
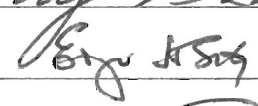


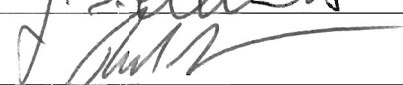




A FRAMEWORK AND METHODOLOGY FOR ONTOLOGY MEDIATION
THROUGH SEMANTIC AND SYNTACTIC MAPPING

by

Saravanan Muthaiyah
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Information Technology

Committee:

	Dr. Larry Kerschberg, Dissertation Director
	Dr. Edgar Sibley, Committee Member
	Dr. Daniel Menasce, Committee Member
	Dr. Jeremy Allnutt, Committee Member
	Dr. Frank Armour, Committee Member
	Dr. Daniel Menascé, Associate Dean for Research and Graduate Studies
	Dr. Lloyd J. Griffiths, Dean, The Volgenau School of Information Technology and Engineering

Date: 03/19/2008

Spring Semester 2008
George Mason University
Fairfax, VA

A Framework and Methodology for Ontology Mediation through Semantic and Syntactic
Mapping

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy at George Mason University

By

Saravanan Muthaiyah
Master of Science
University Putra Malaysia, 2000

Director: Dr. Larry Kerschberg
Department of Computer Science
The Volgenau School of Information Technology and Engineering

Spring Semester 2008
George Mason University
Fairfax, VA

Copyright 2008 Saravanan Muthaiyah
All Rights Reserved

DEDICATION

This thesis is dedicated to my loving wife Mala, my beloved daughter Shibhani and to my parents.

ACKNOWLEDGEMENTS

I would like to thank the US Department of State for giving me the prestigious Fulbright award that made it possible for me to come to the United States to embark on my doctoral studies. I would like to express my gratitude to my mentor and advisor Dr. Larry Kerschberg for all his guidance and support. I would also like to thank Multimedia University, for giving me all the support I needed during my studies. I would like to give special thanks to Dr. Don Gantz who provided me with teaching assistantship opportunities with the Department of Applied Information Technology on a continued basis. I would like to express my gratitude to the members of my doctoral committee Dr. Daniel Menascé, Dr. Jeremy Allnutt, Dr. Edgar Sibley and Dr. Frank Armour for giving me timely feedback that helped me improve my work. Lastly, I would like to thank my friends, colleagues and the staff of the GMU Office of International Programs and Services.

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF EQUATIONS	xi
ABSTRACT	xii
EXECUTIVE SUMMARY	xvi
1. Introduction	1
1.1 Background	4
1.2 Motivation	6
2. Literature Review	11
2.1 State-of-the-art of ontology mediation	15
2.1.1 Cooperative Framework	16
2.1.2 MAFRA Framework	16
2.1.3 OISs Framework	17
2.1.4 OntoMapO Framework	18
2.1.5 IFF Framework	18
2.1.6 FCA-Merge Method	19
2.1.7 IF-Map	19
2.1.8 PROMPT/SMART/PROMPT-DIFF	20
2.1.9 CHIMAERA, GLUE and CAIMAN	21
2.1.10 ONION	21
2.2 Summary of Methodologies	22
2.3 Conclusion	23
3. The Interoperability Problem	24
3.1 Structural Heterogeneity Problem	24
3.2 Semantic Data Heterogeneity Problem	26
3.3 Subjective Mapping Problem	27
3.4 Atomic, Inconsistency and Redundancy Problem	28
3.5 Summary	28
4. Ontology Mediation – early efforts and limitations	30
4.1 Domain Ontology Approach	30
4.2 Hybrid Approach	31
4.3 Multiple Ontology Approach	32
4.4 Need for an Overarching Approach and Thesis Goals	33
4.5 Summary	34
5. Ontology Mediation via Similarity Measures	37
5.1 Introduction	38
5.2 Theoretical Foundation and Assumptions	40

5.2.1 Syntactic Relatedness (SYN)	45
5.2.2 Semantic Relatedness (SEM)	46
5.3 Matching Algorithm.....	47
5.4 Benefits of Combining Syntactic and Semantic Measures	52
5.5 Measuring Similarity and Similarity Measures.....	52
5.5.1 Leacock-Chodorow Measure (LC).....	53
5.5.2 Resnik Measure (RS)	54
5.5.3 Jiang-Conrath Measure (JC).....	54
5.5.4 Lin Measure (LN).....	55
5.5.5 Hirst-St. Onge Measure (HS).....	55
5.5.6 PMI Measure (PM).....	56
5.5.7 NSS Measure (NS)	56
5.5.8 GLSA, LSA and SA Measure (SA)	57
5.5.9 WordNet ::Similarity Measure (WN).....	58
5.5.10 Gloss Vector (GV)	58
5.6 Similarity Relatedness Scores (SRS) and Similarity Function	59
5.6.1 Similarity Matrix and Binary Mappings	61
5.6.2 Similarity Matrix and Multiple Mappings	63
5.7 Summary	65
6. Ontology Mediation Framework.....	66
6.1 Introduction	66
6.2 Process Methodology	67
6.3 Prototyping Tools.....	71
6.4. Mapping Phases.....	73
6.4.1 Phase I – Semantic and Syntactic Agreement.....	73
6.4.2 Phase II - Affinity Measurement (SRS)	74
6.4.3 Phase III - Semantic Bridge	74
6.4.4 Phase IV - Semantic Consistency and Integrity Checking.....	75
6.5. Mediation Architecture Phases and Tasks	75
6.6 Summary	78
7. Mediation Architecture	79
7.1 Mediation Architecture Layers.....	81
7.1.1 User Layer.....	81
7.1.2 Search Layer.....	82
7.1.3 Semantic Layer.....	82
7.1.4 Data Layer.....	82
7.2 Ontology Mediation –Case: Security Policy Domain Model (SPDM).....	83
7.3 Understanding Security Policy.....	87
7.4 Mapping with Protégé and reasoning with RacerPro.....	90
7.5 Summary	94
8. Empirical Evaluation of SRS	95
8.1 Experiment Design.....	96
8.2 Survey Construct-Part I.....	97
8.3 Survey Construct -Part II	98

