R_b$ are respective sets of roles (as defined in Equation 3.5), of concepts $a$ and $b$, whereas, concepts $a$ and $b$ respectively belong to ontologies $A$ and $B$.

**Role-based Degree of Similarity (DoS):** Since there may be multiple roles of same concept therefore while identifying similarity, common roles of $a$ and $b$ are counted and then the $DoS$ of $a$ with respect to $b$ is computed by dividing total number of common roles by total number of roles in their union. Let $M$ is a set of roles of concept $a$ and $N$ is a set of roles of concept $b$ of ontologies $A$ and $B$ respectively, then $DoS$ between concepts $a$ and $b$ may be computed by using the following imperial formula:

$$DoS = \frac{|M \cap N|}{|M \cup N|} \quad (3.10)$$

**Role-based Semantic Relation:** Let $M$ is a set of roles of concept $a$ and $N$ is a set of roles of concept $b$ of ontologies $A$ and $B$ respectively, then semantic relation ($SR$) between pairs of similar concepts $a$ and $b$ may be represented as:

$$(a, b) = SR \quad (3.11)$$

The criteria for computing $SR$ are listed below:

i. $SR = '='; a$ is equivalent to $b$; (=)

$$a = b \iff (|M \cap N| = |M|) \land (|M| - |N| = 0)$$

ii. $SR = '\geq'; a$ is more generic than $b$; ($\geq$)

$$a \geq b \iff (|M \cap N| = |N|)$$

iii. $SR = '\leq'; a$ is less generic than $b$; ($\leq$)

$$a \leq b \iff (|M \cap N| = |M|)$$

iv. $SR = 'x';$ otherwise semantic relation is undefined; ($x$),

*take manual decision*
There may be no \( b_j \) exactly similar to \( a_i \), or there may be multiple \( b_j \)s that are more specific than \( a_i \) or multiple \( b_j \)s that are more generic than \( b_j \)s. In these cases, we have opted two strategies i.e. up-word and down-word strategies. In up-word strategy, we choose a pair \((a_i, b_j)\) with \( SR \) such that \( b_j \) is least granular in all \( b_j \)s. Similarly in down-word strategy we choose a pair with \( b_j \) having the maximum granularity.

**Granularity-based Degree of Similarity (DoS):** Granularity of concept \((Gc)\) is proportional to its level of generality. The generality of a concept may vary from the most generic to least generic or vice versa. Let \( g \) be the generality of concept \( c \) and let \( k \) be the constant of proportionality; then \( Gc \) can be formally defined as:

\[
Gc = k \times g
\]  

In Equation 3.12, \( g \) may vary between 1 and \( n \); where \( n \) is an integer-value. If \( g \) is equal to 1, then the concept is considered to be the most generic concept and if \( g \) is equal to \( n \), then the concept is considered to be the least generic. We compute the \( DoS \), between concepts particularly the non-intellectual concepts from their granularities. Let \( Ga \) and \( Gb \) be the granularities of two primarily similar concepts \( a \) and \( b \) respectively, then their \( DoS \) may be computed by using the empirical formula as given below:

\[
DoS = \frac{|Ga - Gb|}{Max(Ga, Gb)}
\]

**Granularity-based Semantic Relation:** Let \( Ga \) and \( Gb \) be the granularities of primarily similar concepts \( a \) and \( b \) respectively, then \( SR \) between them can be represented as:
$$\text{(a, b)} = \text{SR} \quad (3.14)$$

The criteria for computing SR are listed below:

(i) \(\text{SR} = '=' \); \(a\) is equivalent to \(b\); (\(=\))

\[
a = b \iff (G_a = G_b)
\]

(ii) \(\text{SR} = '\geq ' \); \(a\) is more generic than \(b\); (\(\geq\))

\[
a \geq b \iff (G_a < G_b)
\]

(iii) \(\text{SR} = \text{a is less generic than b; } (\leq)\)

\[
a \leq b \iff (G_a > G_b)
\]

Figure 3.8: Structure diagram of proposed technique

In structure diagram of proposed technique as shown in Figure 3.8, \(M\) and \(N\) are two RDF models of input ontologies \(A\) and \(B\) respectively. The labels \(CA, PA\) and \(RA\) represent concepts acquisition, parent acquisition and roles acquisition processes whereas the labels \(IPS, ICS\) and \(IRS\) represent identifying primary similarity, identifying contextual similarity
and identifying role-based similarity processes respectively. The label 1 represents two separate list of concepts acquired from models $M$ and $N$ respectively. The label 2 represents a list of pairs of primarily similar concepts. The label 3 represents two separate lists of parents of primarily similar concepts whereas the label 4 represents a list pairs of concepts possessing contextual similarity. The label 5 represents two separate lists of roles of contextually similar concepts whereas the label 6 represents a list of pairs of concepts possessing role-based similarity. The $O_1$, $O_2$ and $O_3$ are the vectors containing pairs of primarily, contextually and explicit semantically concepts respectively.

3.3 IPS - Identifying Primary Similarity

There are some preprocessing activities for identifying primary similarity between two concepts such as the design and population of target domain vocabulary and acquiring of concepts from input ontologies.

We have used the domain vocabulary to contain information about concepts such as terms used for their respective names, abbreviated-names and synonyms of concepts. Let $DV$ be the domain-specific vocabulary of input ontologies $A$ and $B$ whose similarity is to be determined. Each element of $DV$ has four components: (i) $Id$; (ii) name (term that is exactly the same spelled as concept); (iii) $SN$ - the synonyms of that concept; (iv) gLevel - granularity level.
While populating synonyms of a concept, the WordNet [22] can be used as helping aid. Concepts are separately extracted by traversing the triples of input ontologies. The working of acquiring concepts is given algorithmically in Figure 3.9. Conets of input ontologies \( A \) and \( B \) are taken into two vectors \( CS_A \) and \( CS_B \) as mathematically represented in Equations 3.15 and 3.16 respectively.

\[
CS_A = \{a_i | \forall \ a_i \in A; \ 1 \leq i \leq M\} \quad (3.15)
\]

\[
CS_B = \{b_j | \forall \ b_j \in B; \ 1 \leq j \leq N\} \quad (3.16)
\]

The primary similarity as defined prior, is not the same as terminological similarity as reported in literature because we focus on concepts rather than terms used to represent them.
Input: \( CS_A \) and \( CS_B \) Vectors (as defined in Eqs. 3.15 & 3.16),
(DV-Domain Vocabulary, an implicit input)

Output: \( \text{Sim}_{PS} \) (as defined in Eq. 3.17);
a vector containing pairs of primarily similar concepts

Begin
For each \( a \) in \( CS_A \)
    For each \( b \) in \( CS_B \)
        \( aId = \text{DV}.\text{getId}(a) \)
        \( Ga = \text{DV}.\text{getGranularity}(a) \)
        \( bId = \text{DV}.\text{getId}(b) \)
        \( Gb = \text{DV}.\text{getGranularity}(b) \)
        \( S1 = aId.\text{size()} ; S2 = bId.\text{size} \)
        If \( S1 = S2 \) then
            If \( aId.\text{equal}(bId) \) then
                \( \text{Temp.SR} = '=' \)
            Else
                \( \text{Temp.SR} = 'x' \)
            End if
        Else if \( S1 < S2 \) Then
            \( T = bId.\text{substr}(1,s1) \)
            If \( T.\text{equal}(aId) \) then
                \( \text{Temp.SR} = '⊇' \)
            Else
                \( \text{Temp.SR} = 'x' \)
            End if
        Else if \( S1 > S2 \) Then
            \( T = aId.\text{substr}(1,s2) \)
            If \( T.\text{equal}(bId) \) then
                \( \text{Temp.SR} = '⊆' \)
            Else
                \( \text{Temp.SR} = 'x' \)
            End if
        End if
        \( \text{Temp.DOS} = \text{absolute} (Ga - Gb) / \text{Maximum} (Ga, Gb) \)
        \( \text{Sim}_{PS}.\text{add}(\text{temp}) \)
    Next
Next
End

Figure 3.10: A slice of pseudo code for identifying primary similarity
The identifying process of primary similarity is given in algorithmic form in Figure 3.10. This phase returns a vector containing pairs of primarily similar concepts with \( DoS \) and \( SR \).

The terms used for concepts in both source ontologies \( A \) and \( B \), as obtained in vectors \( CS_A \), \( CS_B \) as defined in Equations 3.15 & 3.16 are the input and a vector \( \text{Sim}_{PS} \) as formally defined in Equation. 3.17, containing pairs of primarily similar concepts is obtained as output of this phase. In Equation 3.17, the \( \approx^1 \) represent primary or first level similarity as already defined in Equation 3.7 whereas \( DoS \) and \( SR \) are also already defined in Equations 10, 11 and 13, 14 based on roles and granularities of concepts respectively.

\[
\text{Sim}_{PS} = \{(a, b, DoS, SR) | \forall ((a \in CS_A \land b \in CS_B) \\
\land (a \approx^1 b)) \}
\tag{3.17}
\]

### 3.4 ICS - Identifying Contextual Similarity

Since the conceptual similarity between two concepts as defined prior, is based on the similarity of their respective parent concepts, therefore, here we need the parent-concepts of all those concepts which are declared primarily similar in previous phase. Therefore, for all concepts in the resultant vector \( \text{Sim}_{PS} \) obtained from Phase-1, their respective parent concepts are separately extracted in two vectors i.e. \( C^pS_A \) and \( C^pS_B \), which are formally defined as:

\[
C^pS_A = \{(a_i, (p_i, p_{i+1}, \ldots, p_k)) | \forall \ a_i, \ p_i \in A \\
\land \ p_i \text{ isParentOf } (a_i) \}
\tag{3.18}
\]

\[
C^pS_B = \{(b_j, (p_j, p_{j+1}, \ldots, p_k)) | \forall \ b_j, \ p_j \in B \\
\land \ p_j \text{ isParentOf } (b_j) \}
\tag{3.19}
\]
For only those concepts of input ontology A, short listed in previous phase, we fetch all super concepts of each of those concepts separately. There is one triple for each concept ai indicating just immediate super-concept of that concept; where 1 ≤ i ≤ n, n is the total number of concepts of A ontology which are in SimPS. We fetch that concept so-called first parent and store in a list. Then triples are searched again to get the super-concept of the first parent. We fetch and store the immediate super-concept of first parent, so-called second parent of concept ai. Same process is repeated to get the next parent of concept ai; the process is continued until we reach a parent such that the immediate parent of that concept is a root-concept, such as Thing used in OWL-ontology. The same procedure is used to get

<table>
<thead>
<tr>
<th>Input: SimPS vector; A and B ontologies</th>
<th>Output: CSpA and CSpB vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGIN</td>
<td></td>
</tr>
<tr>
<td>Cp Vector;</td>
<td></td>
</tr>
<tr>
<td>For each pair in vSim</td>
<td></td>
</tr>
<tr>
<td>X = pair.Ca</td>
<td></td>
</tr>
<tr>
<td>Do {</td>
<td></td>
</tr>
<tr>
<td>parent =getImmediateParent(X, A)</td>
<td></td>
</tr>
<tr>
<td>CP.superConcepts.add (parent)</td>
<td></td>
</tr>
<tr>
<td>x=parent</td>
<td></td>
</tr>
<tr>
<td>} while (parent &lt;&gt; ‘RootClass’)</td>
<td></td>
</tr>
<tr>
<td>CP.concept = C; temp.add (CP);</td>
<td></td>
</tr>
<tr>
<td>Next</td>
<td></td>
</tr>
<tr>
<td>Return temp</td>
<td></td>
</tr>
<tr>
<td>End Function</td>
<td></td>
</tr>
<tr>
<td>Function getSuperConcepts (A: ontology, vSim Vector) return Vector</td>
<td></td>
</tr>
<tr>
<td>Begin</td>
<td></td>
</tr>
<tr>
<td>Cp Vector;</td>
<td></td>
</tr>
<tr>
<td>For each pair in vSim</td>
<td></td>
</tr>
<tr>
<td>X = pair.Ca</td>
<td></td>
</tr>
<tr>
<td>Do {</td>
<td></td>
</tr>
<tr>
<td>parent =getImmediateParent(X, A)</td>
<td></td>
</tr>
<tr>
<td>CP.superConcepts.add (parent)</td>
<td></td>
</tr>
<tr>
<td>x=parent</td>
<td></td>
</tr>
<tr>
<td>} while (parent &lt;&gt; ‘RootClass’)</td>
<td></td>
</tr>
<tr>
<td>CP.concept = C; temp.add (CP);</td>
<td></td>
</tr>
<tr>
<td>Next</td>
<td></td>
</tr>
<tr>
<td>Return temp</td>
<td></td>
</tr>
<tr>
<td>End Function</td>
<td></td>
</tr>
<tr>
<td>Function getImmediateParent (String C, Vector TriplesV)</td>
<td></td>
</tr>
<tr>
<td>Begin For Each T in TriplesV</td>
<td></td>
</tr>
<tr>
<td>If T.subject= C and T.predicate= ‘subClassOf’ Then</td>
<td></td>
</tr>
<tr>
<td>Return (T.object)</td>
<td></td>
</tr>
<tr>
<td>Endif</td>
<td></td>
</tr>
<tr>
<td>Next</td>
<td></td>
</tr>
<tr>
<td>End Function;</td>
<td></td>
</tr>
<tr>
<td>End Main</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.11: Acquisition of parents of primarily similar concepts
parents of each concept $b_i$ of second input ontology $B$. The process of acquisition of parents in pseudo form is given in Figure 3.11. This phase takes $C^pS_A, C^pS_B$ (From Equations. 3.18 & 3.19) vectors, populated in acquisition process and $Sim_{PS}$ (from Equation. 3.17) populated in previous phase, as input and returns a set $Sim_{CS}$, formally defined in Equation 3.20, containing pairs of taxonomically similar concepts as output.

```
Algorithm: Identifying contextual similarity
Input :(i) $C^pS_A$ and $C^pS_B$ vectors
(ii) $Sim_{PS}$ vector
Output: $Sim_{CS}$ (as defined in Eq. 3.20); a vector containing pairs of taxonomically similar concepts
Begin
    For each $p$ in $Sim_{PS}$
        $parentC_a = C^pS_A.getParents(p.C_a)$
        $parentC_b = C^pS_B.getParents(p.C_b)$
        $same = isSameParent(parentC_a, parentC_b)$
        If same Then $Sim_{CS}.add(p)$
    Next
Function $isSameParent(Vector V_a, Vector V_b)$: Boolean
    {match=False
        For each $p_a$ in $V_a$
            For each $p_b$ in $V_b$
                If $p_a = p_b$ Then
                    {match= True;
                        Break ;}
                Next
        Return match
    }
End
```

Figure 3.12: A slice of pseudo code for identifying contextual similarity
\[ \text{Sim}_{CS} = \{(a, b, \text{DoS, SR}) \mid \forall ((a, b) \in \text{Sim}_{PS} \land (a \approx^2 b))\} \] (3.20)

In Equation 3.20, the \(\approx^2\) represents contextual of 2\textsuperscript{nd} level similarity as defined in Equation 3.8. Contextual similarity is based on taxonomic positions of \(a_i\) and \(b_j\). To measure this similarity, we need to measure the similarity between their respective parents. A slice of pseudo code of identifying process of contextual similarity is given in Figure 3.12.

### 3.5 IRS - Identifying Role-based Similarity

It is based on roles of concepts. In a domain, usually a concept is known by the roles it keeps, in addition to its parents, children, siblings and attributes. The non-taxonomic relations represent roles of concepts and their parts as well. If some pairs of concepts have no intellectual characteristics then they may have no roles. In that case, those pairs of concepts possess third level of similarity implicitly. Figure 3.14, depicts a slice of pseudo code of identifying process of role-based similarity. Similarly, to measure the RS we need to acquire the roles of concepts. The roles of each concept of \(A\) and \(B\) ontologies are separately acquired in two vectors i.e. \(C^{RS}_A\) and \(C^{RS}_B\), formally defined as:

\[ C^{RS}_A = \{(a_i, (r_i, r_{i+1}, \ldots, r_n)) \mid \forall a_i, r_i \in A \land r_i \text{ isRoleOf}(a_i)\} \] (3.21)

\[ C^{RS}_B = \{(b_j, (r_j, r_{j+1}, \ldots, r_n)) \mid \forall b_j, r_j \in B \land r_j \text{ isRoleOf}(b_j)\} \] (3.22)
The \( C^R_s_A \), \( C^R_s_B \) (from Equations 3.21 & 3.22) populated in role-acquisition process and Sim_{CS} (from Equation 3.20) populated in previous phase, are the input and a set Sim_{RS}, formally defined in Equation 3.23, containing pairs of similar concepts based on their roles, is output of this phase.

\[
\text{Sim}_{RS} = \{(a, b, \text{DoS, SR}) \mid \forall ((a, b) \in \text{Sim}_{CS}) \land (a \approx^3 b)\}
\]  

(3.23)

In Equation 3.23, the \( \approx^3 \) represents role-based or 3\textsuperscript{rd} level similarity as defined in Equation 3.9. In order to identify 3\textsuperscript{rd} level similarity of contextual similar concepts short listed in previous phase, we need to acquire their roles from their respective ontologies. A slice of pseudo code for acquiring roles of concepts is given in Figure 3.13. The role of a concept with respect to a specific domain, as defined prior is represented via non-taxonomic relation in ontologies; whereas a concept may belong to domain-set or range-set of a non-taxonomic relation. A concept appearing in the domain-set of a relation may have entirely different explicit meaning to a lexically same concept appearing in the range-set of some relation.

There are different options that we have taken into consideration while computing the degree of similarity between a pair of terminologically similar concepts \( a \) and \( b \) of first and second ontologies respectively and these are:

i. both \( a \) and \( b \) belong to only domains of some relations in their respective ontologies; i.e., in ontology-A, the explicit meaning of a Student concept may include its interactions with Debate-Competition and Project concepts [contestIn(Student, Debating-Competition), workOn(Student, Project)] whereas in ontology-B, the same Student concept may interact
with *Sport* and *Software-Competition* Concepts [playSport(Student, Sport), participateIn(Student, Software-Competition)].

ii. both a and b belong to only the ranges of some relations in their respective ontologies; i.e., in ontology-A, the explicit meaning of a *Project* concept may include its interactions with *Student* concept [workOn(Student, Project)] whereas in ontology-B, the same concept *Project* may interact with *Director* concept [supervise(Director, Project)]

```plaintext
Algorithm: Extracting roles of contextually similar concepts
Input: (i) SimCS vector  
       (ii) A and B ontologies
Output: C^R_S_A, C^R_S_B Vector
BEGIN
    C^R_S_A = getRoles(A, SimCS)
    C^R_S_B = getRoles(B, SimCS)
End sub
Function getRoles(ont: ontology, SimCS Vector): Vector
Begin
    Temp = getRelations(ont)
    For each pair in SimCS
        For each R in temp
            If R.domainConcepts.exist(p.Ca) then T1.add(R)
        Next
        T2.concept = p.Ca
        T2.roles= T1
        conceptRoles.add (T2)
    Next
    Return conceptRoles
End Function;
End Main
```

Figure 3.13: Acquisition of roles of contextually similar concepts
iii. a belongs to domains of some relations and ranges of some other relations simultaneously whereas b belongs to only ranges of some relations; i.e., in ontology-A, the explicit meaning of a Student concept may consist of its interactions with Debate-Competition concept [contestIn(Student, Debate-Competition)] and with Consultant concept [advise(Consultant, Student)] whereas in ontology-B, the same concept Student may interact with Consultant concept [advise(Consultant, Student)]

iv. vice versa of iii; i.e., in ontology-A, concept Student may interact only with Consultant concept [advise(Consultant, Student)] whereas the explicit meaning of a Student concept in Ontology-B may consist of its interactions with Debate-Competition concept [contestIn(Student, Debate-Competition)] and with Consultant concept [advise(Consultant, Student)].

v. a belongs to domains of some relations and ranges of some other relations simultaneously whereas b belongs to only domains of some relations; i.e., the explicit meaning of a (i.e. Student concept) is same as in case iii, whereas the explicit meaning of b may consist of its interaction with Scholarship concept [wins(Student, Scholarship)]

vi. Vice versa of v;

vii. a belongs to only domains whereas b belongs to only ranges of some relations; i.e., in ontology-A, concept Student may interact only with Scholarship concept [wins (Student, Scholarship)] whereas the explicit meaning of a Student concept in Ontology-B may consist of its interaction with Consultant concept [advice (Consultant, Student)].

viii. Vice versa of vii;

ix. a belongs to domains of some relations and ranges of some other relations simultaneously whereas b also belongs to domains of some relations and ranges of some other relations
simultaneously. i.e., in ontology-A, the explicit meaning of a Student concept may include its interactions with Debate-Competition and Project concepts [contestIn(Student, Debating-Competition), and Consultant concept [advise(Consultant, Student)] whereas in ontology-B, the same concept Student may also interact with similar concepts [contestIn(Student, Debating-Competition), [advise(Consultant, Student)]

Algorithm: Measuring of role-based similarity
Input: (i) $C^aS_A$, $C^bS_B$ Vectors
(ii) Sim CS Vector
Output: Sim RS (as defined in Eq. 3.22); a vector containing pairs of role-based similar concepts.

Begin
For each $p$ in Sim CS
  $rC_a = C^aS_A$.getRoles(p.Ca)
  $rC_b = C^bS_B$.getRoles(p.Cb)
  $T = countSame(rC_a, rC_b)$
  DoS = $T / ((rCa.size() + rCb.size()) - T)$
  SR = computeSR (T, rCa.size(), rCb.size())
  temp.Ca = p.Ca;
  temp.Cb = p.Cb
  temp.SR = SR
  temp.DoS = DoS
  Sim RS.add(temp)
Next
End Sub
Function countSame(Vector V_a, Vector V_b): Boolean
{same = 0
  For each $r_a$ in V_a
    For each $r_b$ in V_b
      If $r_a = r_b$ Then
        {same = same +1;
      Next
    Next
  Return same
End Function
Function computeSR(Integer T, integer n, integer m) : Return Char
Begin
  If (t = n) && (n - m = 0) Then Return ‘=’
  Else If (t = m) && (n - m > 0) Then Return ‘≥’
  Else If (t = n) && (n - m < 0) Then Return ‘≤’
  Else Return ‘X’
End Function
End Main

Figure 3.14: A slice of pseudo code for identifying role-based similarity
In Table 3.4, we have made a features based comparison between existing techniques and proposed technique; the $SM$, $DoS$ and $SR$ stand for Similarity Measurement, Degree of Similarity and Semantic Relation respectively. The different parameters as shown in Table 3.4 are already described in the beginning of same Chapter.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Existing Techniques</th>
<th>Proposed Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>I  Matching Strategy</td>
<td>Individual Matching</td>
<td>Integrated and layered matching</td>
</tr>
<tr>
<td>II Explicit-semantics based SM</td>
<td>Not supported</td>
<td>Supported</td>
</tr>
<tr>
<td>III Lexical SM</td>
<td>- Terms are compared</td>
<td>- Concepts are compared.</td>
</tr>
<tr>
<td></td>
<td>- $DoS$ is computed through string-based techniques(edit-distance, prefix, suffix and n-gram)</td>
<td>- $DoS$ is computed from granularities and explicit-semantics of concepts</td>
</tr>
<tr>
<td>IV Linguistic-semantics based SM</td>
<td>Supported</td>
<td>Domain specific semantics of concepts</td>
</tr>
<tr>
<td>V Contextual SM</td>
<td>Both the optional and mandatory characteristics are considered</td>
<td>Only mandatory characteristic with different criterion is considered</td>
</tr>
<tr>
<td>VI Output of overall SM</td>
<td>Pairs of similar concepts with either $DoS$ or $SR$</td>
<td>Pairs of similar concepts with both $DoS$ and $SR$</td>
</tr>
</tbody>
</table>
In previous chapter, we have proposed a technique for identifying and measuring similarity between two ontologies. The different processes of proposed technique have been represented in algorithmic form. Although, the coding is not a problem, to understand the problem and then design its solution is a problem; however, to validate the defined objectives automatically and to check the correctness of algorithms, here we formalize the proposed technique. However, a manual trace of algorithms using case studies is also given. We have taken different pairs of ontologies to analyze the results. The discussion about the performance evaluation of proposed technique is also given in this chapter.

4.1 Implementation

We have implemented the proposed technique in Java language with the help of an integrated development environment - NetBeans IDE 6.1 [83]. In order to load and parse ontologies, OWL API [70, 82] has been used. For retrieval of concepts along-with their respective super-concepts, the relevant methods of certain classes of “org.semanticweb.owl.model” [84], and “org.semanticweb.owl.apibinding” [84], packages have been used. Similarly, “org.semanticweb.owl.io” [84] package has been used for loading and parsing of ontologies. The implementation of prototype as shown in Figure 4.1 includes different modules, which are listed on next page:
Figure 4.1: A prototype for similarity identification and measurement

4.1.1. Main Module

// Sample lines of pseudo code

- OWLOntology ontologyA = loadOntologyFromFile(FILE_PATH);
- OWLOntology ontologyB = loadOntologyFromFile(FILE_PATH);
- Vector<String> classesVectorA = getOntologyConcepts(ontologyA);
- Vector<String> classesVectorB = getOntologyConcepts(ontologyB);
- Class S_Pair { String conceptA; String conceptB; Float DOS; Char SR; }
- Vector<S_Pair> PS_Pairs = identPrimarySim(VectorA, VectorB);
- Class c_Parent { String concept C; Vector superConcepts; }
- Vector <c_parent> CPA = getSuperConcepts(A, PS_Pairs);
Vector <c_parent> CPB = getSuperConcepts(B, PS_Pairs);
Vector <S_Pair> CS_Pairs = identContxtualSim(CPA, CPB, PS_Pairs);
Class c_role{ String concept; Vector setOfRoles;}
Vector <c_role> CRA = getRoles(A, CS_Pairs);
Vector <c_role> CRB = getRoles(B, CS_Pairs);
Vector<S_Pair> RS_Pairs = identRolebasedSim(CRA, CAB, CS_Pairs);

4.1.2. Ontology Loading Module

// Sample lines of pseudo code
OWLOntology loadOntologyFromFile(String filePath){
    OWLOntology ontology = null;
    File owlFile = new File(filePath);
    OWLOntologyInputSource inputSource = new FileInputSource(owlFile);
    OWLOntologyInputSource ontologyInputSource =
        (OWLOntologyInputSource)inputSource;
    OWLOntologyManager owlOntologyManager = OWLManager.createOWLOntologyManager();
    ontology = owlOntologyManager.loadOntology(ontologyInputSource);
    return ontology;
}

4.1.3. Concepts Acquisition Module

// Sample lines of pseudo code
Vector <String> getOntologyConcepts(OwlOntology ontology){
    Set<OWLClass> ontSet = ontology.getReferencedClasses();
    if(ontSet == null || ontSet.size()==0){return null;}
    classMap = new HashMap<String, Vector<String>>();
    Object[] classes = ontSet.toArray();
Vector<String> classNameVector = new Vector<String>(classes.length);
for(k=0; k<classes.length; k++){
  OWLClass owlClass = (OWLClass)classes[k];
  //Algorithm is given in Figure 3.6
  // store owlClass name in classNameVector.
  return classNameVector;
}

4.1.4. Primary Similarity Measurement Module

// Sample lines of code
Vector<S_Pair> identPrimarySimilarity(Vector<String> conceptsA, Vector<String> conceptsB) {
  Vector<S_Pair> sPairVector;
  Algorithm is given in Fig 3.7
  sPairVector population
  return sPairVector;
}

4.1.5. Super-concepts Acquisition Module

// Sample lines of code
Vector<c_parent> getSuperConcepts(OwlOntology ontology, Vector PS_Pairs)
{
  Vector<c_parent> c_parentVector;
  c_parentVector population;
  Detailed algorithm is given in Figure 3.8
  return c_parentVector;
}
4.1.6. Contextual Similarity Measurement Module

// Sample lines of pseudo code
- Vector<S_Pair> identContextualSimilarity
- (Vector<String> A, Vector<String> B, Vector<S_Pair> psPair)
- {Vector <S_Pair> CSVector;
- CSVector population
- Algorithm is given in Fig 3.9
- return CSVector;
- }

4.1.7. Roles Acquisition Module

// Sample lines of pseudo code
- Vector <c_role> getRoles(OwlOntology ontology, Vector <S_Pair> csPair){
- Vector <c_role> cRoles;
- cRoles population;
- Detailed algorithm is given in Figure 3.10
- Return cRoles;
- }

4.1.8. Role-based Similarity Measurement Module

// Sample lines of pseudo code
- Vector <S_Pair> identRolebasedSimilarity (Vector <c_role> CRA,
  Vector <c_role> CRB, Vector <S_Pair> CS_Pairs){
- Vector <S_Pair> rsPairs
- rsPairs population
- Algorithm is given in Fig 3.11
- return rsPairs;
- }
4.2 Testing

We evaluate the proposed technique by taking into consideration the objectives, as listed in Chapter-1. The Education and the Business domains have been taken as sample domains for testing the proposed technique. We took *Software Development organization* from Business domain and *University* from Education domain. From these domains, different pairs of ontologies are chosen as input ontologies for proposed technique. The results are verified by respective domain experts and are declared satisfactory.

4.2.1. Case Study 1: Software Development Organization

The ontologies of software development organizations, as we selected, are mainly concentrated on human-resources along with their roles, i.e., the intellectual concepts and their interactions with non-intellectual concepts. A software organization has different categories of intellectual concepts such as technical and non-technical human resources. The category of technical human resources is further divided in different teams such as Analysis-team, Design-team, Implementation-team, SQA-team, Supplemental-team and Deployment-team. There are different concepts in each team such as Analyst, Use-case Engineer, Software Engineer, Programmer, Coder, SQA-engineer, Technical-Writer, Librarian, and Project Manager. They work on projects, where each project has different modules. A common vocabulary of this domain is given in Appendix V. These intellectual concepts are commonly used in different software development organizations with same, overlapped or different roles. The roles/responsibilities are usually categorized with respective to different development phases and their supporting activities, as partially listed in Appendix-II.
4.2.1.1 List of Roles

In order to manually trace the proposed technique, we have taken a subset of commonly used roles by the intellectual concepts of these ontologies, which are listed in Figure 4.2. For more detail, please see the Appendix-II. A subset of domain vocabulary is given in Appendix-III.

<table>
<thead>
<tr>
<th>(r1)</th>
<th>Analyze Hardware Requirements</th>
<th>(r2)</th>
<th>Analyze Software Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>(r3)</td>
<td>Analyze Functional Requirements</td>
<td>(r4)</td>
<td>Analyze Non Functional Requirements</td>
</tr>
<tr>
<td>(r5)</td>
<td>Analyze Cost Benefit</td>
<td>(r6)</td>
<td>Design Database</td>
</tr>
<tr>
<td>(r7)</td>
<td>Design Algorithms</td>
<td>(r8)</td>
<td>Design Reports</td>
</tr>
<tr>
<td>(r9)</td>
<td>Design Input Screens</td>
<td>(r10)</td>
<td>Design Structure</td>
</tr>
<tr>
<td>(r11)</td>
<td>Design Graphics</td>
<td>(r12)</td>
<td>Design Web Pages</td>
</tr>
<tr>
<td>(r13)</td>
<td>Implement Database</td>
<td>(r14)</td>
<td>Implement Algorithm</td>
</tr>
<tr>
<td>(r15)</td>
<td>Implement Reports</td>
<td>(r16)</td>
<td>Implement GUI</td>
</tr>
<tr>
<td>(r17)</td>
<td>Implement Structure</td>
<td>(r18)</td>
<td>Write Requirements Specifications</td>
</tr>
<tr>
<td>(r19)</td>
<td>Write Design Documents</td>
<td>(r20)</td>
<td>Write Code Documents</td>
</tr>
<tr>
<td>(r21)</td>
<td>Test Functional Requirements</td>
<td>(r22)</td>
<td>Test Non Functional Requirements</td>
</tr>
<tr>
<td>(r23)</td>
<td>Test Procedures</td>
<td>(r24)</td>
<td>Tune Database</td>
</tr>
<tr>
<td>(r25)</td>
<td>Backup Database</td>
<td>(r26)</td>
<td>Cost Management</td>
</tr>
<tr>
<td>(r27)</td>
<td>Resource Management</td>
<td>(r28)</td>
<td>Define standard operating procedures</td>
</tr>
<tr>
<td>(r29)</td>
<td>Change Management</td>
<td>(r30)</td>
<td>Write User Manual</td>
</tr>
<tr>
<td>(r31)</td>
<td>Software configuration control</td>
<td>(r32)</td>
<td>Storing final released products</td>
</tr>
<tr>
<td>(r33)</td>
<td>Developing a test plan for the project</td>
<td>(r34)</td>
<td>Allocating database resources to projects</td>
</tr>
<tr>
<td>(r35)</td>
<td>Compiling source code/linking/building</td>
<td>(r36)</td>
<td>Defining user profiles</td>
</tr>
<tr>
<td>(r37)</td>
<td>Creating test baselines</td>
<td>(r38)</td>
<td>Ensuring Inter-group coordination</td>
</tr>
<tr>
<td>(r39)</td>
<td>Deploying applications in virtual machine environments</td>
<td>(r40)</td>
<td>Ensuring successful project closure</td>
</tr>
<tr>
<td>(r41)</td>
<td>Establishing SCCB and SCRB for projects</td>
<td>(r42)</td>
<td>Ensuring that SQA activities are carried out as planned</td>
</tr>
<tr>
<td>(r43)</td>
<td>Faxing, mailing, shipping and other general administrative duties</td>
<td>(r44)</td>
<td>Ensuring the security of project databases</td>
</tr>
<tr>
<td>(r45)</td>
<td>Handling and maintaining the company store</td>
<td>(r46)</td>
<td>Identification of project based SCM tool(s)</td>
</tr>
</tbody>
</table>

Figure 4.2: A subset of roles in software development organization
4.2.1.2 List of Concepts of First Input Ontology

Table 4.1: A sample slice of intellectual concepts form A ontology

<table>
<thead>
<tr>
<th>Id</th>
<th>Concept</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a1)</td>
<td>SoftwareEngineer</td>
<td>r7, r10, r13, r16</td>
</tr>
<tr>
<td>(a2)</td>
<td>SeniorSoftwareEngineer</td>
<td>r3, r4, r7, r10</td>
</tr>
<tr>
<td>(a3)</td>
<td>Programmer</td>
<td>r12, r13, r14, r15, r16, r17</td>
</tr>
<tr>
<td>(a4)</td>
<td>SeniorProgrammer</td>
<td>r6, r7, r8, r9, r10</td>
</tr>
<tr>
<td>(a5)</td>
<td>Designer</td>
<td>r11, r12</td>
</tr>
<tr>
<td>(a6)</td>
<td>Analyst</td>
<td>r1, r2, r3</td>
</tr>
<tr>
<td>(a7)</td>
<td>SeniorAnalyst</td>
<td>r3, r4, r5</td>
</tr>
<tr>
<td>(a8)</td>
<td>SQAEngineer</td>
<td>r21, r22, r23</td>
</tr>
<tr>
<td>(a9)</td>
<td>DBA</td>
<td>r6, r13, r24, r25</td>
</tr>
<tr>
<td>(a10)</td>
<td>TechnicalWriter</td>
<td>r18, r19, r20, r30</td>
</tr>
<tr>
<td>(a11)</td>
<td>ProjectManager</td>
<td>r26, r27</td>
</tr>
<tr>
<td>(a12)</td>
<td>ProcessManager</td>
<td>r28</td>
</tr>
</tbody>
</table>

The list of sample concepts of first input ontology `dataSoft.owl` is shown in Table 4.1 (for detail, please see Appendix-IV). Just for the sake of simplicity, we have chosen only those concepts which are contextually similar. The domain vocabulary includes the concepts of this ontology.