8.4 Data Analysis and Hypothesis Testing.....	100
8.4.1 Data Analysis – Survey Part I.....	100
8.4.2 Reliability Analysis.....	105
8.4.3 Reliability Function (Precision and Relevance Test).....	106
8.4.4 Hypothesis Testing - Reliability Function	107
8.5 Hypothesis Testing – Survey Part I.....	109
8.5.1 Pearson Product Moment Correlation.....	109
8.5.2 Nonparametric Correlation.....	110
8.6 Data Analysis - Survey Part II.....	113
8.6.1 Hypothesis Testing – Survey Part II	114
8.7 Major Findings	121
8.8 Summary	121
9. Semantic Bridging via SWRL.....	123
9.1 Case Study: Achieving Interoperability in E-Government Services via SWRL...	123
9.2 SWRL rules.....	125
9.3 Semantic Bridging with SWRL rules.....	126
9.4 Writing SWRL rules.....	127
9.4 Benefit of combining SRS with SWRL rules.....	129
9.5 Summary	131
10. Conclusion and Major Results	132
10.1 Thesis Summary.....	134
10.2 Future Research Direction.....	135
APPENDIX II: Survey Part 2.....	139
APPENDIX III: OWL results after reasoning and mapping.....	147
APPENDIX IV: Levenshtein Distance, in Three Flavors.....	150
APPENDIX V: Levenshtein Distance Algorithm.....	160
APPENDIX VI: Matching Algorithm and SWRL rules	161
REFERENCES.....	168
REFERENCES.....	169

LIST OF TABLES

Table	Page
1. Table 1 Mapping Phases	75
2. Table 2 Mediation Architecture Phases and Tasks	76
3. Table 3 Semantic Data Heterogeneity Problems.....	86
4. Table 4 Merging two SPRO Policy Data Elements	91
5. Table 5 Word pairs, SRS and HCR scores.....	100
6. Table 6 Case Summary Output for SRS and HCR scores.....	101
7. Table 7 Descriptive Statistics on SRS and HCR scores.....	103
8. Table 8 Case Processing Summary	105
9. Table 9 Reliability Analysis.....	106
10. Table 10 Pearson Product Moment Correlation.....	109
11. Table 11 Nonparametric Correlations	110
12. Table 12 Symbols, Word-Pairs and Scores	111
13. Table 13 Responses for Part A, B and C.....	115
14. Table 14 HCR Scores for Part A, B and C.....	116
15. Table 15 Part D –HCR Rank Score.....	118
16. Table 16 Part E –HCR Rank Score	119
17. Table 17 Comparing Part D and E	120

LIST OF FIGURES

Figure	Page
1. Figure 1 Structural Heterogeneity Problems	25
2. Figure 2 Semantic Data Heterogeneity Problem.....	26
3. Figure 3 Subjective Mapping Problems.....	27
4. Figure 4 System Level Interoperability of Atomic Data Storage Problem	28
5. Figure 5 Domain Ontology Approach.....	30
6. Figure 6 Hybrid Approach	31
7. Figure 7 Multiple Ontology Approach.....	32
8. Figure 8 XMapper System Architecture	35
9. Figure 9 Achieving Semantic Correspondence via Mappings	40
10. Figure 10 Matching Algorithm	49
11. Figure 11 Similarity Matrix.....	62
12. Figure 12 Similarity Measures for Binary Mappings.....	63
13. Figure 13 Similarity Measures for Multiple Mapping	64
14. Figure 14 Process Methodology for Ontology Mediation	67
15. Figure 15 Local ontology classes for white wine.....	68
16. Figure 16 Concepts shared from upper ontology.....	69
17. Figure 17 Updated classes in shared ontology.....	70
18. Figure 18 Prototyping Tools for the Process Methodology	72
19. Figure 19 Stage II - Reasoning Phase	77
20. Figure 20 Stage III-Mapping in PROMPT.....	77
21. Figure 21 Semantic Mediation Architecture (SMA).....	83
22. Figure 22 Security Policies –Structural Heterogeneity Problem	86
23. Figure 23 SPDM Process Methodology.....	89
24. Figure 24 Mapping results for scenario A.....	93
25. Figure 25 Case Summary Output for SRS scores	104
26. Figure 26 Case Summary Output for HCR scores	104
27. Figure 27 Comparing the Reliability SRS and Syntactic Scores	108
28. Figure 28 Comparing SYN to HCR Scores	112
29. Figure 29 Comparing SRS to HCR Scores	112
30. Figure 30 Part A, B and C for HCR Rank Score	116
31. Figure 31 Part A and B HCR Rank Score.....	117
32. Figure 32 Part A HCR Rank Score	117
33. Figure 33 Part D HCR Rank Score (Staying 1 st class only).....	118
34. Figure 34 Part E- HCR Rank Score (Outdoor activities).....	119
35. Figure 35 Part D and E – HCR Rank Score.....	120
36. Figure 36 DMV license renewal process	124