4.2.1.3 List of Concepts of Second Input Ontology

The ontology `ridos.owl` is chosen as second input ontology. This ontology is also considered while populating domain vocabulary. A subset of its concepts is shown in Table 4.2 (For detail please sees Appendix-V).
Table 4.2: A sample slice of intellectual concepts form B ontology

<table>
<thead>
<tr>
<th>Id</th>
<th>Concept</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b1)</td>
<td>ProjectManager</td>
<td>r26, r27, r28</td>
</tr>
<tr>
<td>(b2)</td>
<td>SofConfigManager</td>
<td>r29</td>
</tr>
<tr>
<td>(b3)</td>
<td>SoftwareEngineer</td>
<td>r4, r7, r10, r13</td>
</tr>
<tr>
<td>(b4)</td>
<td>SQAEngineer</td>
<td>r21, r22</td>
</tr>
<tr>
<td>(b5)</td>
<td>Programmer</td>
<td>r6, r7, r8, r9, r10, r12-r17</td>
</tr>
<tr>
<td>(b6)</td>
<td>Designer</td>
<td>r11, r12</td>
</tr>
<tr>
<td>(b7)</td>
<td>Analyst</td>
<td>r1, r2, r3, r4</td>
</tr>
<tr>
<td>(b8)</td>
<td>Coder</td>
<td>r12, r13, r14, r15, r16, r17</td>
</tr>
<tr>
<td>(b9)</td>
<td>DBA</td>
<td>r13, r24, r25</td>
</tr>
<tr>
<td>(b10)</td>
<td>SoftwareArchitect</td>
<td>r8, r9, r11, r12</td>
</tr>
<tr>
<td>(b11)</td>
<td>TechnicalWriter</td>
<td>r18, r19, r20</td>
</tr>
</tbody>
</table>

4.2.1.4 List of Pairs of Similar Concepts

Table 4.3: A sample list of similar concepts, including pairs having minimum degree of similarity

<table>
<thead>
<tr>
<th>Pairs</th>
<th>Pair of Concepts</th>
<th>DoS</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a1, b3)</td>
<td>(A:SoftwareEngineer, B:SoftwareEngineer)</td>
<td>0.60</td>
<td>X</td>
</tr>
<tr>
<td>(a1, b5)</td>
<td>(A:SoftwareEngineer, B:Programmer)</td>
<td>0.36</td>
<td>&lt;</td>
</tr>
<tr>
<td>(a1, b8)</td>
<td>(A:SoftwareEngineer, B:Coder)</td>
<td>0.25</td>
<td>X</td>
</tr>
<tr>
<td>(a2, b3)</td>
<td>(A:SeniorSoftwareEngineer, B:SoftwareEngineer)</td>
<td>0.60</td>
<td>X</td>
</tr>
<tr>
<td>(a2, b5)</td>
<td>(A:SeniorSoftwareEngineer, B:Programmer)</td>
<td>0.15</td>
<td>X</td>
</tr>
<tr>
<td>(a3, b3)</td>
<td>(A:Programmer, B:SoftwareEngineer)</td>
<td>0.11</td>
<td>X</td>
</tr>
<tr>
<td>(a3, b5)</td>
<td>(A:Programmer, B:Programmer)</td>
<td>0.55</td>
<td>&lt;</td>
</tr>
<tr>
<td>(a3, b6)</td>
<td>(A:Programmer, B:Designer)</td>
<td>0.14</td>
<td>X</td>
</tr>
<tr>
<td>(a3, b8)</td>
<td>(A:Programmer, B:Coder)</td>
<td>1.00</td>
<td>=</td>
</tr>
<tr>
<td>(a3, b10)</td>
<td>(A:Programmer, B:SoftwareArchitect)</td>
<td>0.11</td>
<td>X</td>
</tr>
<tr>
<td>(a4, b3)</td>
<td>(A:SeniorProgrammer, B:SoftwareEngineer)</td>
<td>0.29</td>
<td>X</td>
</tr>
<tr>
<td>(a4, b5)</td>
<td>(A:SeniorProgrammer, B:Programmer)</td>
<td>0.45</td>
<td>&lt;</td>
</tr>
<tr>
<td>(a4, b10)</td>
<td>(A:SeniorProgrammer, B:SoftwareArchitect)</td>
<td>0.29</td>
<td>X</td>
</tr>
<tr>
<td>(a5, b6)</td>
<td>(A:Designer, B:Designer)</td>
<td>1.00</td>
<td>=</td>
</tr>
<tr>
<td>(a5, b8)</td>
<td>(A:Designer, B:Coder)</td>
<td>0.14</td>
<td>X</td>
</tr>
<tr>
<td>(a5, b10)</td>
<td>(A:Designer, B:SoftwareArchitect)</td>
<td>0.50</td>
<td>&lt;</td>
</tr>
<tr>
<td>(a6, b7)</td>
<td>(A:Analyst, B:Analyst)</td>
<td>0.75</td>
<td>&lt;</td>
</tr>
<tr>
<td>(a7, b3)</td>
<td>(A:SeniorAnalyst, B:SoftwareEngineer)</td>
<td>0.17</td>
<td>X</td>
</tr>
<tr>
<td>(a7, b7)</td>
<td>(A:SeniorAnalyst, B:Analyst)</td>
<td>0.40</td>
<td>X</td>
</tr>
<tr>
<td>(a8, b4)</td>
<td>(A:SQAEngineer, B:SQAEngineer)</td>
<td>0.66</td>
<td>&gt;</td>
</tr>
<tr>
<td>(a9, b3)</td>
<td>(A:DBA, B:SoftwareEngineer)</td>
<td>0.14</td>
<td>X</td>
</tr>
<tr>
<td>(a9, b5)</td>
<td>(A:DBA, B:Programmer)</td>
<td>0.15</td>
<td>X</td>
</tr>
<tr>
<td>(a9, b8)</td>
<td>(A:DBA, B:Coder)</td>
<td>0.11</td>
<td>X</td>
</tr>
<tr>
<td>(a9, b9)</td>
<td>(A:DBA, B:DBA)</td>
<td>0.75</td>
<td>&gt;</td>
</tr>
<tr>
<td>(a10, b11)</td>
<td>(A:TechnicalWriter, B:TechnicalWriter)</td>
<td>0.75</td>
<td>&gt;</td>
</tr>
<tr>
<td>(a11, b1)</td>
<td>(A:ProjectManager, B:ProjectManager)</td>
<td>0.66</td>
<td>&lt;</td>
</tr>
<tr>
<td>(a12, b1)</td>
<td>(A:ProcessManager, B:ProjectManager)</td>
<td>0.33</td>
<td>&lt;</td>
</tr>
</tbody>
</table>
In Table 4.3, the values of DoS and SR are computed according to formulas given in Equations 3.9 and 3.10. The resultant pairs in Table 4.3 are further short-listed by applying a certain threshold value i.e., 0.45, as shown in Table 4.4.

Table 4.4 A slice of role-based similar concepts with a threshold-value

<table>
<thead>
<tr>
<th>Pairs</th>
<th>Pair of Concepts</th>
<th>DoS</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a1, b3)</td>
<td>(A:SoftwareEngineer, B:SoftwareEngineer)</td>
<td>0.60</td>
<td>X</td>
</tr>
<tr>
<td>(a2, b3)</td>
<td>(A:SeniorSoftwareEngineer, B:SoftwareEngineer)</td>
<td>0.60</td>
<td>X</td>
</tr>
<tr>
<td>(a3, b5)</td>
<td>(A:Programmer, B:Programmer)</td>
<td>0.55</td>
<td>&lt;</td>
</tr>
<tr>
<td>(a3, b8)</td>
<td>(A:Programmer, B:Coder)</td>
<td>1.00</td>
<td>=</td>
</tr>
<tr>
<td>(a4, b5)</td>
<td>(A:SeniorProgrammer, B:Programmer)</td>
<td>0.45</td>
<td>&lt;</td>
</tr>
<tr>
<td>(a5, b6)</td>
<td>(A:Designer, B:Designer)</td>
<td>1.00</td>
<td>=</td>
</tr>
<tr>
<td>(a5, b10)</td>
<td>(A:Designer, B:SoftwareArchitect)</td>
<td>0.50</td>
<td>&lt;</td>
</tr>
<tr>
<td>(a6, b7)</td>
<td>(A:Analyst, B:Analyst)</td>
<td>0.75</td>
<td>&lt;</td>
</tr>
<tr>
<td>(a8, b4)</td>
<td>(A:SQAEngineer, B:SQAEngineer)</td>
<td>0.66</td>
<td>&gt;</td>
</tr>
<tr>
<td>(a9, b9)</td>
<td>(A:DBA, B:DBA)</td>
<td>0.75</td>
<td>&gt;</td>
</tr>
<tr>
<td>(a10, b11)</td>
<td>(A:TechnicalWriter, B:TechnicalWriter)</td>
<td>0.75</td>
<td>&gt;</td>
</tr>
<tr>
<td>(a11, b1)</td>
<td>(A:ProjectManager, B:ProjectManager)</td>
<td>0.66</td>
<td>&lt;</td>
</tr>
</tbody>
</table>

The following scenarios are also examined for a pair P (a, b | a ∈ A ∧ b ∈ B):

i. The P is primarily similar whereas a and b holds equivalent (i.e., =) semantic relation.
ii. The P is primarily similar whereas a is more generic than b (i.e. a ⊇ b).
iii. The P is primarily similar whereas a is less generic than b (i.e. a ⊆ b).
iv. The P is not primarily similar.
v. Scenario (i) is true and the P is also contextually similar.
vi. Scenario (i) is true but the P is not contextually similar.
vii. Scenario (ii) is true and P is also contextually similar.
viii. Scenario (ii) is true but the P is not contextually similar.
ix. Scenario (iii) is true and P is also contextually similar.
x. Scenario (iii) is true but P is not contextually similar.
xi. Scenario (iv) is true and P is contextually similar.
xii. Scenario (iv) is true and $P$ is not contextually similar.

xiii. Scenarios (v), (vii) & (ix) are true and $P$ is role-based similar.

xiv. Scenarios (v), (vii) & (ix) are true but $P$ is not role-based similar.

xv. Scenario (xii) is true and $P$ is role-based similar.

xvi. Scenario (xii) is true but $P$ is not role-based similar

All above scenarios are also summarized in Table 4.5.

Table 4.5: Different scenarios for a pair of concepts to similarity identification

<table>
<thead>
<tr>
<th>Test-cases</th>
<th>Pair of Concepts</th>
<th>Primary Similarity (1st Level)</th>
<th>Contextual Similarity (2nd Level)</th>
<th>Role-based Similarity (3rd Level)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case: 1.</td>
<td>(a, b)</td>
<td>True, =</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Case: 2.</td>
<td>(a, b)</td>
<td>True, $\supseteq$</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Case: 3.</td>
<td>(a, b)</td>
<td>True, $\subseteq$</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Case: 4.</td>
<td>(a, b)</td>
<td>False,</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Case: 5.</td>
<td>(a, b)</td>
<td>True, =</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>Case: 6.</td>
<td>(a, b)</td>
<td>True, $\supseteq$</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>Case: 7.</td>
<td>(a, b)</td>
<td>True, $\subseteq$</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>Case: 8.</td>
<td>(a, b)</td>
<td>False,</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>Case: 9.</td>
<td>(a, b)</td>
<td>True, =</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Case: 10.</td>
<td>(a, b)</td>
<td>True, $\supseteq$</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Case: 11.</td>
<td>(a, b)</td>
<td>True, $\subseteq$</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Case: 12.</td>
<td>(a, b)</td>
<td>False,</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Case: 13.</td>
<td>(a, b)</td>
<td>True, =</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>Case: 14.</td>
<td>(a, b)</td>
<td>True, $\supseteq$</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>Case: 15.</td>
<td>(a, b)</td>
<td>True, $\subseteq$</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>Case: 16.</td>
<td>(a, b)</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

4.3 Case Study 2: University – The Education Organization

For manual exercise of different scenarios as listed in Table 4.5, we have taken $CSUET^1$ as $A$ ontology and the $LCWU^2$ is taken as $B$ ontology. The semantic relation between a pair of concepts has been computed based on their respective granularities.

4.3.1 List of Sample Concepts of Input Ontologies

The sample concepts from A ontology are: (a₁) Project, (a₂) ITConsultant, (a₃) Director, (a₄) Manager, (a₅) UnderGradStudent, (a₆) Convener, (a₇) Course, (a₈) Professor, (a₉) Quiz, (a₁₀) Workshop, (a₁₁) NationalConference, (a₁₂) ResearchCentre, (a₁₃) PostGradStudent, (a₁₄) Person, (a₁₅) Deptt. The sample concepts from B ontology are: (b₁) TermProject, (b₂) Consultant, (b₃) Director, (b₄) SupportManager, (b₅) ConvenerAdmission, (b₆) Student, (b₇) Professor, (b₈) PostGradCourse, (b₉) Workshop, (b₁₀) Conference, (b₁₁) ResearchCentre, (b₁₂) Department, (b₁₃) SoftwareEngineer, (b₁₄) Person, (b₁₅) Employee, (b₁₆) Faculty. The sample pairs are (a₁, b₁₂), (a₂, b₂), (a₃, b₃), (a₄, b₄), (a₅, b₆), (a₆, b₅), (a₇, b₇), (a₈, b₇), (a₉, b₈), (a₉, b₁₀), (a₁₀, b₉) and (a₁₁, b₁₀) respectively.

4.3.2 Primary Similarity Identification and Measurement

Input: A = (a₁, a₂, a₃, a₄, a₅, a₆, a₇, a₈, a₉, a₁₀, a₁₁, a₁₂)
B = (b₁, b₂, b₃, b₄, b₅, b₆, b₇, b₈, b₉, b₁₀, b₁₁, b₁₁)

Output:
\[ \text{Sim}_{PS} = \{(a₂, b₂, 0.80, \leq), (a₃, b₃, 1.00, =), (a₄, b₄, 0.75, \geq), (a₅, b₆, 0.84, \leq), (a₆, b₅, 0.80, \geq), (a₇, b₇, 1, =), (a₈, b₈, 0.66, \geq), (a₁₀, b₉, 1, =), (a₁₁, b₁₀, 0.86, \leq)\} \]

4.3.3 Contextual Similarity Identification and Measurement

Input: Sim_{PS} and super-concepts of (a₂, a₃, a₄, a₅, a₆, a₈, a₉, a₁₀, a₁₁) and super-concepts of (b₂, b₃, b₄, b₅, b₆, b₇, b₈, b₉ and b₁₀).

Output:
\[ \text{Sim}_{CS} = \{(a₂, b₂, 0.80, \leq), (a₃, b₃, 1, =), (a₄, b₄, 0.75, \geq), (a₅, b₆, 0.84, \leq), (a₇, b₇, 1, =), (a₉, b₈, 0.66, \geq)\} \]
4.3.4 Role-based Similarity Identification and Measurement

Input: $\text{Sim}_\text{CS}$ and roles of ($a_2$, $a_3$, $a_4$, $a_5$, $a_6$, $a_8$, $a_9$) and roles of ($b_2$, $b_3$, $b_4$, $b_6$, $b_7$, $b_8$).

Output: $\text{Sim}_\text{RS} = \{(a_3, b_3, 1, '='), (a_5, b_6, 0.84, '\leq'), (a_8, b_7, 1, '='), (a_9, b_8, 0.66, '\geq')\}$

The individual status of primary similarity ($PS$), contextual similarity ($CS$) and the role-based similarity ($RS$) of different pairs of concepts as examined above is shown in the Table 4.6 which can be helpful for analyzing the overall result of proposed technique.

Table 4.6: Results of first, second and third levels of similarities

<table>
<thead>
<tr>
<th>Case</th>
<th>Pair of Concepts</th>
<th>PS (1st Level Sim.)</th>
<th>CS (2nd Level Sim.)</th>
<th>RS (3rd Level Sim.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>case i.</td>
<td>(a8, b7)</td>
<td>True, $=$</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>case ii.</td>
<td>(a9, b8)</td>
<td>True, $\supseteq$</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>case iii.</td>
<td>(a5, b6)</td>
<td>True, $\subseteq$</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>case iv.</td>
<td>(a1, b12)</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>case v.</td>
<td>(a10, b9)</td>
<td>True, $=$</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>case vi.</td>
<td>(a6, b5)</td>
<td>True, $\supseteq$</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>case vii.</td>
<td>(a11, b10)</td>
<td>True, $\subseteq$</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>case viii.</td>
<td>(a9, b10)</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>case ix.</td>
<td>(a3, b3)</td>
<td>True, $=$</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>case x.</td>
<td>(a4, b4)</td>
<td>True, $\supseteq$</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>case xi.</td>
<td>(a2, b2)</td>
<td>True, $\subseteq$</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>case xii.</td>
<td>(a7, b7)</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>case xiii.</td>
<td>(a10, b9)</td>
<td>True, $=$</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>case xiv.</td>
<td>(a6, b5)</td>
<td>True, $\supseteq$</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>case xv.</td>
<td>(a11, b10)</td>
<td>True, $\subseteq$</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>case xvi.</td>
<td>(a9, b10)</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>case xvii.</td>
<td>(a1, b1)</td>
<td>True, $\supseteq$</td>
<td>True</td>
<td>Null</td>
</tr>
<tr>
<td>case xviii.</td>
<td>(a14, b12)</td>
<td>True, $=$</td>
<td>True</td>
<td>Null</td>
</tr>
</tbody>
</table>
4.4 Observations about Results of Different Test-scenarios

We have the following observations about results summarized in Table 4.6

Observation 1: If the 1st level similarity for a pair is true then there may be true or false for the next levels of similarities.

Observation 2: If the 1st level similarity for a pair is false, then for its 2nd level and 3rd level of similarities is always false.

Observation 3: The third level similarity is null for a pair of non-intellectual concepts possessing 2nd level of similarity.

Table 4.7: Observations about results from different test-scenarios

<table>
<thead>
<tr>
<th>Case</th>
<th>Pair of Concepts</th>
<th>PS (1st Level)</th>
<th>CS (2nd Level)</th>
<th>RS (3rd Level)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case: 1.</td>
<td>(a, b)</td>
<td>True, =</td>
<td>True(It’s Possible)</td>
<td>True(It’s Possible)</td>
</tr>
<tr>
<td>Case: 2.</td>
<td>(a, b)</td>
<td>True, &gt;</td>
<td>True(It’s Possible)</td>
<td>True(It’s Possible)</td>
</tr>
<tr>
<td>Case: 3.</td>
<td>(a, b)</td>
<td>True, &lt;</td>
<td>True(It’s Possible)</td>
<td>True(It’s Possible)</td>
</tr>
<tr>
<td>Case: 4.</td>
<td>(a, b)</td>
<td>False</td>
<td>True (It’s Always False)</td>
<td>True(It’s Always False)</td>
</tr>
<tr>
<td>Case: 5.</td>
<td>(a, b)</td>
<td>True, =</td>
<td>False(It’s Possible)</td>
<td>True(It’s Always False)</td>
</tr>
<tr>
<td>Case: 6.</td>
<td>(a, b)</td>
<td>True, &gt;</td>
<td>False(It’s Possible)</td>
<td>True(It’s Always False)</td>
</tr>
<tr>
<td>Case: 7.</td>
<td>(a, b)</td>
<td>True, &lt;</td>
<td>False(It’s Possible)</td>
<td>True(It’s Always False)</td>
</tr>
<tr>
<td>Case: 8.</td>
<td>(a, b)</td>
<td>False</td>
<td>False(It’s Always True)</td>
<td>True(It’s Always False)</td>
</tr>
<tr>
<td>Case: 9.</td>
<td>(a, b)</td>
<td>True, =</td>
<td>True(It’s Possible)</td>
<td>False (It’s Possible)</td>
</tr>
<tr>
<td>Case: 10.</td>
<td>(a, b)</td>
<td>True, &gt;</td>
<td>True(It’s Possible)</td>
<td>False (It’s Possible)</td>
</tr>
<tr>
<td>Case: 11.</td>
<td>(a, b)</td>
<td>True, &lt;</td>
<td>True(It’s Possible)</td>
<td>False (It’s Possible)</td>
</tr>
<tr>
<td>Case: 12.</td>
<td>(a, b)</td>
<td>False</td>
<td>True(It’s Always False)</td>
<td>False (It’s Possible)</td>
</tr>
<tr>
<td>Case: 13.</td>
<td>(a, b)</td>
<td>True, =</td>
<td>False(It’s Possible)</td>
<td>False (It’s Always True)</td>
</tr>
<tr>
<td>Case: 14.</td>
<td>(a, b)</td>
<td>True, &gt;</td>
<td>False(It’s Possible)</td>
<td>False (It’s Always True)</td>
</tr>
<tr>
<td>Case: 15.</td>
<td>(a, b)</td>
<td>True, &lt;</td>
<td>False(It’s Possible)</td>
<td>False(It’s Always True)</td>
</tr>
<tr>
<td>Case: 16.</td>
<td>(a, b)</td>
<td>False</td>
<td>False(It’s Always True)</td>
<td>False(It’s Always True)</td>
</tr>
</tbody>
</table>
Observation 4: There is a role-based similarity between concepts of pair \((a_{13}, b_7)\), i.e., \(A:PostGradStudent\) and \(B:Professor\), because both work-on research-project; also, there is contextual-similarity between these concepts. Same is the case of \((a_8, b_{13})\) i.e. \(A:Professor\) and \(B:SoftwareEngineer\) both are working on Project. These pairs are not primarily similar. Since the main motive behind finding similarity between concepts is merging, aligning or mapping of ontologies for knowledge sharing; therefore, the merging, aligning or mapping of \(PostGradStudent\) concept with \(Professor\) concept, is not recommended. In proposed technique, a pair having no primary similarity is simply discarded.

From these observations, the correctness of layer strategy adopted in proposed technique has been proved. The primary similarity of concepts is the prerequisite of contextual similarity and the contextual similarity is prerequisite of role-based similarity. However, it is not necessary that two primarily similar concepts are contextually similar or two contextually similar concepts are role-based similar.

4.5 Results: Analysis and Discussion

To realize achievement of the different objectives, as listed in Chapter 1, we compare the results of proposed technique with the results from some existing techniques. The criteria for comparison include the (i) completeness; (ii) correctness and (iii) overall quality of results in a less execution-time. The existing techniques [15, 16, 19, 23, 24, 26, 30, 52, 61] used in comparison are already overviewed in Chapter 2. The achievement of objectives has been confirmed during the analysis of results.
4.5.1 Completeness

The completeness of a similarity identifying technique is just like the precision measures used in information retrieval [31-33]. It is the ratio of correct number of pairs found divided by the total number of pairs found. Let totalPairsFound be the total number of pairs found in which CorrectPairsFound number of pairs are correct, such as totalPairsFound >= CorrectPairsFound, then the completeness can be formally written as:

\[
\text{Completeness} = \frac{\text{CorrectPairsFound}}{\text{TotalPairsFound}}
\]  (4.1)

4.5.2 Correctness

The correctness of a similarity identifying technique is just like the recall measures used in information retrieval [31-33]. The correctness is the ratio of correct number of pairs found, divided by the expected number of correct pairs. Let Correct_Pairs_Expected be the total number of correct pairs expected and Correct_Pairs_Found number is of correct pairs found by a technique such as Correct_Pairs_Expected >= Correct_Pairs_Found, then the correctness can be formally written as

\[
\text{Correctness} = \frac{\text{Correct}_\text{Pairs}_\text{Found}}{\text{Correct}_\text{Pairs}_\text{Expected}}
\]  (4.2)

4.5.3 Overall Quality of Result

The overall quality (OQ) of result is based on correctness and completeness of result. It is computed just as f-measure [31-33], used in information retrieval.

\[
OQ = 2 \times \frac{\text{Completeness} \times \text{Correctness}}{\text{Completeness} + \text{Correctness}}
\]  (4.3)
Through layered strategy, the output of first layer is used as input for the second layer and so on, whereas the output of first layer is set of pairs of concepts having primary similarity while all other concepts are discarded in the output. This means that the input to second layer is a short list of concepts instead of all concepts which reduce a reasonable execution-time for 2\textsuperscript{nd} level of similarity identification. Similarly the concepts shorted-listed in second layer are input to third layer. Therefore, the overall execution-time of proposed technique is comparatively short.

4.5.4 Test Cases for Evaluating Performance

We have taken four pairs of ontologies as shown in Table 4.8 to evaluate the completeness, correctness and overall quality of results of proposed technique up to second level of similarity. And, then it is followed by the evaluation of the role-based similarity i.e. the 3\textsuperscript{rd} level similarity, based on new criterion. The details of these ontologies are given in Appendix-IV. Comparisons of results are then made with expected results and with the results of existing matching techniques used in different tools and systems. While choosing sample input pairs in each test case, we have considered different test scenarios which are already summarized in Table 4.5.

<table>
<thead>
<tr>
<th>TestCase</th>
<th>Ontologies</th>
<th>Input Pairs</th>
<th>Similar Pairs (expected)</th>
<th>Similar Pairs (with different terms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1, B_1</td>
<td>37</td>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>A2, B_2</td>
<td>40</td>
<td>22</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>A_3, B_3</td>
<td>28</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>A_4, B_4</td>
<td>25</td>
<td>15</td>
<td>12</td>
</tr>
</tbody>
</table>
4.5.4.1 Evaluating Performance with First Test Case

Sample input pairs: 37

Pairs of similar concepts (expected): 25

Similar pairs (out of 25) with different terms: 10

With respect to test case 1, the results from proposed technique (SIMTO) and from some existing techniques are compared in Table 4.9, with respect to their completeness, correctness and overall quality. The graphical representation of comparison is also given in the Figures 4.3, 4.4 and 4.5 respectively. A comparative improvement in result of proposed technique, with respect to completeness is realized.

<table>
<thead>
<tr>
<th></th>
<th>Pairs of similar concepts returned</th>
<th>Correct pairs found</th>
<th>Completeness</th>
<th>Correctness</th>
<th>Overall Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSBO[61]</td>
<td>19</td>
<td>15</td>
<td>0.79</td>
<td>0.6</td>
<td>0.68</td>
</tr>
<tr>
<td>SEMC[19]</td>
<td>20</td>
<td>17</td>
<td>0.85</td>
<td>0.68</td>
<td>0.76</td>
</tr>
<tr>
<td>LOM[23]</td>
<td>20</td>
<td>17</td>
<td>0.85</td>
<td>0.68</td>
<td>0.76</td>
</tr>
<tr>
<td>HCOME[15]</td>
<td>20</td>
<td>16</td>
<td>0.8</td>
<td>0.64</td>
<td>0.71</td>
</tr>
<tr>
<td>RTHO[24]</td>
<td>21</td>
<td>18</td>
<td>0.86</td>
<td>0.72</td>
<td>0.78</td>
</tr>
<tr>
<td>ASCO[16]</td>
<td>19</td>
<td>16</td>
<td>0.84</td>
<td>0.64</td>
<td>0.73</td>
</tr>
<tr>
<td>SSMO[26]</td>
<td>19</td>
<td>16</td>
<td>0.84</td>
<td>0.64</td>
<td>0.73</td>
</tr>
<tr>
<td>EOMT[52]</td>
<td>21</td>
<td>18</td>
<td>0.86</td>
<td>0.72</td>
<td>0.78</td>
</tr>
<tr>
<td>CACOM[30]</td>
<td>20</td>
<td>18</td>
<td>0.9</td>
<td>0.72</td>
<td>0.8</td>
</tr>
<tr>
<td>SIMTO</td>
<td>23</td>
<td>22</td>
<td>0.96</td>
<td>0.88</td>
<td>0.92</td>
</tr>
</tbody>
</table>
Figure 4.3: Completeness wise comparison of results with respect to first test case

Figure 4.4: Correctness wise comparison of results with respect to first test case
4.5.4.2 Evaluating Performance with Second Test Case

Sample input pairs: 40

Pairs of similar concepts (expected): 22

Similar pairs (out of 22) with different terms: 5

With respect to test case 2, the results from SIMTO and from some existing techniques are compared in Table 4.10, with respect to their completeness, correctness and overall quality. The graphical representation of comparison is also given in the Figures 4.6, 4.7 and 4.8 respectively. It has observed that when the number of similar pairs having different names, decrease, the completeness of results increases. Furthermore, the result of proposed technique, with respect to completeness is better than the results of existing techniques.
Table 4.10: Results of different techniques with second test case

<table>
<thead>
<tr>
<th>Technique</th>
<th>Pairs of similar concepts returned</th>
<th>Correct pairs found</th>
<th>Completeness</th>
<th>Correctness</th>
<th>Overall Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSBO[61]</td>
<td>18</td>
<td>17</td>
<td>0.94</td>
<td>0.77</td>
<td>0.85</td>
</tr>
<tr>
<td>SEMC[19]</td>
<td>20</td>
<td>18</td>
<td>0.9</td>
<td>0.81</td>
<td>0.85</td>
</tr>
<tr>
<td>LOM[23]</td>
<td>21</td>
<td>19</td>
<td>0.91</td>
<td>0.86</td>
<td>0.84</td>
</tr>
<tr>
<td>HCOME[15]</td>
<td>17</td>
<td>16</td>
<td>0.94</td>
<td>0.72</td>
<td>0.81</td>
</tr>
<tr>
<td>RTHO[24]</td>
<td>21</td>
<td>19</td>
<td>0.91</td>
<td>0.86</td>
<td>0.88</td>
</tr>
<tr>
<td>ASCO[16]</td>
<td>17</td>
<td>16</td>
<td>0.94</td>
<td>0.72</td>
<td>0.82</td>
</tr>
<tr>
<td>SSMO[26]</td>
<td>17</td>
<td>16</td>
<td>0.94</td>
<td>0.72</td>
<td>0.82</td>
</tr>
<tr>
<td>EOMT[52]</td>
<td>21</td>
<td>19</td>
<td>0.91</td>
<td>0.86</td>
<td>0.88</td>
</tr>
<tr>
<td>CACOM[30]</td>
<td>20</td>
<td>18</td>
<td>0.9</td>
<td>0.81</td>
<td>0.85</td>
</tr>
<tr>
<td>SIMTO</td>
<td>21</td>
<td>21</td>
<td>1</td>
<td>0.95</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Figure 4.6: Completeness wise comparison of results with respect to second test case
Figure 4.7: Correctness wise comparison of results with respect to second test case

Figure 4.8: Overall quality wise comparison of results with respect to second test case
4.5.4.3 Evaluating Performance with Third Test Case

Sample input pairs: 28

Pairs of similar concepts (expected): 12

Similar pairs (out of 12) with different terms: 2

With respect to test case 3, the results from SIMTO and from some existing techniques are compared in Table 4.11, with respect to their completeness, correctness and overall quality. It has observed that when the number of similar pairs having different names, decreases, the completeness of results increases. Furthermore, the result of proposed technique, with respect to completeness, correctness and overall quality is better than the results of existing techniques.

Table 4.11: Results of different techniques with third test case

<table>
<thead>
<tr>
<th>Technique</th>
<th>Pairs of similar concepts returned</th>
<th>Correct pairs found</th>
<th>Completeness</th>
<th>Correctness</th>
<th>Overall Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSBO[61]</td>
<td>14</td>
<td>10</td>
<td>0.71</td>
<td>0.83</td>
<td>0.77</td>
</tr>
<tr>
<td>SEMC[19]</td>
<td>14</td>
<td>12</td>
<td>0.86</td>
<td>1</td>
<td>0.93</td>
</tr>
<tr>
<td>LOM[23]</td>
<td>14</td>
<td>12</td>
<td>0.86</td>
<td>1</td>
<td>0.93</td>
</tr>
<tr>
<td>HCON[15]</td>
<td>15</td>
<td>10</td>
<td>0.67</td>
<td>0.83</td>
<td>0.74</td>
</tr>
<tr>
<td>RTHO[24]</td>
<td>14</td>
<td>12</td>
<td>0.86</td>
<td>1</td>
<td>0.93</td>
</tr>
<tr>
<td>ASCO[16]</td>
<td>15</td>
<td>10</td>
<td>0.67</td>
<td>0.83</td>
<td>0.74</td>
</tr>
<tr>
<td>SSMO[26]</td>
<td>15</td>
<td>10</td>
<td>0.67</td>
<td>0.83</td>
<td>0.74</td>
</tr>
<tr>
<td>EOMT[52]</td>
<td>14</td>
<td>12</td>
<td>0.86</td>
<td>1</td>
<td>0.93</td>
</tr>
<tr>
<td>CACOM[30]</td>
<td>14</td>
<td>12</td>
<td>0.86</td>
<td>1</td>
<td>0.93</td>
</tr>
<tr>
<td>SIMTO</td>
<td>12</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
The graphical representation of comparison is also given in the Figures 4.9, 4.10 and 4.11 respectively. An improvement in result of proposed technique, with respect to completeness, correctness and overall quality, is realized in comparison.
Figure 4.11: Overall quality wise comparison of results with respect to third test case

4.5.4.4 Evaluating Performance with Fourth Test Case

Sample input pairs: 25

Pairs of similar concepts (expected): 15

Similar pairs (out of 15) with different terms: 12

Table 4.12: Results of different techniques with fourth test case

<table>
<thead>
<tr>
<th>Technique</th>
<th>Pairs of similar concepts returned</th>
<th>Correct pairs found</th>
<th>Completeness</th>
<th>Correctness</th>
<th>Overall Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSBO[61]</td>
<td>5</td>
<td>7</td>
<td>0.71</td>
<td>0.33</td>
<td>0.45</td>
</tr>
<tr>
<td>SEMC[19]</td>
<td>7</td>
<td>9</td>
<td>0.77</td>
<td>0.47</td>
<td>0.58</td>
</tr>
<tr>
<td>LOM[23]</td>
<td>8</td>
<td>9</td>
<td>0.88</td>
<td>0.53</td>
<td>0.66</td>
</tr>
<tr>
<td>HCOME[15]</td>
<td>6</td>
<td>10</td>
<td>0.6</td>
<td>0.4</td>
<td>0.48</td>
</tr>
<tr>
<td>RTHO[24]</td>
<td>8</td>
<td>9</td>
<td>0.88</td>
<td>0.53</td>
<td>0.66</td>
</tr>
<tr>
<td>ASCO[16]</td>
<td>6</td>
<td>10</td>
<td>0.6</td>
<td>0.4</td>
<td>0.48</td>
</tr>
<tr>
<td>SSMO[26]</td>
<td>6</td>
<td>10</td>
<td>0.6</td>
<td>0.4</td>
<td>0.48</td>
</tr>
<tr>
<td>EOMT[52]</td>
<td>8</td>
<td>9</td>
<td>0.88</td>
<td>0.53</td>
<td>0.66</td>
</tr>
<tr>
<td>CACOM[30]</td>
<td>7</td>
<td>8</td>
<td>0.88</td>
<td>0.47</td>
<td>0.61</td>
</tr>
<tr>
<td>SIMTO</td>
<td>14</td>
<td>16</td>
<td>0.93</td>
<td>1</td>
<td>0.96</td>
</tr>
</tbody>
</table>
With respect to test case 4, the results from SIMTO and from some existing techniques are compared in Table 4.12, with respect to their completeness, correctness and overall quality. The graphical representation of comparison is also given in the Figures 4.12, 4.13 and 4.14 respectively. It has observed that when the number of similar pairs having different names increase, the completeness of results decreases. There is a considerable decrease in correctness of results from existing techniques particularly the techniques excluding the linguistic similarity of terms. Furthermore, the overall qualities of results are also badly affected. However, the result of proposed technique, with respect to completeness is better than the results of existing techniques.

Figure 4.12: Completeness wise comparison of results with respect to fourth test case
Figure 4.13: Correctness wise comparison of results with respect to fourth test case

Figure 4.14: Overall quality wise comparison of results with respect to fourth test case
The comparisons between results from some current techniques and from proposed technique (up to 2nd level of similarity) are shown in Figures 4.3-4.14. Furthermore, a comparison between results from proposed technique with the new criterion (i.e. role-based similarity) and expected results has also made in Figure 4.14. We examined more than ten ontologies of different software houses for evaluating the new criterion of proposed technique. The results are verified by respective domain experts and are declared satisfactory.

![Graph showing role-based similarity](image)

**Figure 4.15: Role-based similarity: expected results vs. produced results**

### 4.6 Discussion about Overall Performance

Through analysis of results, we come to some conclusions which are described next.

(i) Through string-based approaches, as used in existing techniques, some dissimilar pairs are declared similar pairs, which decrease the completeness and overall quality of results.