37. Figure 37 Semantically Bridging DMV Records and DMV Licence Renewal ontologies via SWRL rules	127
38. Figure 38 Implementing SWRL Rules.....	130

LIST OF EQUATIONS

Equations	Page
1. Jaccard Similarity Coefficient - $J(A, B) = P(A \cap B) / P(A \cup B)$ (1)	39
2. Jaccard Distance = $[P(A \cup B) - P(A \cap B)] / P(A \cup B)$ (2).....	39
3. SimName (Cc,Co) = $1 - (\text{Lev}(\text{CcName}, \text{CoName}))$ (3)	46
4. $S(f_x)$ where, $x = \{E, IC, CN, D, SYN, SEM\}$ (4).....	47
5. $\text{sim}_{LC}(c1, c2) = -\log \text{len}(c1, c2)$ (5).....	54
6. $\text{sim}_R(c1, c2) = -\log p(\text{lso}(c1, c2))$ (6).....	54
7. $\text{dist}_{JC}(c1, c2) = 2\log(p(\text{lso}(c1, c2))) - (\log(p(c1)) + \log(p(c2)))$ (7)	55
8. $\text{sim}_L(c1, c2) = 2 \times \log p(\text{lso len}(c1, c2))$ (8).....	55
9. $\text{rel}_{HS}(c1, c2) = C\text{- path length} - k \times d$ (9)	55
10. $\text{PMI}(c1, c2) = \log_2 P(c1, c2)$ (10).....	56
11. $\text{NGD}(c1, c2) = \max \{\log f(c1), \log f(c2)\} - \log f(c1, c2)$ (11).....	57
12. $\cos\theta_{xy} = x.y / x y $ (12).....	57
13. $\text{SA}(w1, w2) = \log P(X=1 Y=1)$ (13)	58
14. $\text{SRS} = f_x \{LC, RS, JC, LN, HS, PM, NS, LSA, WN, GV, SYN\}$ (14)	60
15. $\text{SRS} = f_x \{LN, LSA, WN, GV, SYN\}$ (15)	61
16. $\text{REL} = \{P_s \text{ and } R_L\}$ (16)	106
17. P_s = number of correct responses (17)	106
18. R_L = number of relevant responses (18).....	106

ABSTRACT

A FRAMEWORK AND METHODOLOGY FOR ONTOLOGY MEDIATION THROUGH SEMANTIC AND SYNTACTIC MAPPING

Saravanan Muthaiyah, Ph.D.

George Mason University, 2008

Dissertation Director: Dr. Larry Kerschberg

Ontology mediation is the process of establishing a common ground for interoperability between domain ontologies. Ontology mapping is the task of identifying concept and attribute correspondences between ontologies through a matching process. Ontology mediation and mapping enable ontologists to borrow and reuse rich schema definitions from existing domain ontologies that have already been developed by other ontologists. For example, a white wine distributor could maintain a white wine ontology that only has white wine concepts. This distributor may then decide at some point in the future to include other wine classifications as well in his current ontology. Instead of creating red wine or desert wine concepts in his existing ontology, the distributor could just borrow these concepts from existing red wine and desert wine ontologies. As such ontology mapping becomes necessary.

The practice of matching ontology schemas today is one that is labor-intensive. Although semi-automated systems have been introduced, they are based on syntactic matching algorithms which do not produce reliable results. Thus my thesis statement is that *the hybrid approach i.e., Semantic Relatedness Score (SRS), which combines both semantic and syntactic matching algorithms, provides better results in terms of greater reliability and precision when compared to pure syntactic matching algorithms.*

This research validates that SRS provides higher precision and relevance compared to syntactic matching techniques that have been used previously. SRS was developed through the process of rigorously testing thirteen well established matching algorithms and choosing a composite measure of the best combination of five out of those thirteen measures. This thesis also provides an end-to-end approach by providing a framework, process methodology and architecture for the process of ontology mediation.

Since implementing a fully automated system without any human intervention would not be a realistic goal, a semi-automated approach is undertaken in this thesis. In this approach, an ontologist is assisted by a mapping system which selects the best candidates to be matched from the source and target ontology using SRS. The goal was not only to reduce the workload of the ontologist, but also provide results that are reliable. Literature survey on current ontology mediation research initiatives such as InfoSleuth, XMapper,

ONION, FOAM, FCA-Merge, KRAFT, CHIMERA, PROMPT and OBSERVER, among others, revealed that the state-of-art of ontology mediation is to a large extent based on mainly syntactic schema matching that supported binary schema matches (1:1) only.

A generic solution for schema matching based on SRS is presented in this thesis to overcome these limitations. A similarity matrix for concept similarity measures is introduced based on several cognitive and quantitative techniques such as computational linguistics, Latent Semantic Analysis (LSA), distance vectors and lexical databases (WordNet). The six-part matching algorithm is used to analyze RDF, OWL and XML schemas and to provide a similarity scores which are then used to populate a similarity matrix. The contribution here is twofold. Firstly, this approach gives a composite similarity metric and also supports complex mappings (1:n, 1:m, m:1 and n:m). Secondly, it provides higher relevance, reliability and precision.