(ii) Overall results of existing techniques heavily rely on the heterogeneity of terms as shown in Table 4.14.; higher the number of concepts represented with different terms, lower the completeness and overall quality of results will be.
Table 4.13: Test cases along with status of WordNet and Domain Vocabulary

<table>
<thead>
<tr>
<th>TestCase</th>
<th>Input Pairs</th>
<th>Similar Pairs (expected)</th>
<th>Similar Pairs (with different terms)</th>
<th>Similar pairs with different terms supported by WordNet</th>
<th>Similar pairs with different terms supported by DV</th>
<th>Similar Pairs with different terms not supported by DV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37</td>
<td>25</td>
<td>10</td>
<td>3</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>22</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>28</td>
<td>12</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>15</td>
<td>12</td>
<td>2</td>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.14: Comparison of results related to Heterogeneous Pairs of Similar Concepts (HPoSCs)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>3</td>
<td>7</td>
<td>0.30</td>
<td>0.70</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>0.20</td>
<td>0.80</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>2</td>
<td>11</td>
<td>0.17</td>
<td>0.92</td>
</tr>
</tbody>
</table>

(iii) Although in existing techniques, the WordNet has good support for matching linguistics semantics of terms, but linguistic semantics of several domain-specific terms particularly the abbreviated terms and composite terms are not supported by WordNet. Also, due to same linguistic semantic of different terms, some unnecessary pairs are identified, which reduce the completeness and overall quality of result.

(iv) The proposed technique produces better results than the results of existing techniques, such as shown in Tables 4.9-4.12 and in the Figures 4.3-4.13.

(v) Although the proposed technique is also dependent on domain-specific vocabulary, but we empirically observed that domain-specific vocabulary is much better than WordNet.

(vi) The proposed technique may produce 100 percent complete and correct result, but it is not always true, due to absence of some new concepts in domain-specific vocabulary as shown in Table 4.13.

(vii) We manually populate domain-specific vocabulary and it is some time consuming task. Domain vocabulary is not a static, it is updated dynamically.
Chapter 5

CONCLUSION AND FUTURE WORK

In this thesis a semi-automatic, integrated and layered technique for identification and measurement of similarity between concepts of two ontologies has been presented. The proposed technique borrows the innovative theme of the semantic web that measures similarity between two concepts based on their explicit semantics and granularities. The proposed technique is not only helpful in different ontology integration operations such as merging, mapping, alignment and querying but also for engineering new ontologies.

5.1 Conclusion

Identification and measurement of similarity between the ontologies is a mandatory pre-requisite of various reuse operations of ontologies such as merging, mapping and alignment. It is also a mandatory requirement for engineering new ontologies by assembling exiting ontologies or components of ontologies.

Although the proposed similarity identification technique uses, as core, the innovative ideas of semantic web however essential modifications related to the issues and trends specific to the similarity between concepts of ontologies has been made. The proposed technique upgrades similarity measurement criteria, from terms to concepts, from linguistic semantic to explicit semantic and from all taxonomic characteristics of concepts to mandatory and optional characteristics. In addition we introduced the concepts of similarity levels: primary similarity or 1st level similarity; contextual similarity or 2nd level similarity and the role-based similarity or 3rd level similarity.
In the primary similarity, the actual concepts have been focused rather than terms used to name them. The super-concepts have been taken into consideration while identifying contextual similarity and the similarity of siblings and sub-concepts have been relaxed. The third level of similarity has been identified using a new criterion. Furthermore, a layered strategy has been adopted in proposed technique to enable matching process more efficient. New criteria have been introduced for measuring the degree of similarity and the semantic relations between concepts.

We have implemented the proposed technique using Java language. For loading and parsing web ontologies, OWL API [70, 82, 84] has been used. The technique has been validated through two case studies. We have exercised different test-scenarios to validate the proposed idea. The correctness, completeness and overall quality of results of proposed technique, has confirmed the achievement of said objectives i.e., none of the candidate pair of similar concepts have been omitted from result; the pairs possessing role-based similarity have been correctly identified; and the working of layered strategy for similarly identification has been proved.

We conclude the research result as follows:

- The techniques used for identifying and measuring of similarity between database schemas or between XML schemas are not well suited for identifying and measuring of similarity between ontologies schemas.
- The role of domain-specific vocabulary is vital in identification and measurement of similarity between ontologies.
• The identification of primary similarity is the prerequisite for the contextual similarity identification process whereas the identification of contextual similarity is the prerequisite for the role-based similarity identification.

• For a pair of concepts, the degree of similarity and semantic relation are complements to each others.

• It is difficult to get hundred percent correct and complete results due to the lack of standardization in the use of terminologies for concepts and their roles.

5.2 Recommendations for Further Research

As discussed above, the similarity identification is a core and prerequisite phase for ontologies integration operations. In addition, this phase is also required for ontologies engineering through reuse of ontologies. Some ontologies integration operations such as merging, mapping, aligning have also been discussed. We plan to work on design and development of methodologies for these ontologies reuse and integration operations. Abstract work plan related to the ontologies reuse and integration operations is as follows:

(i) **Ontology Mapping Process:** Before copying or transforming data from one ontology to the other ontology, the mapping between similar concepts of those ontologies is established. The proposed technique can be used to determine similarity between concepts to enable this mapping.

(ii) **Ontology Merging:** In some scenarios, when certain domains are merged, then their corresponding ontologies are needed to be merged. Before merging ontologies, the corresponding concepts of those ontologies have to be identified, whereas the proposed technique can be used to determine corresponding concepts.

(iii) **Ontology Aligning:** When data and knowledge need to be shared between domains, then their corresponding ontologies need to be aligned. Before aligning the ontologies,
similar concepts of these ontologies have to be identified. The proposed technique can be used to determine the similar concepts.

(iv) **Ontology Engineering through Ontology-reuse**: While engineering new ontologies, it is preferred to use, related existing ontologies or even segments of the existing ontologies, as a parts of new ontology instead of re-engineering those parts or those sub-ontologies. The proposed technique can be used to identify the required existing components while assembling the existing ontologies.

(v) **Knowledge Acquisition**: The proposed technique can be used in the design of knowledge acquisition agents, for the domain of semantic web.

(vi) **Semantic Search Engines**: To search the relevant semantic web applications, their corresponding ontologies are searched. Before retrieval, the similarity identification needs to be performed between their corresponding ontologies. In order to design and develop such search engines, the proposed technique can become a useful ingredient for those search engines.

(vii) **Ontology Selection and Modification**: The proposed technique can be used to find relevant ontology for a certain domain and sometime the ontologies need to be readjusted to meet the certain requirements of new domains. Before modification in the ontology, it is required to know whether some important concepts are missing! How many irrelevant concepts are there? Is there a need to replace two different concepts by single concept with some modification in its definition? The proposed similarity identification and measurement technique can provide solutions for all such types of questions.
REFERENCES


[37] Plessers, P., Casteleyn, S., De Troyer, O. 2005 "SW Development with WSDM", In 5th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2005), Galway, Ireland.


[40] Semantic Web Case Studies and Use Cases.


[64] Web Ontology Language (OWL); http://www.w3.org/2004/OWL/

[65] The DARPA Agent Markup Language Homepage: http://www.daml.org/

[66] Zhang, Proscovia Olango, “Populating an Ontology Using Wikipedia’s Taxonomy To Know More” Information Science, University of Groningen.


[87] RDF Vocabulary Description Language 1.0: RDF Schema;
http://www.w3.org/TR/rdf-schema/

[88] W3C Semantic Web Activity; http://www.w3.org/2001/sw/


# APPENDIX-I
(Ontology of a person’s family)

<!-- Axioms: 35 -->

<Ontology
xmlns="http://www.w3.org/2006/12/owl11-xml#"
xml:base="http://www.w3.org/2006/12/owl11-xml#"
xmlns:owl11="http://www.w3.org/2006/12/owl11#"
xmlns:owl11xml="http://www.w3.org/2006/12/owl11-xml#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#"

URI="http://www.semanticweb.org/ontologies/2009/1/Ontology/personFamily.owl">

<DataPropertyAssertion>
  <DataProperty URI="&personFamily;hasName"/>
  <Individual URI="&personFamily;2"/>
  <Constant>Aasia</Constant>
</DataPropertyAssertion>

<ObjectProperty
URI="&personFamily;isSisterOf"/>
  <OWLClass URI="&personFamily;MalePerson"/>
  <ObjectPropertyAssertion>
  <ObjectProperty URI="&personFamily;isMotherOf"/>
  <Individual URI="&personFamily;2"/>
  <Individual URI="&personFamily;3"/>
</ObjectPropertyAssertion>

<DataPropertyDomain>
  <DataProperty URI="&personFamily;hasName"/>
  <OWLClass URI="&personFamily;Person"/>
</DataPropertyDomain>

<ObjectPropertyAssertion>
  <ObjectProperty URI="&personFamily;isBrotherOf"/>
  <Individual URI="&personFamily;5"/>
  <Individual URI="&personFamily;2"/>
</ObjectPropertyAssertion>

<DataPropertyAssertion>
  <DataProperty URI="&personFamily;hasName"/>
  <Individual URI="&personFamily;3"/>
  <Constant>Humza</Constant>
</DataPropertyAssertion>

<DataPropertyAssertion>
  <DataProperty URI="&personFamily;hasName"/>
  <Individual URI="&personFamily;4"/>
  <Constant>Fozia</Constant>
</DataPropertyAssertion>
<ObjectProperty
URI="&personFamily;isFatherOf"/>
<Individual
URI="&personFamily;1"/>
</ObjectPropertyAssertion>
<ObjectPropertyDomain>
<ObjectProperty
URI="&personFamily;isSisterOf"/>
<OWLClass
URI="&personFamily;FemalePerson"/>
</ObjectPropertyDomain>
<DataPropertyAssertion>
<DataProperty
URI="&personFamily;hasName"/>
<Individual
URI="&personFamily;9"/>
<Constant>Murym</Constant>
</DataPropertyAssertion>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty
URI="&personFamily;isFatherOf"/>
<Individual
URI="&personFamily;8"/>
<Individual
URI="&personFamily;1"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty
URI="&personFamily;isMotherOf"/>
<Individual
URI="&personFamily;9"/>
<Individual
URI="&personFamily;1"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty
URI="&personFamily;isFatherOf"/>
<Individual
URI="&personFamily;6"/>
</ObjectPropertyAssertion>
<ObjectPropertyDomain>
<ObjectProperty
URI="&personFamily;isMotherOf"/>
<OWLClass
URI="&personFamily;FemalePerson"/>
</ObjectPropertyDomain>
<DataPropertyAssertion>
<DataProperty
URI="&personFamily;hasName"/>
<Individual
URI="&personFamily;6"/>
<Constant>Zia</Constant>
</DataPropertyAssertion>
</ObjectPropertyAssertion>
<ObjectPropertyRange>
<ObjectProperty
URI="&personFamily;isMotherOf"/>
<OWLClass
URI="&personFamily;FemalePerson"/>
</ObjectPropertyRange>
</SubClassOf>
</ObjectPropertyAssertion>
<ObjectProperty
URI="&personFamily;isFatherOf"/>
<Individual
URI="&personFamily;2"/>
</ObjectPropertyAssertion>
<ObjectPropertyDomain>
<ObjectProperty
URI="&personFamily;isMotherOf"/>
<OWLClass
URI="&personFamily;FemalePerson"/>
</ObjectPropertyDomain>
<DataPropertyAssertion>
<DataProperty
URI="&personFamily;hasName"/>
<Individual
URI="&personFamily;2"/>
<Constant>Murym</Constant>
</DataPropertyAssertion>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
<ObjectProperty
URI="&personFamily;isSisterOf"/>
<Individual
URI="&personFamily;2"/>
</ObjectPropertyAssertion>
<ObjectPropertyRange>
<ObjectProperty
URI="&personFamily;isMotherOf"/>
<OWLClass
URI="&personFamily;FemalePerson"/>
</ObjectPropertyRange>
</SubClassOf>
</ObjectPropertyAssertion>
<ObjectProperty
URI="&personFamily;isFatherOf"/>
<Individual
URI="&personFamily;4"/>
</ObjectPropertyAssertion>
<ObjectPropertyDomain>
<ObjectProperty
URI="&personFamily;isSisterOf"/>
<OWLClass
URI="&personFamily;FemalePerson"/>
</ObjectPropertyDomain>
<DataPropertyAssertion>
<DataProperty
URI="&personFamily;hasName"/>
<Individual
URI="&personFamily;4"/>
<Constant>Zia</Constant>
</DataPropertyAssertion>
</ObjectPropertyAssertion>
<ObjectPropertyRange>
<ObjectProperty
URI="&personFamily;isMotherOf"/>
<OWLClass
URI="&personFamily;FemalePerson"/>
</ObjectPropertyRange>
</SubClassOf>
</ObjectPropertyAssertion>
<ObjectProperty
URI="&personFamily;isFatherOf"/>
<Individual
URI="&personFamily;4"/>
</ObjectPropertyAssertion>
<ObjectPropertyDomain>
<ObjectProperty
URI="&personFamily;isMotherOf"/>
<OWLClass
URI="&personFamily;FemalePerson"/>
</ObjectPropertyDomain>
<DataPropertyAssertion>
<DataProperty
URI="&personFamily;hasName"/>
<Individual
URI="&personFamily;4"/>
<Constant>Zia</Constant>
</DataPropertyAssertion>
</ObjectPropertyAssertion>
<ObjectPropertyRange>
<ObjectProperty
URI="&personFamily;isMotherOf"/>
<OWLClass
URI="&personFamily;FemalePerson"/>
</ObjectPropertyRange>
</SubClassOf>
</ObjectPropertyAssertion>
<OWLClass URI="&personFamily;Person"/>
</SubClassOf>
<ObjectPropertyRange>
<ObjectProperty URI="&personFamily;isFatherOf"/>
<OWLClass URI="&personFamily;FemalePerson"/>
</ObjectPropertyRange>
<ObjectPropertyAssertion>
<ObjectProperty URI="&personFamily;isMotherOf"/>
<Individual URI="&personFamily;7"/>
</Individual URI="&personFamily;2"/>
<ObjectPropertyAssertion>
<ObjectProperty URI="&personFamily;hasName"/>
<Individual URI="&personFamily;7"/>
<Constant>TusNeem</Constant>
</DataPropertyAssertion>
<ObjectPropertyDomain>
<ObjectProperty URI="&personFamily;isFatherOf"/>
<OWLClass URI="&personFamily;MalePerson"/>
</ObjectPropertyDomain>
<DataPropertyRange>
<DataProperty URI="&personFamily;hasName"/>
<Datatype URI="&xsd;string"/>
</DataPropertyRange>
<DataPropertyAssertion>
<Individual URI="&personFamily;5"/>
<Constant>Athar</Constant>
</DataPropertyAssertion>
</Declaration>
<ObjectPropertyRange>
<ObjectProperty URI="&personFamily;isBrotherOf"/>
<OWLClass URI="&personFamily;MalePerson"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
<ObjectProperty URI="&personFamily;isBrotherOf"/>
<OWLClass URI="&personFamily;FemalePerson"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
<ObjectProperty URI="&personFamily;isFatherOf"/>
<OWLClass URI="&personFamily;MalePerson"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
<ObjectProperty URI="&personFamily;isFatherOf"/>
<OWLClass URI="&personFamily;MalePerson"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
<ObjectProperty URI="&personFamily;isFatherOf"/>
<OWLClass URI="&personFamily;MalePerson"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
<ObjectProperty URI="&personFamily;isFatherOf"/>
<OWLClass URI="&personFamily;MalePerson"/>
</ObjectPropertyRange>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
</ObjectPropertyAssertion>
<ObjectPropertyURI="#"/>
### APPENDIX-II
(A Subset of Roles from Software Development Organization)