The validation of this approach is demonstrated by comparing SRS results with that of human domain experts. Empirical evidence provided in this document clearly shows that the hybrid method resulted in a higher correlation, better relevance and more reliable results than purely syntactic matching systems. Predefined Semantic Web Rule Language (SWRL) rules are also introduced to concatenate attributes, discover new relations and enforce the assertion box (ABox) instances.

Reasoning for consistency, coherence, ontology classification and inference measures are also introduced. An actual implementation of this framework and process methodology for the mapping of security policy ontologies (SPRO) is provided as a case study. Another case study on achieving interoperability for e-government services with SWRL rules is also presented. Both SRS and SWRL rules are highlighted in this document as being complementary measures for the process of semantic bridging. Several tools were used for a proof-of-concept for the implementation of the methodology, including Protégé, Racer Pro, Rice and PROMPT.

EXECUTIVE SUMMARY

The main obstacle to data interoperability is data heterogeneity, where similar source data is represented differently using different naming conventions and structures. This thesis addresses the problem of heterogeneity between ontologies in the context of the Semantic Web. Domain ontologies have become an integral part of the Semantic Web and as their usage increases, the need for resolving semantic differences among them becomes very important. Ontologies *per se* do not solve interoperability problems, because the conceptualizations they represent are not commonly shared and agreed by everyone. In order to realize the Semantic Web vision, disparate ontological representations must be connected through mappings and mediation.

The goal of this thesis is to provide both a framework and a methodology for ontology mediation. This is to enable disparate schemata of ontologies to be bridged in a semi-automated way via a hybrid technique that combines both syntactic and semantic matching algorithms. Although many proposed solutions for ontology mediation exist, they are either too dependent on manual human input or heavily dependant on mathematical theorems which leave out the human element. Methods that include human expertise at all stages of the mediation process on the other hand, do not scale well for larger ontologies. Some other works are database oriented which cater to mainly

structured data and are not suitable for ontology mediation. On the whole, the techniques reviewed seem to provide solutions to only one aspect of the mediation process. The approach presented in this thesis is semi-automated and relies on human expertise in the final stage of the mediation process.

A process methodology is introduced which consists of the ontology selection process, semantic and syntactic equivalence analysis, conflict resolution, inclusiveness test, disjoint test and consistency test. A detailed match algorithm is specified and implemented and this is another important contribution. The process methodology has a six-step approach, which includes a reasoning process for equivalency checks, consistency checks and integrity check. After all inconsistencies, conflicts and integrity checks have been completed with the aid of a reasoning engine, the semantic alignment process is initiated. The methodology proposes that the consistency checks be performed before and after mapping to ensure that the original sets of data, concepts and instances remain consistent between source ontology (SO) and target ontology (TO).

This pre-consistency and post-consistency checking is a unique feature of this approach. A hybrid model which combines syntactic and semantic mediation has been introduced for computing similarity (i.e., Similarity Relatedness Scores). Unlike existing methods, a domain expert's input is only required at the end of the process after SRS scores have been produced. This unique feature reduces the processing time of the human expert who formerly had to analyze manually all the candidate data labels for matching. With this

new approach, similarity scores determine which candidates are more likely to be matched and thus filters extraneous data. A threshold value is set for the scores and data labels, and those scores above the threshold value are automatically selected for matching. The semantic engine component in the proposed architecture is based on this hybrid model.

A mediation architecture is introduced in this thesis. It has four layers: user layer, search layer, semantic layer and data layer. The user layer processes posted queries and connects to search agents, which then pass requests containing information such as URI, ontology, and keyword, etc to the broker agent. Broker agents are connected to the semantic engine, which houses the reasoning engine, WordNet lexical database and a mapping agent. The tasks of the semantic engine also include conflict analysis, resolution and mapping. A unique feature of the semantic engine is the WordNet agent interface that determines semantics or meanings. This allows for computation of meanings based on word senses such as synonyms, hyponyms, etc.

Another unique feature is how the match agent computes SRS values. It uses a combination of linguistic algorithms such as Lin (LN), Gloss Vector (GV), LSA (Latent Semantic Analysis) and WordNet Vector-UMN (WN). This combination is empirically proven to give the best results. The associative strength values are computed based on SRS and the scores are entered into a similarity matrix. The similarity matrix is flexible and supports binary (1:1) as well as complex mappings (1:n, 1:m, m:1 and n:m). This is another important contribution because most mediation methods today focus on binary

matches and do not measure cognitive relationships between and among concepts that are matched. The use of computational and cognitive means for determining the association strength between concepts have been well tested in essay grading algorithms and have been widely applied for decades in a number of areas such as text completion and TOEFL grading algorithms. This further validates the approach taken in this thesis.

To validate the applicability of the framework and methodology, empirical tests were conducted based on several hypothesis. Based on studies done by Miller and Charles of Princeton [1], a research questionnaire was distributed to human subjects. The entire premise of the study was to test if the hybrid model would provide results that were consistent with results provided by domain experts. The higher correlation indicates greater consistency and accuracy. The cognitive responses that were collected from human domain experts validate the study with results showing a high degree of positive correlation (i.e., 92%) between human scores and the combined scores of the hybrid model.