<table>
<thead>
<tr>
<th>Id</th>
<th>Label/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1.</td>
<td>Handling all administrative / accounting matters related to Irvine office.</td>
</tr>
<tr>
<td>r2.</td>
<td>Accurately scheduling appointments</td>
</tr>
<tr>
<td>r3.</td>
<td>Acting as a liaison with the customer on regular bases.</td>
</tr>
<tr>
<td>r4.</td>
<td>Administering complicated networks at multiple locations.</td>
</tr>
<tr>
<td>r5.</td>
<td>Allocating a configuration management team to the project.</td>
</tr>
<tr>
<td>r6.</td>
<td>Allocating database resources to projects.</td>
</tr>
<tr>
<td>r7.</td>
<td>Analyzing and troubleshooting complex computer systems and network issues, identifying the reasons for network and network device problems, failures and malfunctions and developing optimal solutions.</td>
</tr>
<tr>
<td>r8.</td>
<td>Analyzing organizational metrics and devising relevant process improvements.</td>
</tr>
<tr>
<td>r9.</td>
<td>Analyzing products and recommending the use of new products and services, upgrade and improvements to the senior management.</td>
</tr>
<tr>
<td>r10.</td>
<td>Analyzing project performance on periodic bases.</td>
</tr>
<tr>
<td>r11.</td>
<td>Analyze Hardware Requirements</td>
</tr>
<tr>
<td>r12.</td>
<td>Analyze Software Requirements</td>
</tr>
<tr>
<td>r13.</td>
<td>Analyze Functional Requirements</td>
</tr>
<tr>
<td>r14.</td>
<td>Analyze Non Functional Requirements</td>
</tr>
<tr>
<td>r15.</td>
<td>Analyze Cost Benefit</td>
</tr>
<tr>
<td>r16.</td>
<td>Answering the main telephone line (operator)</td>
</tr>
<tr>
<td>r17.</td>
<td>Approving SQA, Test and CM and other functional plans for the project.</td>
</tr>
<tr>
<td>r18.</td>
<td>Assigning the team for project quality assurance.</td>
</tr>
<tr>
<td>r19.</td>
<td>Assist Chief Financial Officer in organizational financial activities.</td>
</tr>
<tr>
<td>r20.</td>
<td>Assisting the database administration team.</td>
</tr>
<tr>
<td>r21.</td>
<td>Automation of build procedures, wherever possible.</td>
</tr>
<tr>
<td>r22.</td>
<td>Be a part of preparing, reviewing and finalizing application design.</td>
</tr>
<tr>
<td>r23.</td>
<td>Be a part of the preparation of the Bi-Traceability matrix.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>r24</td>
<td>Building and maintaining subject matter expertise of the firm's core system offerings.</td>
</tr>
<tr>
<td>r25</td>
<td>Checking in/out documents to/from the library.</td>
</tr>
<tr>
<td>r26</td>
<td>Clean utensils.</td>
</tr>
<tr>
<td>r27</td>
<td>Clearly communicating QA results, concerns, and project status to senior management and other stakeholders in a timely manner.</td>
</tr>
<tr>
<td>r28</td>
<td>Communicating the results of SQA activities to concerned teams.</td>
</tr>
<tr>
<td>r29</td>
<td>Compiling source code/linking/building.</td>
</tr>
<tr>
<td>r30</td>
<td>Conducting reviews of project artifacts, typically project defined processes, to make them compliant with documented organizational processes and policies.</td>
</tr>
<tr>
<td>r31</td>
<td>Conferring with developers about quality assurance of new products in design and existing products to rectify problems.</td>
</tr>
<tr>
<td>r33</td>
<td>Controlling configured software.</td>
</tr>
<tr>
<td>r34</td>
<td>Coordinating new employee starts and new hire paperwork.</td>
</tr>
<tr>
<td>r35</td>
<td>Coordinating with project resources closely, on periodic bases, and tracking their activities.</td>
</tr>
<tr>
<td>r36</td>
<td>Creating and updating project baseline(s).</td>
</tr>
<tr>
<td>r37</td>
<td>Creating approved baselines from test baselines.</td>
</tr>
<tr>
<td>r38</td>
<td>Creating developmental baselines.</td>
</tr>
<tr>
<td>r39</td>
<td>Creating final developmental baselines.</td>
</tr>
<tr>
<td>r40</td>
<td>Creating final release packages.</td>
</tr>
<tr>
<td>r41</td>
<td>Creating payment requests for employees traveling abroad.</td>
</tr>
<tr>
<td>r42</td>
<td>Creating purchase requests.</td>
</tr>
<tr>
<td>r43</td>
<td>Creating test baselines.</td>
</tr>
<tr>
<td>r44</td>
<td>Defining user profiles.</td>
</tr>
<tr>
<td>r45</td>
<td>Design Database</td>
</tr>
<tr>
<td>r46</td>
<td>Design Algorithms</td>
</tr>
<tr>
<td>r47</td>
<td>Design Reports</td>
</tr>
<tr>
<td>r48.</td>
<td>Design Input Screens</td>
</tr>
<tr>
<td>------</td>
<td>---------------------</td>
</tr>
<tr>
<td>r49.</td>
<td>Design Structure</td>
</tr>
<tr>
<td>r50.</td>
<td>Design Graphics</td>
</tr>
<tr>
<td>r51.</td>
<td>Design Web Pages</td>
</tr>
<tr>
<td>r52.</td>
<td>Deploying applications in virtual machine environments.</td>
</tr>
<tr>
<td>r53.</td>
<td>Deploying builds for executing cycles of testing.</td>
</tr>
<tr>
<td>r54.</td>
<td>Developing a culture of quality integration within software projects right from the beginning, and within the entire organization.</td>
</tr>
<tr>
<td>r55.</td>
<td>Developing a test plan for the project.</td>
</tr>
<tr>
<td>r56.</td>
<td>Developing and executing performance, stress, stability, and benchmarking test strategies that ensure our software meets or exceeds performance targets.</td>
</tr>
<tr>
<td>r57.</td>
<td>Developing conceptual frameworks and applying state-of-the-art technology to the design and management of network infrastructures.</td>
</tr>
<tr>
<td>r58.</td>
<td>Developing physical configuration audit schedules and ensuring their successful execution.</td>
</tr>
<tr>
<td>r59.</td>
<td>Developing the testing approach/strategy and methodology for all of the firm's software offerings.</td>
</tr>
<tr>
<td>r60.</td>
<td>Devising database schemas for various projects.</td>
</tr>
<tr>
<td>r61.</td>
<td>Direct incoming calls in a timely and professional manner</td>
</tr>
<tr>
<td>r62.</td>
<td>Directing the implementation of reported deficiencies against CM activities or products.</td>
</tr>
<tr>
<td>r63.</td>
<td>Documenting project backup and archiving strategies in SCMP.</td>
</tr>
<tr>
<td>r64.</td>
<td>Documenting project repository and directory structures in the SCMP.</td>
</tr>
<tr>
<td>r65.</td>
<td>Documenting test cases for engineering development models.</td>
</tr>
<tr>
<td>r66.</td>
<td>Ensuring an overall healthy, safe and clean office environment.</td>
</tr>
<tr>
<td>r67.</td>
<td>Ensuring Inter-group coordination.</td>
</tr>
<tr>
<td>r68.</td>
<td>Ensuring that all project issues and risks are identified at an earlier stage and their resolution is done effectively.</td>
</tr>
<tr>
<td>r69.</td>
<td>Ensuring that SQA activities are carried out as planned.</td>
</tr>
<tr>
<td>r70.</td>
<td>Ensuring that the configuration management processes are being followed</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>r71.</td>
<td>Ensuring the implementation and maintenance of CM processes in accordance with standard organizational definitions.</td>
</tr>
<tr>
<td>r72.</td>
<td>Ensuring the preparation and execution of project schedules.</td>
</tr>
<tr>
<td>r73.</td>
<td>Ensuring the security of project databases.</td>
</tr>
<tr>
<td>r74.</td>
<td>Ensuring the successful execution of all project plans.</td>
</tr>
<tr>
<td>r75.</td>
<td>Ensuring the successful execution of software quality assurance audit plans and the analysis of results obtained for future improvements.</td>
</tr>
<tr>
<td>r76.</td>
<td>Ensuring the successful resource availability and management for projects.</td>
</tr>
<tr>
<td>r77.</td>
<td>Establishing and implementing policies and procedures for LAN/WAN usage throughout the organization.</td>
</tr>
<tr>
<td>r78.</td>
<td>Establishing operational objectives for department or functional area and participate with other managers to establish group objectives.</td>
</tr>
<tr>
<td>r79.</td>
<td>Establishing SCCB and SCRB for projects.</td>
</tr>
<tr>
<td>r80.</td>
<td>Evaluating quality assurance tools and facilities and identifying those which are required or which may add value within the organization.</td>
</tr>
<tr>
<td>r81.</td>
<td>Executing the plan and conducting reviews to make sure that project teams are working in accordance with defined project software processes and also to make sure that the artifacts are produced accordingly.</td>
</tr>
<tr>
<td>r82.</td>
<td>Executing the test plan, ensuring verification and validation of end product, to be delivered to the client.</td>
</tr>
<tr>
<td>r83.</td>
<td>Faxing, mailing, shipping and other general administrative duties.</td>
</tr>
<tr>
<td>r84.</td>
<td>Filing student files, transcripts and other records</td>
</tr>
<tr>
<td>r85.</td>
<td>Final product inspection and delivery of product.</td>
</tr>
<tr>
<td>r86.</td>
<td>Finalizing user requirements after reviewing them.</td>
</tr>
<tr>
<td>r87.</td>
<td>Follow up and correspondence with employees, potential employees, and clients.</td>
</tr>
<tr>
<td>r88.</td>
<td>Forwarding changes and developmental baselines to QA.</td>
</tr>
<tr>
<td>r89.</td>
<td>Forwarding copies of preliminary release packages to project management and QA.</td>
</tr>
<tr>
<td>r90.</td>
<td>Giving instructions to implement reported deficiencies against CM activities or</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>products.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>r91.</strong> Greeting visitors in a prompt and professional manner</td>
<td></td>
</tr>
<tr>
<td><strong>r92.</strong> Handling and maintaining the company store.</td>
<td></td>
</tr>
<tr>
<td><strong>r93.</strong> Heading meetings concerning administration activities.</td>
<td></td>
</tr>
<tr>
<td><strong>r94.</strong> Helping the PM to define, document and implement processes for their project.</td>
<td></td>
</tr>
<tr>
<td><strong>r95.</strong> Identification of project based SCM tool(s).</td>
<td></td>
</tr>
<tr>
<td><strong>r96.</strong> Identifying and correcting differences between final released items and preliminary versions.</td>
<td></td>
</tr>
<tr>
<td><strong>r97.</strong> Identifying and correcting the differences between final released items and preliminary versions.</td>
<td></td>
</tr>
<tr>
<td><strong>r98.</strong> Identifying and tracking deviations in the activities of the team in accordance with the set standards, procedures &amp; plans.</td>
<td></td>
</tr>
<tr>
<td><strong>r99.</strong> Identifying items for final configuration.</td>
<td></td>
</tr>
<tr>
<td><strong>r100.</strong> Identifying key learning from previous releases and building QA plans to address these learnings.</td>
<td></td>
</tr>
<tr>
<td><strong>r101.</strong> Identifying organizational process needs and further developing and implementing them.</td>
<td></td>
</tr>
<tr>
<td><strong>r102.</strong> Identifying project based SCM items and documenting them.</td>
<td></td>
</tr>
<tr>
<td><strong>r103.</strong> Identifying project specific trainings and ensuring their accomplishments.</td>
<td></td>
</tr>
<tr>
<td><strong>r104.</strong> Implement Database</td>
<td></td>
</tr>
<tr>
<td><strong>r105.</strong> Implementing Algorithm</td>
<td></td>
</tr>
<tr>
<td><strong>r106.</strong> Implementing Reports</td>
<td></td>
</tr>
<tr>
<td><strong>r107.</strong> Implementing GUI</td>
<td></td>
</tr>
<tr>
<td><strong>r108.</strong> Implementing Structure of software</td>
<td></td>
</tr>
<tr>
<td><strong>r109.</strong> Implementing and maintaining CM processes in accordance with organizational definitions.</td>
<td></td>
</tr>
<tr>
<td><strong>r110.</strong> Implementing code as per the impact analysis description.</td>
<td></td>
</tr>
<tr>
<td><strong>r111.</strong> Implementing code as per the impact analysis details.</td>
<td></td>
</tr>
<tr>
<td><strong>r112.</strong> Implementing scalable, secure, and stable network solutions as directed by the lead.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>r113.</td>
<td>Implementing database backup design.</td>
</tr>
<tr>
<td>r114.</td>
<td>Installing, configuring, maintaining and managing the operations of complex network systems to achieve optimal technical performance and end user support.</td>
</tr>
<tr>
<td>r115.</td>
<td>Integrating the functional plans of project into the project management plan and finalizing all project based planning.</td>
</tr>
<tr>
<td>r116.</td>
<td>Interacting with and reporting project status to Senior Management.</td>
</tr>
<tr>
<td>r117.</td>
<td>Internal and external release management.</td>
</tr>
<tr>
<td>r118.</td>
<td>Inventory management (related stationery, utility and general items)</td>
</tr>
<tr>
<td>r119.</td>
<td>Inventory management (related Stationery, utility and general items).</td>
</tr>
<tr>
<td>r120.</td>
<td>Involvement in preparing, reviewing and finalizing application design.</td>
</tr>
<tr>
<td>r121.</td>
<td>Issue cheques to parties and maintain their records.</td>
</tr>
<tr>
<td>r122.</td>
<td>Keep and record all expense vouchers in Peachtree</td>
</tr>
<tr>
<td>r123.</td>
<td>Keeping all affected groups informed.</td>
</tr>
<tr>
<td>r124.</td>
<td>Keeping the VPN up with onsite offices.</td>
</tr>
<tr>
<td>r125.</td>
<td>Logging new documents into the library and database.</td>
</tr>
<tr>
<td>r126.</td>
<td>Mailing requested materials to prospective clients, suppliers</td>
</tr>
<tr>
<td>r127.</td>
<td>Maintain company store as directed.</td>
</tr>
<tr>
<td>r128.</td>
<td>Maintaining library records (books, CD’s, videos, etc.)</td>
</tr>
<tr>
<td>r129.</td>
<td>Maintaining and monitoring the configuration status of all project configuration items.</td>
</tr>
<tr>
<td>r130.</td>
<td>Maintaining database schemas for all projects.</td>
</tr>
<tr>
<td>r131.</td>
<td>Maintaining Internet connectivity.</td>
</tr>
<tr>
<td>r132.</td>
<td>Maintaining project executables in the central repository.</td>
</tr>
<tr>
<td>r133.</td>
<td>Making arrangements for traveling including air tickets, visas, hotel &amp; car booking, and documents etc.</td>
</tr>
<tr>
<td>r134.</td>
<td>Making backup copies and distributing final release packages to Project Management.</td>
</tr>
<tr>
<td>r135.</td>
<td>Manage inventory records in an effective manner.</td>
</tr>
<tr>
<td>r136.</td>
<td>Manage the employee leave management system.</td>
</tr>
<tr>
<td>r137.</td>
<td>Management of the Source Control Repository (SCR).</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>r138.</td>
<td>Managing administrative activities, across all teams, and overseeing all staff.</td>
</tr>
<tr>
<td>r139.</td>
<td>Managing and operating the telephone exchange</td>
</tr>
<tr>
<td>r140.</td>
<td>Managing expeditious processing of proposed changes against approved baselines.</td>
</tr>
<tr>
<td>r141.</td>
<td>Managing lunch facility for employees.</td>
</tr>
<tr>
<td>r142.</td>
<td>Managing organizational administrative tasks.</td>
</tr>
<tr>
<td>r143.</td>
<td>Managing the inventory of recorded material.</td>
</tr>
<tr>
<td>r144.</td>
<td>Managing the inventory of recorded material.</td>
</tr>
<tr>
<td>r145.</td>
<td>Managing the database administration team.</td>
</tr>
<tr>
<td>r146.</td>
<td>Managing the expeditious processing of proposed changes against approved baselines.</td>
</tr>
<tr>
<td>r147.</td>
<td>Managing the inventory of recorded material.</td>
</tr>
<tr>
<td>r148.</td>
<td>Managing the inventory of recorded material.</td>
</tr>
<tr>
<td>r149.</td>
<td>Managing the leadership team to ensure efficient operations within performance standards, corporate policies, regulatory guidelines and budget parameters.</td>
</tr>
<tr>
<td>r150.</td>
<td>Managing the processing of authorized changes into approved baselines.</td>
</tr>
<tr>
<td>r151.</td>
<td>Team management.</td>
</tr>
<tr>
<td>r152.</td>
<td>Managing the suppliers’ data and prioritize them for future references.</td>
</tr>
<tr>
<td>r153.</td>
<td>Managing the support staff (peons, receptionists, etc.).</td>
</tr>
<tr>
<td>r154.</td>
<td>Meeting all the build timelines and schedules.</td>
</tr>
<tr>
<td>r155.</td>
<td>Monitoring and tracking project based activities.</td>
</tr>
<tr>
<td>r156.</td>
<td>Obtaining external quotes for purchases</td>
</tr>
<tr>
<td>r157.</td>
<td>Operate and look after power generator, bring diesel for generator, general service and handling related issues.</td>
</tr>
<tr>
<td>r158.</td>
<td>Operating and looking after the power generator, including but not limited to bringing diesel, general servicing and handling related issues.</td>
</tr>
<tr>
<td>r159.</td>
<td>Participating in technical reviews of organizational assets.</td>
</tr>
<tr>
<td>r160.</td>
<td>Participating in the development and implementation of the annual budget and managing costs to ensure compliance.</td>
</tr>
<tr>
<td>r162.</td>
<td>Perform electronic operations in the absence of an electrician.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>r163.</td>
<td>Perform month-end and year-end accounting (prepare and input journals, review of accruals, etc.).</td>
</tr>
<tr>
<td>r164.</td>
<td>Perform month-end and year-end accounting for Provident Fund Trust account.</td>
</tr>
<tr>
<td>r165.</td>
<td>Performing all testing activities as planned and scheduled.</td>
</tr>
<tr>
<td>r166.</td>
<td>Performing and documenting a root cause analysis of any defects.</td>
</tr>
<tr>
<td>r168.</td>
<td>Performing electronic operations in the absence of an electrician.</td>
</tr>
<tr>
<td>r169.</td>
<td>Performing internal audits of projects or other functional areas periodically, as planned, to make sure that assigned resources are working in accordance with defined processes.</td>
</tr>
<tr>
<td>r170.</td>
<td>Performing other miscellaneous duties and special projects as assigned, or as deemed necessary</td>
</tr>
<tr>
<td>r171.</td>
<td>Performing other network support duties as assigned</td>
</tr>
<tr>
<td>r172.</td>
<td>Performing physical configuration audits for project.</td>
</tr>
<tr>
<td>r173.</td>
<td>Performing test cases and code review.</td>
</tr>
<tr>
<td>r174.</td>
<td>Plan and conduct new employee orientation to foster positive attitude toward organizational objectives</td>
</tr>
<tr>
<td>r175.</td>
<td>Planning SQA activities.</td>
</tr>
<tr>
<td>r176.</td>
<td>Play a role in build preparation activities.</td>
</tr>
<tr>
<td>r177.</td>
<td>Play a role in the preparation of the Bi-Traceability matrix.</td>
</tr>
<tr>
<td>r178.</td>
<td>Playing an active role in all quality oriented process improvement activities.</td>
</tr>
<tr>
<td>r179.</td>
<td>Preparation of requests for reproduction of material.</td>
</tr>
<tr>
<td>r180.</td>
<td>Prepare weekly bank reconciliation statements and check registers.</td>
</tr>
<tr>
<td>r181.</td>
<td>Preparing and publishing organizational software quality assurance audit plans.</td>
</tr>
<tr>
<td>r182.</td>
<td>Preparing software configuration management plans, in accordance with defined organizational standards, for all projects.</td>
</tr>
<tr>
<td>r183.</td>
<td>Proactively managing the performance of team members through coaching, training, rewarding strong performance, and addressing performance issues directly and with fairness if/when they arise.</td>
</tr>
<tr>
<td>r184.</td>
<td>Process monthly records for all payrolls.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>r185.</td>
<td>Provide bill payment and other minor shopping and bank oriented services to executives; as required.</td>
</tr>
<tr>
<td>r186.</td>
<td>Provide lunch and tea facility to employees.</td>
</tr>
<tr>
<td>r187.</td>
<td>Providing an adequate, organized supply of forms for office use and distribution.</td>
</tr>
<tr>
<td>r188.</td>
<td>Providing assistance to the QA manager in defining and implementing QA and QC activities within the organization.</td>
</tr>
<tr>
<td>r189.</td>
<td>Providing assistance to the QA manager in developing SQA audit plans.</td>
</tr>
<tr>
<td>r190.</td>
<td>Providing consultancy and help, for facilitation, to the PM and project team, to get adherence with defined processes.</td>
</tr>
<tr>
<td>r191.</td>
<td>Providing consultancy and help, for facilitation, to the PM and project team.</td>
</tr>
<tr>
<td>r192.</td>
<td>Providing leadership and working with staff to create a high performance, customer-service-oriented work environment that supports the department’s mission, objectives and service expectations.</td>
</tr>
<tr>
<td>r193.</td>
<td>Providing organization wide LAN/WAN support.</td>
</tr>
<tr>
<td>r194.</td>
<td>Providing support to project teams in resolving database related issues.</td>
</tr>
<tr>
<td>r195.</td>
<td>Providing technical expertise in the use of software testing tools to QA engineers developing automation architectures, custom test tools, and complex test environments.</td>
</tr>
<tr>
<td>r196.</td>
<td>Publishing coding standards and compliance.</td>
</tr>
<tr>
<td>r197.</td>
<td>Purchasing goods, hardware and software items as required within the organization.</td>
</tr>
<tr>
<td>r198.</td>
<td>Repair, maintain and clean office furniture, sanitary and hardware items etc.</td>
</tr>
<tr>
<td>r199.</td>
<td>Repairing and maintaining office furniture, sanitary items, etc.</td>
</tr>
<tr>
<td>r200.</td>
<td>Repairing hardware items like printers, monitors etc.</td>
</tr>
<tr>
<td>r201.</td>
<td>Replacing old products.</td>
</tr>
<tr>
<td>r202.</td>
<td>Reporting progress to the quality assurance manager on a weekly basis.</td>
</tr>
<tr>
<td>r203.</td>
<td>Reporting SQA activities to the QA manager.</td>
</tr>
<tr>
<td>r204.</td>
<td>Reproducing and analyzing reported defect(s).</td>
</tr>
<tr>
<td>r205.</td>
<td>Reproducing internal and external defects, reported by clients, and supporting developers in their removal.</td>
</tr>
<tr>
<td>r206</td>
<td>Resolving any email problems and network issues for onsite offices.</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------------------------------</td>
</tr>
<tr>
<td>r207</td>
<td>Responsible for department results in terms of planning, costs, methods, and variances from the budget.</td>
</tr>
<tr>
<td>r208</td>
<td>Responsible for enforcing organizational processes at all levels and for verifying their compliance with plans and standards.</td>
</tr>
<tr>
<td>r209</td>
<td>Responsible for keeping all the servers up and running while ensuring the user access security policies to be implemented.</td>
</tr>
<tr>
<td>r210</td>
<td>Responsible for leading the quality assurance (QA) team.</td>
</tr>
<tr>
<td>r211</td>
<td>Retrieving and distributing incoming mail.</td>
</tr>
<tr>
<td>r212</td>
<td>Review of payroll, cash forecasts, expense vouchers and other general documentation.</td>
</tr>
<tr>
<td>r213</td>
<td>Reviewing and signing-off acceptance on requirement documents.</td>
</tr>
<tr>
<td>r214</td>
<td>Reviewing build packages.</td>
</tr>
<tr>
<td>r215</td>
<td>Reviewing test cases developed by testers.</td>
</tr>
<tr>
<td>r216</td>
<td>Verifying the design report.</td>
</tr>
<tr>
<td>r217</td>
<td>Reviewing the impact analysis of defects and finalizing them.</td>
</tr>
<tr>
<td>r218</td>
<td>Servicing/ repairing air conditioners.</td>
</tr>
<tr>
<td>r219</td>
<td>Serving in the capacity of a network security administrator, including administration of the firewall rule base and monitoring of network intrusions.</td>
</tr>
<tr>
<td>r220</td>
<td>Software configuration control.</td>
</tr>
<tr>
<td>r221</td>
<td>Storing final released products.</td>
</tr>
<tr>
<td>r222</td>
<td>Supporting the development of technical standards and application uses.</td>
</tr>
<tr>
<td>r223</td>
<td>Take care of overall office environment (health, safety, cleanliness).</td>
</tr>
<tr>
<td>r224</td>
<td>Taking decisions related to management issues.</td>
</tr>
<tr>
<td>r225</td>
<td>Test Functional Requirements</td>
</tr>
<tr>
<td>r226</td>
<td>Test Non Functional Requirements</td>
</tr>
<tr>
<td>r227</td>
<td>Test Procedures</td>
</tr>
<tr>
<td>r228</td>
<td>Tune Database</td>
</tr>
<tr>
<td>r229</td>
<td>To accommodate process improvement activities through involvement in project’s weekly progress meetings and otherwise.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>r230.</td>
<td>To get adherence with defined processes and to accommodate process improvement activities through involvement in project’s weekly progress meetings and otherwise.</td>
</tr>
<tr>
<td>r231.</td>
<td>Tracking and managing software defects from beginning to end following a structured code migration path.</td>
</tr>
<tr>
<td>r232.</td>
<td>Training of developers in database related issues.</td>
</tr>
<tr>
<td>r233.</td>
<td>Transforming application business requirements and analysis use cases into test cases and scripts.</td>
</tr>
<tr>
<td>r234.</td>
<td>Traveling inside/outside Pakistan and working onsite with the client.</td>
</tr>
<tr>
<td>r235.</td>
<td>Unit testing of fixed code.</td>
</tr>
<tr>
<td>r236.</td>
<td>Verifying that final released products match preliminary products.</td>
</tr>
<tr>
<td>r237.</td>
<td>Write Requirements Specifications</td>
</tr>
<tr>
<td>r238.</td>
<td>Write Design Documents</td>
</tr>
<tr>
<td>r239.</td>
<td>Write Code Documents</td>
</tr>
</tbody>
</table>
## APPENDIX- III
(Domain Vocabulary of a Software Development Organization)