Since the science for semantic similarity measures emerged from Word Sense Disambiguation (WSD) and Information Retrieval (IR), the measures that would normally be used for evaluation are precision, recall and f-measure. However given the nature of the data and the approach of this thesis, such measures are not totally useful. Therefore, this thesis introduces a modified evaluation measure, which are comprised of precision (P_s), relevance (R_L) and a combined measure of P_s and R_L called reliability

(REL). In line with findings in cognitive science, which states that a combined measure resulted in higher precision, the combined scores of both syntactic and semantic scores provided higher precision (40%) and relevance (96.67%) compared to pure syntactic measures which had only (16.67%) for precision and (73.33%) for relevance, respectively. The approach used here and the modified measures proposed for evaluation is in itself, a significant contribution of this thesis.

A two-part survey was designed. A total of 50 questionnaires were distributed to domain experts and 50 responses were received, giving the study a 100% response rate. However, only 38 responses were actually used for data analysis and this was because of 12 incomplete responses that had to be filtered out. The first part of the survey was dedicated to the discussion above. The second part was oriented towards providing empirical evidence that similarity between concepts can change when the context in which the concept is used changes. Five sections were introduced for the second part where each section provided a different context with the same number of options. The tests revealed that respondents made changes to their cognitive rankings when contexts were changed. This supports the hypothesis that context evaluation is an important feature to be included for ontology mediation. Mean and standard deviation scores for the ranks provide further evidence that rankings were affected by changes in contexts. This thesis emphasizes the need for context analysis to be added as an additional feature to measure semantic relatedness.

CHAPTER 1

1. Introduction

Tim Berners-Lee's vision of a *global reasoning* web is commonly referred to as the Semantic Web. The European Commission has adopted it in their national agenda, which is the Sixth Framework Program. In the US it is largely linked to research projects of DARPA (Defense Advanced Research Projects Agency) and DoD (Department of Defense). Ontology usage and development has also slowly gained importance as part of this quest. In philosophy, ontology is defined as the *knowledge of existence*. Others say it is a set of definitions of formal vocabulary [2]. The ontology concept is said to be a systematic account of existence [3]. Nevertheless, the knowledge of existence cannot just be described by a single ontology. As such, finding a common ground for interoperability (i.e. mediation) between domain ontologies is the only way to achieve this goal.

A global ontology according to some researchers would be the closest to realizing this for the Semantic Web. To achieve a global ontology, domain ontologies should be mapped. However, mapping of these ontologies is not a trivial task. This is because every ontology

has its own data definitions and specifications and would not in general, be compatible. To achieve compatibility, data heterogeneity amongst these ontologies must first be resolved. This problem has a historical link to the problem of finding compatibility among disparate systems in enterprise information systems (EIS). This is usually referred to as platform integration of information systems and databases. Researchers and practitioners have long before dealt with similar issues, which revolve around the theme of incompatibility. The objective of resolving data heterogeneity amongst different data sources in enterprise systems and database schemata integration efforts is analogous to the need for achieving ontology interoperability in the Semantic Web.

The main objective of the Semantic Web is to enable systems to exchange information and services seamlessly with one another in semantically rich ways via machine understandable web resources [4, 5]. As such, rich representations of data via ontologies and taxonomies are necessary for creating rich semantics and meta-data. Web resources must be annotated with meta-data so that they could be correctly discovered and invoked. Agent ontology systems can then interact with other agents via automated service discovery to share data and provide web services on-the-fly. The advent of Semantic Web technologies and the scale of its growth have resulted in the creation of ontologies that are incompatible. The main obstacle for achieving seamless interaction is due to the fact that data is being represented differently amongst existing ontologies.

Ontologies are meant to provide a shared conceptualization of various domains and agent ontology systems would utilize them to cater to web services requests. Although

ontologies provided a shared conceptualization for various domains of knowledge, the possibility for a single ontology to support all the definitions of various domains is clearly impossible. As such many ontologies for similar domains co-exist, creating a plethora of data, definitions and concepts that have to be integrated in some fashion so that a shared conceptualization can exist. To address this problem, integration of metadata must be implemented with well-defined ontology alignment techniques coupled with proper integration tools. Such alignment procedures are commonly referred to as ontology integration, mediation or mapping. This is important to achieve compatibility between heterogeneous ontology domains [3-6] which would then bring us closer to realizing the Semantic Web objective discussed earlier.

There is a lot of emphasis for the need of data integration for ontologies in the literature. Significant efforts are being carried out at the enterprise level, towards establishing a Reference Ontology (RO) of some kind to mitigate the problem. FEARMO¹ is a good example of such efforts initiated by the DoD in leveraging their network-centric architecture. The US government has implemented similar efforts for their federated services as well. However, the question of how ontology mediation should be done is still an ongoing research topic and several methods have been proposed in recent years [5-10]. Some highlight the importance of reasoning [6-9] others highlight declarative specification of mappings [10], some provide comprehensive support to mappings between classes and slots via ConceptBridge and AttributeBridge [11].

¹ Federal Enterprise Architecture Reference Model Ontology (<http://www.osera.modeldriven.org>)

An overarching approach is still missing on how to handle this problem, which is the focus of this research. A mapping methodology, bridging framework and architecture is proposed in this thesis. Semantic reasoning, conflict checks, integrity checks and consistency checks are also introduced. The new architecture proposed, highlights ontology mediation with emphasis on semantics. It will serve as a key interoperability enabler for the Semantic Web.

1.1 Background

Ontologies are concerned with description of concepts and relationships that exist for an agent or a community of agents. It is generally written as a set of definitions using a formal vocabulary. A domain-specific ontology (i.e., domain ontology) specifies meanings and terminology for a given domain. The meanings quite often differ between those domains. For example the word “virus” has different meanings in a “human disease” domain ontology as compared to “computer attack” domain ontology. Domain ontologies provide a glossary of terms, which are widely applied to a range of specified high-level domains or global ontologies (i.e., Dublin Core², GFO³, OpenCyc⁴/ResearchCyc⁵, SUMO⁶, WordNet⁷ and DOLCE⁸). The lower ontologies are

² Metadata initiative for the development of interoperable online metadata standards (<http://dublincore.org>)

³ Genealogical Forum of Oregon (<http://www.gfo.org>)

⁴ An open source version of the Cyc technology (i.e. knowledge base and commonsense reasoning engine)

⁵ Non proprietary Cyc knowledgebase (<http://research.cyc.com>)

⁶ Suggested Upper Merged Ontology, that promotes data interoperability, search, retrieval, inferencing, and natural language processing (<http://ontology.teknowledge.com>)

⁷ Lexical Database for the English Language (<http://wordnet.princeton.edu>)

⁸ Descriptive Ontology for Linguistic and Cognitive Engineering (<http://www.loa-cnr.it/DOLCE.html>)

more specific to the target domain application. As such, it causes incompatibility and presents a great challenge to the ontology engineer who actually needs general representations.

Different ontologies within the same domains also have variations because ontologists who design them, see the world from different viewpoints. Ontology designers, who are just human and have diverse backgrounds, create ontologies based upon their individual backgrounds, work experience, knowledge, education, culture and ideologies. This causes the variations among the ontologies created. Efforts to solve the incompatibility problem via integration or mediation are largely performed manually. This is very time consuming and quite expensive for the ontology integrator. Good ontology mediation techniques are crucial for building semantic bridges between ontologies. Commonly agreed standards do not exist for ontologies at present in the manner that we have for enterprise systems such as EDI and XML. Even if a RO did exist, researchers currently argue that it would be inflexible and would not be “semantically rich” [12]. Work in this area is largely theoretical at present and there is a critical need for better methods, methodologies, tools and frameworks for resolving incompatibility issues via ontology mediation.

Ontology mediation is one way to achieve compatibility among ontologies. However, current methods suffer from a number of problems. Most techniques do not provide formal semantics for mapping data structures. They also rely heavily on string-based and structure-based similarity measures, which often times fail to produce accurate mappings.

This thesis attempts to address these problems by introducing an end-to-end framework with a new mapping framework and bridging architecture. Semantics for concept matching via SRS is at the core of the proposed “bridging architecture”. Current ontologies are meant to provide enough semantics for machine processable data, but sadly, decentralized creation of these ontologies causes heterogeneity among data labels within ontologies. Efforts to “bridge” the data labels continue but there is no clear end-to-end framework or methodology for it. As such the research presented in this thesis can be said to be timely.

1.2 Motivation

Ontologies provide rich expressions of a knowledge domain by specifying meanings and expressions. Such specifications encompass data models, concepts, schemata and data sets. Ontologies are being used in the areas of electronic commerce and web services. Implementers are also creating their own ontologies for similar domains, which raises the level of data heterogeneity problems. The Web Ontology Language (OWL) is useful for the design and creation of ontologies but does not help to integrate disparate ontologies.

Intuitively speaking, it is always easier to create OWL-based ontologies and classes for our own needs, rather than integrating existing ontologies into our ontologies, due to reasons such as different naming conventions used, diversity in philosophical orientation, as well as conceptual framework and domain nuances that exists in the mind of ontologists. There will always be design biases and tradeoffs amongst different ontology

structures. As such, integrating across such diverse views of the world and its underlying constraints is not a trivial task. A plausible solution is ontology mediation where correspondences between related entities are semantically established. Correspondences are useful for ontology merging, query answering and data translation. Thus, by matching ontologies, one enables the interoperation of knowledge represented in the matched ontologies. However, assuming that various classes or properties are equivalent may not be accurate enough because the inherited values of those classes and properties might not have a precise semantic match. Classes could be semantically in conflict, even though they seem to have equivalent concept names. If OWL were more expressive in defining mappings between ontologies, one could resolve such nuances. An alternative to mapping would be importing external ontologies into extant ontologies but, as mentioned previously, it might make things worse.

It would be desirable to conveniently reuse existing ontologies with one's own ontologies. Although this is theoretically possible, practically it is difficult. As such, this research's goal is to develop a framework, architecture and methodology to allow the reuse of ontologies [13] for ontology mediation. As mentioned earlier, two ways to solve this problem is to connect ontologies together or have everyone use the same ontology (i.e., the upper or global ontology). Currently, there are no unified solutions for either. The former is a better approach and is in line with what other researchers are doing. It would bridge concept definitions of different ontologies. The latter is an unrealistic goal as discussed previously.

Tim Berners-Lee's vision of the Semantic Web is that someday there will be thousands of ontologies containing millions of instances, and somehow they would be integrated, or at least if they were not integrated at the semantic level, there would be some "magic sauce" that would enable the integration. It is also envisioned that someday, intelligent agents will be able to freely and seamlessly roam around the Semantic Web to harvest, integrate and store the data into knowledgebases with reasoning performed across them. Semantic interoperability is really the crux of this requirement which enables machine-understandable applications to share, reuse and exchange metadata.