<table>
<thead>
<tr>
<th>Id</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Accounting Manager</td>
</tr>
<tr>
<td>2</td>
<td>Administration Assistant</td>
</tr>
<tr>
<td>3</td>
<td>Administration Manager</td>
</tr>
<tr>
<td>4</td>
<td>AJAX WD</td>
</tr>
<tr>
<td>5</td>
<td>Algorithm Design Module</td>
</tr>
<tr>
<td>6</td>
<td>Algorithm Implementation Module</td>
</tr>
<tr>
<td>7</td>
<td>Analysis Team</td>
</tr>
<tr>
<td>8</td>
<td>Analysis Unit</td>
</tr>
<tr>
<td>9</td>
<td>Analyst</td>
</tr>
<tr>
<td>10</td>
<td>Software Application Manager</td>
</tr>
<tr>
<td>11</td>
<td>Architecture Team</td>
</tr>
<tr>
<td>12</td>
<td>Assistant Manager Finance</td>
</tr>
<tr>
<td>13</td>
<td>BIOS Dev. Engg</td>
</tr>
<tr>
<td>14</td>
<td>Build Release Developer</td>
</tr>
<tr>
<td>15</td>
<td>BOA Developer</td>
</tr>
<tr>
<td>16</td>
<td>CEO Manager</td>
</tr>
<tr>
<td>17</td>
<td>CEO Assistant Manager</td>
</tr>
<tr>
<td>18</td>
<td>C++ C# Developer</td>
</tr>
<tr>
<td>19</td>
<td>CI Officer</td>
</tr>
<tr>
<td>20</td>
<td>Client Support Analyst</td>
</tr>
<tr>
<td>21</td>
<td>Coder</td>
</tr>
<tr>
<td>22</td>
<td>Coding Module</td>
</tr>
<tr>
<td>23</td>
<td>Component Engineer</td>
</tr>
<tr>
<td>24</td>
<td>Configuration Engineer</td>
</tr>
<tr>
<td>25</td>
<td>Configuration Manager</td>
</tr>
<tr>
<td>26</td>
<td>Database Administrator</td>
</tr>
<tr>
<td>27</td>
<td>Database Manager</td>
</tr>
<tr>
<td>28.</td>
<td>DE Operator</td>
</tr>
<tr>
<td>29.</td>
<td>Db Design Module</td>
</tr>
<tr>
<td>30.</td>
<td>DBA</td>
</tr>
<tr>
<td>31.</td>
<td>Deployment Team</td>
</tr>
<tr>
<td>32.</td>
<td>Deputy Project Manager</td>
</tr>
<tr>
<td>33.</td>
<td>Design Module</td>
</tr>
<tr>
<td>34.</td>
<td>Designer</td>
</tr>
<tr>
<td>35.</td>
<td>Design Team</td>
</tr>
<tr>
<td>36.</td>
<td>Detail Design Module</td>
</tr>
<tr>
<td>37.</td>
<td>Developer</td>
</tr>
<tr>
<td>38.</td>
<td>Development Lead Engineer</td>
</tr>
<tr>
<td>39.</td>
<td>Director Software Development</td>
</tr>
<tr>
<td>40.</td>
<td>Documenter</td>
</tr>
<tr>
<td>41.</td>
<td>Domain Expert</td>
</tr>
<tr>
<td>42.</td>
<td>Embedded S. Engineer</td>
</tr>
<tr>
<td>43.</td>
<td>Employee</td>
</tr>
<tr>
<td>44.</td>
<td>Front Desk Officer/Receptionist</td>
</tr>
<tr>
<td>45.</td>
<td>Functional Requirement Task</td>
</tr>
<tr>
<td>46.</td>
<td>Graphic Designer</td>
</tr>
<tr>
<td>47.</td>
<td>GUI Design Module</td>
</tr>
<tr>
<td>48.</td>
<td>GUI Implementation Module</td>
</tr>
<tr>
<td>49.</td>
<td>HR Assistant</td>
</tr>
<tr>
<td>50.</td>
<td>Identity Access Manager</td>
</tr>
<tr>
<td>51.</td>
<td>Implementation Team</td>
</tr>
<tr>
<td>52.</td>
<td>In-House Project Procedure</td>
</tr>
<tr>
<td></td>
<td>Name of the Position</td>
</tr>
<tr>
<td>---</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>53.</td>
<td>Intern Software Developer</td>
</tr>
<tr>
<td>54.</td>
<td>ISO Project Procedure</td>
</tr>
<tr>
<td>55.</td>
<td>IT Manager</td>
</tr>
<tr>
<td>56.</td>
<td>IT Network Manager</td>
</tr>
<tr>
<td>57.</td>
<td>Junior Systems Administrator</td>
</tr>
<tr>
<td>58.</td>
<td>Lead Software Dev. Engineer</td>
</tr>
<tr>
<td>59.</td>
<td>Librarian</td>
</tr>
<tr>
<td>60.</td>
<td>Manager - Software Development</td>
</tr>
<tr>
<td>61.</td>
<td>Marketing Manager</td>
</tr>
<tr>
<td>62.</td>
<td>Net. Engg</td>
</tr>
<tr>
<td>63.</td>
<td>Network Manager</td>
</tr>
<tr>
<td>64.</td>
<td>Network Administrator</td>
</tr>
<tr>
<td>65.</td>
<td>Network Support Manager</td>
</tr>
<tr>
<td>66.</td>
<td>None Functional Requirement Task</td>
</tr>
<tr>
<td>67.</td>
<td>Non Technical Employee</td>
</tr>
<tr>
<td>68.</td>
<td>Person</td>
</tr>
<tr>
<td>69.</td>
<td>PHP Developer</td>
</tr>
<tr>
<td>70.</td>
<td>Principal .NET S. Engineer</td>
</tr>
<tr>
<td>71.</td>
<td>Principal Configuration Engineer</td>
</tr>
<tr>
<td>72.</td>
<td>Principal Database Administrator</td>
</tr>
<tr>
<td></td>
<td>Position</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>73.</td>
<td>Principal QAE</td>
</tr>
<tr>
<td>74.</td>
<td>Principal S. Engineer</td>
</tr>
<tr>
<td>75.</td>
<td>Process Manager</td>
</tr>
<tr>
<td>76.</td>
<td>Product Developer</td>
</tr>
<tr>
<td>77.</td>
<td>Programmer</td>
</tr>
<tr>
<td>78.</td>
<td>Project</td>
</tr>
<tr>
<td>79.</td>
<td>Project Procedure (SOP)</td>
</tr>
<tr>
<td>80.</td>
<td>Project Manager</td>
</tr>
<tr>
<td>81.</td>
<td>QA Analyst, Operations – Development</td>
</tr>
<tr>
<td>82.</td>
<td>QA Tester</td>
</tr>
<tr>
<td>83.</td>
<td>QAE</td>
</tr>
<tr>
<td>84.</td>
<td>Quality Assurance Manager</td>
</tr>
<tr>
<td>85.</td>
<td>Report Design Module</td>
</tr>
<tr>
<td>86.</td>
<td>Senior Analyst</td>
</tr>
<tr>
<td>87.</td>
<td>Senior Configuration Engineer</td>
</tr>
<tr>
<td>88.</td>
<td>Sr DBA</td>
</tr>
<tr>
<td>89.</td>
<td>Senior Device Networking Software Dev. Engineer</td>
</tr>
<tr>
<td>90.</td>
<td>Senior Director Software Development</td>
</tr>
<tr>
<td>91.</td>
<td>Senior IT Support Analyst</td>
</tr>
<tr>
<td>92.</td>
<td>Senior LINUX / UNIX Administrator</td>
</tr>
<tr>
<td></td>
<td>Position Description</td>
</tr>
<tr>
<td>---</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>93.</td>
<td>Sr Network Engg</td>
</tr>
<tr>
<td>94.</td>
<td>Senior QA Test Engineer</td>
</tr>
<tr>
<td>95.</td>
<td>Senior QAE</td>
</tr>
<tr>
<td>96.</td>
<td>Senior Software Development Consultant</td>
</tr>
<tr>
<td>97.</td>
<td>Senior Software Dev. Engineer</td>
</tr>
<tr>
<td>98.</td>
<td>Senior S. Engineer</td>
</tr>
<tr>
<td>99.</td>
<td>Senior SQL Server Database Developer</td>
</tr>
<tr>
<td>100.</td>
<td>Senior SQL Server DBA</td>
</tr>
<tr>
<td>101.</td>
<td>Senior Teradata Developer/DBA</td>
</tr>
<tr>
<td>102.</td>
<td>SOA J2EE / Oracle ADF Software Developer</td>
</tr>
<tr>
<td>103.</td>
<td>Software Architect</td>
</tr>
<tr>
<td>104.</td>
<td>Software Configuration Manager</td>
</tr>
<tr>
<td>105.</td>
<td>Software Design Engineer</td>
</tr>
<tr>
<td>106.</td>
<td>Software Developer</td>
</tr>
<tr>
<td>107.</td>
<td>Software Dev. Engineer</td>
</tr>
<tr>
<td>108.</td>
<td>Software Development Functional Lead</td>
</tr>
<tr>
<td>109.</td>
<td>Software Development Manager</td>
</tr>
<tr>
<td>110.</td>
<td>Software Development Trainer</td>
</tr>
<tr>
<td>111.</td>
<td>S. Engineer</td>
</tr>
<tr>
<td>112.</td>
<td>Software Programmer CNet</td>
</tr>
<tr>
<td></td>
<td>Job Title</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>113</td>
<td>Software Quality Assurance Manager</td>
</tr>
<tr>
<td>114</td>
<td>Software Technician</td>
</tr>
<tr>
<td>115</td>
<td>Software Tester</td>
</tr>
<tr>
<td>116</td>
<td>SQA Engineer</td>
</tr>
<tr>
<td>117</td>
<td>SQA Team</td>
</tr>
<tr>
<td>118</td>
<td>Sr Information Developer</td>
</tr>
<tr>
<td>119</td>
<td>Sr Software Development Manager</td>
</tr>
<tr>
<td>120</td>
<td>Sr S. Engineer</td>
</tr>
<tr>
<td>121</td>
<td>Sr Web Developer</td>
</tr>
<tr>
<td>122</td>
<td>Structure Design Module</td>
</tr>
<tr>
<td>123</td>
<td>Structure Implementation Module</td>
</tr>
<tr>
<td>124</td>
<td>System Integrator</td>
</tr>
<tr>
<td>125</td>
<td>Systems Manager</td>
</tr>
<tr>
<td>126</td>
<td>Systems S. Engineer</td>
</tr>
<tr>
<td>127</td>
<td>Technical Employee</td>
</tr>
<tr>
<td>128</td>
<td>Technical Lead</td>
</tr>
<tr>
<td>129</td>
<td>Technical Writer</td>
</tr>
<tr>
<td>130</td>
<td>Use Cases Engineer</td>
</tr>
<tr>
<td>131</td>
<td>Use Cases Specifier</td>
</tr>
<tr>
<td>132</td>
<td>Web Architect</td>
</tr>
</tbody>
</table>
The ontologies of the following Software Houses have been used to populate domain vocabulary

1. Imzee Consulting (www.imzeeconsulting.com)
2. Xavor Corporation (www.xavor.com)
3. Effectivesoft http://www.effectivesoft.com
5. Softmantechologies(www.softmantech.com)
7. AEDSYS Pvt Ltd
8. Adam Soft International
9. OOOBER Pvt Ltd (www.ooober.com)
10. Title Developments (www.titledevelopments.com)
12. Cambridge Docs www.cambridgedocs.com
13. I2C Inc. Lahore, Pakistan
APPENDIX-IV
(Datasoft.owl)

<!-- Axioms: 135 -->

<Ontology
xmlns="http://www.w3.org/2006/12/owl11-xml#"
xml:base="http://www.w3.org/2006/12/owl11-xml#"
xmlns:owl11="http://www.w3.org/2006/12/owl11#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
URI="http://www.semanticweb.org/ontologies/2008/2/dataSoft.owl#"

<Declaration>
  <ObjectProperty
    URI="&dataSoft;designWebpage"/>
</Declaration>

<Declaration>
  <ObjectProperty
    URI="&dataSoft;runAlgorithm"/>
</Declaration>

<Declaration>
  <ObjectProperty
    URI="&dataSoft;runAlgorithm"/>
</Declaration>

<SubClassOf>
  <OWLClass URI="&dataSoft;Analyst"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;SeniorAnalyst"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;SeniorSoftwareEngineer"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;Designer"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;BackupDB"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;Database"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;Algorithms"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;GUI"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;SeniorAnalyst"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;SeniorSoftwareEngineer"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;Designer"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;BackupDB"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;Designer"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;BackupDB"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;Database"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;GUI"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;SeniorAnalyst"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;SeniorSoftwareEngineer"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;Designer"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;BackupDB"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;SeniorAnalyst"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;SeniorSoftwareEngineer"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;Designer"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;BackupDB"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>