The Semantic Web vision necessitates a robust architecture, framework, methodology as well as integration tools for mapping among ontologies. Mapping tools are critical because reusing ontologies may be impractical, especially in the case of large and complex ontologies. Ontologists would rather write their own internally consistent ontologies and map them to other ontologies rather than importing other ontologies into what they have created so that they can avoid dealing with inconsistencies that will arise after its implementation.

The requirements for good semantic integration tools are numerous. Most significant among them is that such tools need to move beyond merely helping with integration between two ontologies and also help an ontologist map their ontology to other ontologies. Tools must also provide error checking, consistency checking and integrity checking capabilities focusing on logical problems and inheritance incompatibilities that

may arise in complex mappings. They must also identify classes and properties that should have been mapped but were missed. Perhaps by analyzing instance data from different ontologies (such as different ontology's representation of the same unique entities or concepts), these tools could even learn or suggest mappings in order to assist or automate the mapping process to some degree. This would bring about semantic integration and full interoperability on the semantic web⁹. In summary, the Semantic Web dream is not really far fetched.

It requires metadata, machine-processable and understandable information to be freely exchanged. Bridging gaps between ontologies is crucial for this, especially for similar domain ontologies. However, bridging ontologies that are not from similar domains is also very much desired. The survey of literature shows that syntactic matches are performed widely to bridge ontologies and depend largely on human input. Clearly this will neither be flexible nor scalable for the Semantic Web.

The question now is, what do we do to rectify this situation? There are several options to be considered here: 1) we could wait for a W3C standard to specify the structure needed for all ontologists to follow, 2) use syntactic matching and allow independent ontology mediators to worry about their own matching needs, 3) resort to using a global ontology (federated approach), 4) merge local ontologies with global ontologies and 5) maintain local ontologies independently but use a "bridge" to map data labels with global

⁹ Source: http://novaspivack.typepad.com/nova_spivacks_weblog/2006/08/the_ontology_in.html

ontologies for our own needs when needed. Obviously, the first option would be good but would not be timely for the Semantic Web. The second option is simple but lacks scalability. The third option is expensive, complex, biased and difficult to maintain. However, the third option would be good in the long run. The fourth option is effective but expensive.

The best option is the fifth, even though it is expensive and requires real time processing. Meaningful matches based on concept meaning (i.e. semantics) instead of coarse matching (i.e. syntactic matching) would be the best solution for bridging data labels of local ontologies with global ontology definitions. With the existence of corpus data, lexical databases (i.e., WordNet) and cognitive measures, more accurate match of data labels can be achieved. The higher the accuracy in matches, the greater the probability of accurate “bridging” and thus the closer we can get to the Semantic Web dream. Empirical tests carried out in this research validates that this can be done. Experimental details and results obtained are presented in chapter 8.

CHAPTER 2

2. Literature Review

Rapid growth of information system platforms creates disparate technological resources and the growth of heterogeneous data makes data sharing and integration more difficult. Data heterogeneity among ontologies also suffers from the same problem. The vision for the Semantic Web has raised high expectations and to achieve them the heterogeneity problem must be addressed at once [4, 5]. Several integration methods were surveyed to identify the appropriate approach to ontology mediation. One such effort is the Grid model. It brings together dispersed data and information spaces via a common platform called the Grid infrastructure. Virtual Organization (VO) that participate and interact on the Grid can seamlessly inter-exchange and collaborate on data, thereby resolving much of the disparity issues. Although this method provided some important facts, it did not show how ontologies could be matched.

Graph based solutions have also been used in the effort of creating semantic bridges. The entire theory is based on semantic correspondence between concepts that are tied together based on their node locations on the graph. Graph-like structures are used to highlight

heterogeneity amongst data in the graph hierarchy. The hierarchy can also be observed as a taxonomy hierarchy. Concepts are matched based on the nearest neighbor approach for similarity matching. A match operator takes two graph-like structures (e.g., database schemas or ontologies) and produces mappings between their data elements in the form of graphs. The graphs that correspond syntactically to each other are then generated. The problem with this method is that it lacks flexibility.

The InfoSleuth project uses Inductive Machine Learning (IML), a revolutionary approach that provides good results. It uses an agent architecture for creating semantic mappings [7, 14]. The Semantic Web is viewed in the same manner as the Grid where various layers exist and heterogeneous security policies exist among them. Heterogeneous data labels use a reference ontology (RO) [9]. Such access-level integration was only used in commercial tools and the real need was for semantic data-level integration. IML was successfully implemented in their agent architecture for semantic integration of web based information resources [14]. However, this method is based purely on an XML representation of data sources, and does not scale for the Semantic Web environment.

Semantic interoperability via a Reference Ontology (RO) and Semantic Annotation Language (SAL) provided the basis for early semantic mapping efforts, and provided better results than previous efforts. There are two essential issues for achieving semantic interoperability. First, is the identification of semantically-related data with subsequent resolution of their schematic differences. Second, is the access to and usage of large

autonomous databases without prior knowledge of their content [9]. The first issue involves making semantics explicit and this depends on the context in which a data object is used, and its contextual representation [15]. The second issue requires users to be familiar with the content and structure of the information sources in order to be able to specify a query. The former is necessary for ontology mediation, while the latter is not. This approach provides a uniform method for query translation and heterogeneity resolution in a multidatabase environment [13, 19]. Since the Semantic Web has to deal with more than just structured data sources, this technique was not sufficient.