<SubClassOf>
  <OWLClass URI="&dataSoft;DetailDesign"/>
<OWLClass URI="&dataSoft;SeniorAnalyst"/>
</Declaration>
<ObjectPropertyDomain>
<ObjectProperty
URI="&dataSoft;implementDB"/>
</ObjectPropertyDomain>
<SubClassOf>
<OWLClass URI="&dataSoft;TechnicalWriter"/>
</SubClassOf>
<SubClassOf>
<OWLClass URI="&dataSoft;SQAEngineer"/>
</SubClassOf>
<SubClassOf>
<OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>
<ObjectProperty>
URI="&dataSoft;testNFReq"/>
</Declaration>
<ObjectProperty>
URI="&dataSoft;SQAEngineer"/>
</SubOfWeek>
<ObjectProperty>
URI="&dataSoft;designGUI"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
URI="&dataSoft;designGraphic"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
URI="&dataSoft;prepareCodeDoc"/>
</Declaration>
<ObjectPropertyRange>
URI="&dataSoft;analyzeSReq"/>
</OWLClass>
<ObjectPropertyRange>
URI="&dataSoft;SoftwareReqs"/>
</ObjectPropertyRange>
<ObjectPropertyDomain>
<ObjectProperty>
URI="&dataSoft;designStructure"/>
</ObjectProperty>
<ObjectProperty>
URI="&dataSoft;SoftwareEngineer"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
<ObjectProperty>
URI="&dataSoft;implementDB"/>
</ObjectPropertyDomain>
<ObjectPropertyRange>
<ObjectProperty>
URI="&dataSoft;SoftwareEngineer"/>
</ObjectProperty>
<ObjectProperty>
URI="&dataSoft;implementDB"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
<ObjectProperty>
URI="&dataSoft;SeniorSoftwareEngineer"/>
</ObjectProperty>
<ObjectProperty>
URI="&dataSoft;designGUI"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
<ObjectProperty>
URI="&dataSoft;SeniorProgrammer"/>
</ObjectProperty>
<ObjectProperty>
URI="&dataSoft;designGUI"/>
</ObjectPropertyRange>
<ObjectProperty URI="&dataSoft;analyzeHReq"/>
<OWLClass URI="&dataSoft;Analyst"/>
</ObjectPropertyDomain>
<Declaration>
<OWLClass URI="&dataSoft;SeniorSoftwareEngineer"/>
</Declaration>
<ObjectProperty URI="&dataSoft;implementStructure"/>
</Declaration>
<SubClassOf>
<OWLClass URI="&dataSoft;Structure"/>
<OWLClass URI="&dataSoft;Implementation"/>
</SubClassOf>
<ObjectPropertyRange>
<ObjectProperty URI="&dataSoft;analyzeHReq"/>
</ObjectPropertyRange>
<Declaration>
<ObjectProperty URI="&dataSoft;prepareUserManual"/>
</Declaration>
<ObjectPropertyDomain>
<ObjectProperty URI="&dataSoft;manageResource"/>
</ObjectPropertyDomain>
<Declaration>
<ObjectProperty URI="&dataSoft;designGraphic"/>
</Declaration>
<ObjectPropertyDomain>
<ObjectProperty URI="&dataSoft;implementGUI"/>
</ObjectPropertyDomain>
<SubClassOf>
<OWLClass URI="&dataSoft;Coder"/>
<OWLClass URI="&owl;Thing"/>
</SubClassOf>
<Declaration>
<OWLClass URI="&dataSoft;Procedure"/>
</Declaration>
<ObjectPropertyDomain>
<ObjectProperty URI="&dataSoft;designWebpage"/>
</ObjectPropertyDomain>
<SubClassOf>
<OWLClass URI="&dataSoft;Graphics"/>
<OWLClass URI="&dataSoft;DetailDesign"/>
</SubClassOf>
<Declaration>
<OWLClass URI="&dataSoft;DetailDesign"/>
</Declaration>
<ObjectProperty URI="&dataSoft;implementGUI"/>
</ObjectProperty>
<Declaration>
<ObjectProperty URI="&dataSoft;SeniorSoftwareEngineer"/>
</Declaration>
<ObjectProperty URI="&dataSoft;designStructure"/>
</Declaration>
<ObjectProperty URI="&dataSoft;designStructure"/>
</ObjectProperty>
<Declaration>
<ObjectProperty URI="&dataSoft;SoftwareEngineer"/>
</Declaration>
<ObjectProperty URI="&dataSoft;designGUI"/>
</SubClassOf>
<ObjectProperty URI="&dataSoft;SoftwareEngineer"/>
</ObjectProperty>
<Declaration>
<ObjectProperty URI="&dataSoft;Programmer"/>
</Declaration>
<ObjectProperty URI="&dataSoft;implementGUI"/>
</ObjectProperty>
<Declaration>
<ObjectProperty URI="&dataSoft;Programmer"/>
</Declaration>
<ObjectProperty URI="&dataSoft;implementGUI"/>
</ObjectProperty>
<Declaration>
<ObjectProperty URI="&dataSoft;Programmer"/>
</Declaration>
<ObjectProperty URI="&dataSoft;implementGUI"/>
</ObjectProperty>
<OWLClass URI="&dataSoft;Implementation"/>
</SubClassOf>
<OWLClass URI="&dataSoft;SeniorAnalyst"/>
  <OWLClass URI="&owl;Thing"/>
</SubClassOf>
</OWLClass URI="&dataSoft;Database"/>
</ObjectPropertyRange>
<SubClassOf>
  <OWLClass URI="&dataSoft;SoftwareReqs"/>
  <OWLClass URI="&dataSoft;ProjectReq"/>
</SubClassOf>
<ObjectPropertyDomain>
<ObjectProperty URI="&dataSoft;analyzeFReq"/>
  <OWLClass URI="&dataSoft;Analyst"/>
</ObjectPropertyDomain>
<ObjectProperty URI="&dataSoft;estimateCost"/>
  <OWLClass URI="&dataSoft;Project"/>
</ObjectPropertyDomain>
<ObjectProperty URI="&dataSoft;designGraphic"/>
  <OWLClass URI="&dataSoft;Designer"/>
</ObjectPropertyDomain>
<ObjectProperty URI="&dataSoft;prepareDesignDoc"/>
</OWLClass URI="&dataSoft;TechnicalWriter"/>
<ObjectPropertyDomain>
<ObjectProperty URI="&dataSoft;manageResource"/>
</Declaration>
<OWLClass URI="&dataSoft;StructureDesign"/>
<OWLClass URI="&dataSoft;ProjectDesign"/>
</SubClassOf>
<Declaration>
<ObjectProperty URI="&dataSoft;implementGUI"/>
</Declaration>
<ObjectPropertyDomain>
<ObjectProperty URI="&dataSoft;testFReq"/>
  <OWLClass URI="&dataSoft;SQAEngineer"/>
</ObjectPropertyDomain>
<ObjectProperty URI="&dataSoft;implementStructure"/>
  <OWLClass URI="&dataSoft;Structure"/>
</ObjectPropertyRange>
<ObjectProperty URI="&dataSoft;backupDB"/>
  <OWLClass URI="&dataSoft;DBA"/>
</ObjectPropertyRange>
<SubClassOf>
  <OWLClass URI="&dataSoft;SoftwareEngineer"/>
  <OWLClass URI="&owl;Thing"/>
</SubClassOf>
<ObjectProperty Domain>
<ObjectProperty URI="&dataSoft;analyzeFReq"/>
  <OWLClass URI="&dataSoft;SeniorSoftwareEngineer"/>
</ObjectPropertyDomain>
<ObjectProperty URI="&dataSoft;designReport"/>
  <OWLClass URI="&dataSoft;SeniorProgrammer"/>
</ObjectPropertyDomain>
<ObjectProperty URI="&dataSoft;designWebpage"/>
  <OWLClass URI="&dataSoft;WebPages"/>
</ObjectPropertyRange>
<ObjectProperty URI="&dataSoft;testNFReq"/>
  <OWLClass URI="&dataSoft;NonFunctionalReqs"/>
</ObjectPropertyRange>
<SubClassOf>
  <OWLClass URI="&dataSoft;Procedure"/>
  <OWLClass URI="&dataSoft;Project"/>
</SubClassOf>
APPENDIX-V
(ridos.owl)

<!-- Axioms: 92 -->

URI="http://www.semanticweb.org/ontologies/2008/2/ridos.owl">
  <ObjectPropertyDomain>
    <ObjectProperty URI="&ridos;designGUI"/>
    <OWLClass URI="&ridos;SoftwareArchitect"/>
  </ObjectPropertyDomain>
  <SubClassOf>
    <OWLClass URI="&ridos;Algorithms"/>
    <OWLClass URI="&ridos;DetailDesign"/>
  </SubClassOf>
  <ObjectPropertyDomain>
    <ObjectProperty URI="&ridos;designReport"/>
    <OWLClass URI="&ridos;SoftwareArchitect"/>
  </ObjectPropertyDomain>
  <ObjectProperty Domain URI="&ridos;managerResources"/>
    <OWLClass URI="&ridos;ProjectManager"/>
  </ObjectPropertyDomain>
  <ObjectProperty Domain URI="&ridos;designGraphic"/>
    <OWLClass URI="&ridos;SoftwareArchitect"/>
  </ObjectPropertyDomain>
  <ObjectProperty Range URI="&ridos;implementAlgorithm"/>
    <OWLClass URI="&ridos;Algorithms"/>
  </ObjectPropertyRange>
  <ObjectProperty Domain URI="&ridos;designStructure"/>
    <OWLClass URI="&ridos;StructureDesign"/>
  </ObjectPropertyRange>
  <SubClassOf>
    <OWLClass URI="&ridos;Procedure"/>
    <OWLClass URI="&owl;Thing"/>
  </SubClassOf>
  <SubClassOf>
    <OWLClass URI="&ridos;GUI"/>
    <OWLClass URI="&ridos;DetailDesign"/>
  </SubClassOf>
  <ObjectProperty Domain URI="&ridos;implementDB"/>
    <OWLClass URI="&ridos;DBA"/>
  </ObjectPropertyDomain>
  <ObjectProperty Range URI="&ridos;backupDB"/>
    <OWLClass URI="&ridos;Database"/>
  </ObjectPropertyRange>
  <SubClassOf>
    <OWLClass URI="&ridos;TechnicalWriter"/>
    <OWLClass URI="&owl;Thing"/>
  </SubClassOf>
  <SubClassOf>
    <OWLClass URI="&ridos;Reports"/>
    <OWLClass URI="&ridos;DetailDesign"/>
  </SubClassOf>
  <SubClassOf>
    <OWLClass URI="&ridos;SQAEngineer"/>
    <OWLClass URI="&owl;Thing"/>
  </SubClassOf>
  <ObjectProperty Range URI="&ridos;designGUI"/>
    <OWLClass URI="&ridos;GUI"/>
  </ObjectPropertyRange>
  <ObjectProperty Range URI="&ridos;designGraphic"/>
    <OWLClass URI="&ridos;Graphics"/>
  </ObjectPropertyRange>
  <ObjectProperty Range URI="&ridos;analyzeSReq"/>
    <OWLClass URI="&ridos;SoftwareReqs"/>
  </ObjectPropertyRange>
  <ObjectProperty Range URI="&ridos;testProcedure"/>
    <OWLClass URI="&ridos;Procedure"/>
  </ObjectPropertyRange>
  <ObjectProperty Range URI="&ridos;implementGUI"/>
    <OWLClass URI="&ridos;Coder"/>
  </ObjectPropertyRange>
  <ObjectProperty Range URI="&ridos;implementGUI"/>
    <OWLClass URI="&ridos;GUI"/>
  </ObjectPropertyRange>
</ObjectPropertyDomain>
<ObjectPropertyURI="&ridos;implementStructure"/>
   <OWLClassURI="&ridos;Coder"/>
</ObjectPropertyDomain>
<SubClassOf>
   <OWLClassURI="&ridos;Database"/>
   <OWLClassURI="&ridos;Service"/>
</SubClassOf>
<SubClassOf>
   <OWLClassURI="&ridos;DBA"/>
   <OWLClassURI="&owl;Thing"/>
</SubClassOf>
<ObjectPropertyRange>
   <ObjectPropertyURI="&ridos;designAlgorithm"/>
   <OWLClassURI="&ridos;Algorithms"/>
</ObjectPropertyRange>
<ObjectPropertyURI="&ridos;analyzeNFReq"/>
   <OWLClassURI="&ridos;NonFunctionalReqs"/>
</ObjectPropertyRange>
<SubClassOf>
   <OWLClassURI="&ridos;ProjectReq"/>
   <OWLClassURI="&ridos;Project"/>
</SubClassOf>
<ObjectPropertyDomain>
   <ObjectPropertyURI="&ridos;testProcedure"/>
   <OWLClassURI="&ridos;SQAEngineer"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
   <ObjectPropertyURI="&ridos;defineProcess"/>
   <OWLClassURI="&ridos;ProjectManager"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
   <ObjectPropertyURI="&ridos;tuneDB"/>
   <OWLClassURI="&ridos;DBA"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
   <ObjectPropertyURI="&ridos;designWebpage"/>
   <OWLClassURI="&ridos;SoftwareArchitect"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
   <ObjectPropertyURI="&ridos;manageChange"/>
   <OWLClassURI="&ridos;SofConfigManager"/>
</ObjectPropertyDomain>
<ObjectPropertyURI="&ridos;implementReport"/>
   <OWLClassURI="&ridos;Reports"/>
</ObjectPropertyRange>
<ObjectPropertyDomain>
   <ObjectPropertyURI="&ridos;designDB"/>
   <OWLClassURI="&ridos;Programmer"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
   <ObjectPropertyURI="&ridos;implementReport"/>
   <OWLClassURI="&ridos;Coder"/>
</ObjectPropertyDomain>
<SubClassOf>
   <OWLClassURI="&ridos;DetailDesign"/>
   <OWLClassURI="&ridos;ProjectDesign"/>
</SubClassOf>
<ObjectPropertyDomain>
   <ObjectPropertyURI="&ridos;prepareCodeDoc"/>
   <OWLClassURI="&ridos;TechnicalWriter"/>
<owl:Ontology>
  <owl:Class URI="&ridos;Programmer"/>
  <owl:Class URI="&owl;Thing"/>
</owl:ClassOf>

<owl:ObjectProperty> URI="&ridos;implementAlgorithm"/>
  <owl:Class URI="&ridos;Coder"/>
</owl:ObjectPropertyDomain>

<owl:SubClassOf>
  <owl:Class URI="&ridos;Reports"/>
  <owl:Class URI="&ridos;Implementation"/>
</owl:SubClassOf>

<owl:ObjectProperty> URI="&ridos;analyzeNFReq"/>
  <owl:Class URI="&ridos;Analyst"/>
</owl:ObjectPropertyDomain>

<owl:SubClassOf>
  <owl:Class URI="&ridos;WebPages"/>
  <owl:Class URI="&ridos;DetailDesign"/>
</owl:SubClassOf>

<owl:ObjectProperty> URI="&ridos;designReport"/>
  <owl:Class URI="&ridos;Reports"/>
</owl:ObjectPropertyRange>

<owl:ObjectProperty> URI="&ridos;analyzeSReq"/>
  <owl:Class URI="&ridos;Analyst"/>
</owl:ObjectPropertyDomain>

<owl:SubClassOf>
  <owl:Class URI="&ridos;Database"/>
  <owl:Class URI="&ridos;DetailDesign"/>
</owl:SubClassOf>

<owl:Class URI="&ridos;SofConfigManager"/>
</owl:Declaration>

<owl:ObjectProperty> URI="&ridos;writeReqSpec"/>
  <owl:Class URI="&ridos;TechnicalWriter"/>
</owl:ObjectPropertyDomain>

<owl:ObjectProperty> URI="&ridos;analyseNFReq"/>
  <owl:Class URI="&ridos;Analyst"/>
</owl:ObjectPropertyDomain>

<owl:ObjectProperty> URI="&ridos;designReport"/>
  <owl:Class URI="&ridos;Reports"/>
</owl:ObjectPropertyRange>

<owl:ObjectProperty> URI="&ridos;analyzeSReq"/>
  <owl:Class URI="&ridos;Analyst"/>
</owl:ObjectPropertyDomain>

<owl:SubClassOf>
  <owl:Class URI="&ridos;Designer"/>
  <owl:Class URI="&owl;Thing"/>
</owl:ClassOf>

<owl:ObjectProperty> URI="&ridos;implementDB"/>
  <owl:Class URI="&ridos;Database"/>
</owl:ObjectPropertyRange>

<owl:SubClassOf>
  <owl:Class URI="&ridos;GUI"/>
</owl:ClassOf>
</owl:Ontology>
The sub-sets similar pairs (with different names/terms) of ontologies A1-A4 and B1-B2 are listed below:

A1.
(i) SQA Engineer; (ii) Systems Software Engineer; (iii) Senior Analyst; (iv) IT Network Manager; (v) Process Manager; (vi) Programmer
A2
(i) Technical Writer; (ii) Sr Software Engineer (iii) PHP Developer; (iv) Software Development Manager; (v) Software Engineer (vi) Software Quality Assurance Engineer; (vii) Coder
A3.
(i) Sr Web Developer; (ii) Senior Software Development Consultant; (iii) Software Design Engineer; (iv) Programmer; (v) Software Tester; (vi) Software Development Trainer; (vii) Senior Software Development Consultant
A4.
(i) Sr Web Developer; (ii) Senior Software Development Consultant; (iii) Software Design Engineer; (iv) SQA Engineer; (v) Systems Software Engineer; (vi) Senior Analyst; (vii) IT Network Manager; (viii) Process Manager (ix) Programmer; (x) Software Tester; (xi) Software Development Trainer; (xii) Director Software Development; (xiii) Development Lead Engineer
B1.
(i) Software Quality Assurance Manager; (ii) BIOS Development Engineer; (iii) Use Cases Engineer; (iv) Network Support Manager; (v) SOP Manager
B2
(i) Documenter; (ii) Senior Software Engineer; (iii) Web Developer; (iv) Software Manager; (v) Software Development Engineer
B3.
(i) Web Developer; (ii) Sr. Software Development Manager; (iii) Software Engineer; (iv) Component Engineer; (v) SQA Engineer; (vi) Senior Software Engineer; (vii) Senior IT Support Analyst; (viii) Documenter
B4.
(i) Web Developer; (ii) Sr. Software Development Manager; (iii) Software Engineer; (iv) Component Engineer; (v) Documenter; (vi) Software Manager; (vii) Software Development Engineer; (viii) SQA Engineer; (ix) Senior Software Engineer; (x) Senior IT Support Analyst; (xi) Configuration Manager; (xii) Lead Software Development Engineer