Capturing explicitly the semantic content of the individual databases is another important challenge. It is important to understand the semantics of each schema component and to capture and reason them using semantics. Semantic interoperability is the ability for disparate system domains to comprehend meanings and terminologies via axiomatic mapping of agreed-upon concepts to create semantically compatible information. Establishing semantic interoperability among heterogeneous and disparate information sources has been a critical research area within the database community for the past two decades [16-18].

However, semantic data models do not quite capture semantics of a database, and the meta-information (i.e., tacit knowledge) captured during its design phase is not explicitly represented in the resulting database. Hence, such information cannot be completely accessible to applications, queries, or users. As such, semantic data models that capture

domain meta-information (i.e., entity classes, relationships, constraints, cardinalities, etc.) alone are not enough to support semantic interoperability among heterogeneous databases [19].

Two widely used approaches for semantic interoperability are the federated schema approach (i.e., domain ontology approach) and the tacit knowledge capture approach. The former attempts to construct a global schema and establish mappings between federated schemas and participating local schemas. However, the drawback of this approach is its lack of semantic richness and flexibility [15, 21, 23]. The latter tries to solve the problem of lack of semantic richness by capturing the tacit knowledge within a certain domain in great detail in order to provide a rich conceptualization of data objects and their relationships [18, 20]. Even though it is theoretically valid, in practice it is not feasible due to the inherent complexities of the knowledge domain. Hence, it can only serve restricted application domains. This limits its general applicability. Thus a hybrid approach was initiated [20].

A hybrid approach uses a common ontology, which specifies a vocabulary to describe and interpret shared information among its users. This approach is similar to the federated schema approach as it has a high-level domain model playing the role of shared schema while ensuring the autonomy of the local schemas. However, a domain model is different from the conventional federated schema because domain knowledge captured in the domain model is generally represented in logic using the vocabulary provided by the

ontology editor. An ontology-based domain model captures much richer semantics and covers a much broader range of knowledge within a target domain [12, 20]. This approach provides a simple formalism to capture only the domain knowledge pertaining to potential semantic conflicts. The advantage of this simplified ontology is that it is not domain-specific and does not lose any semantic richness.

The researchers stress that using both a common ontology and a semantic data model provides a more complete understanding of the application domain [20]. Approaches introduced for ontology mapping specifically focus on certain shortcomings only and are disjointed [11, 14, 21-23]. For example reasoning, declarative mapping specifications, mappings between classes and slots are highlighted in [4, 9, 15]. Conflict in mappings is highlighted in [18]; however, it does not address conflict resolution protocols for the Semantic Web. Others focus mainly on database related structures and are not relevant for ontology structures [8]. As mentioned before, there is a need to extend these techniques to resolve ontology mapping problems.

2.1 State-of-the-art of ontology mediation

In this section, a review of current ontology mediation applications and methods is presented. The survey of existing methods is significant as it helps us to better understand the landscape of ongoing ontology mediation work. More than 15 specific technologies and frameworks have been studied and surveyed for this section. In chapter 4 these

frameworks are classified into three approaches, i.e., domain approach, hybrid approach and multiple approach.

2.1.1 Cooperative Framework

Fernandez and Martinez-Bejars introduce a model for ontology integration called the cooperative framework [24]. The algorithm proposed is meant to create a global ontology. The model is intended for two groups of users - normal and expert. The former seeks information and provides specific information of concepts and the latter integrates the ontology in the author's nomenclature. The algorithm is based on taxonomic features and synonym detection of concepts from the source ontologies (SO). Attributes of concepts can also be defined in this framework and the algorithm integrates attributes of the same concept. However, both the concepts that are to be integrated (e.g. PERSON and PEOPLE) must possess the same exact attributes (i.e. age and name). This rigid criterion makes the model less flexible.

2.1.2 MAFRA Framework

Maedche and Staab introduce a mapping framework called MAFRA [11]. It caters for distributed ontologies in the Semantic Web. The authors of this framework argue that mapping of existing ontologies is easier than creating a common ontology. This supports my discussion earlier on why the reference ontology (RO) would be hard to establish. The reason for this is two fold. Firstly, only a small community is actually involved in this process. Secondly, it would be difficult to coordinate and agree upon all the design

activities before ontologies are introduced into the Semantic Web environment. MAFRA is part of a multi-ontology system and aims to automatically detect similarity between entities of separate ontologies. Ontologies are normalized into RDFS format to eliminate syntax differences and to make SO (source ontologies) and TO (target ontologies) more apparent. This is done via a tool called LIFT, which is capable of taking DTDs, XML schema and relational databases and normalizing them to the structural level of the ontology. This framework also introduces the semantic bridge for matching concepts (ConceptBridge) and attributes (AttributeBridge). Details on algorithms however were not available for analysis.

2.1.3 OISs Framework

Calvanese and colleagues propose a framework called ontology integration systems (OISs) [25]. The framework is based on Description Logic (DL) knowledge bases and mappings are expressed via queries. There is no explicit mechanism for the notion of queries. Mapping of concepts into ontology views is first achieved based on the query results. Two approaches for query views are introduced i.e. global-centric and local-centric. Associating each relation in the global schema to one relational query over the source relations specifies the global-centric approach to mapping. The local-centric approach requires reformulation of query in terms of the queries to the local sources. The authors provide examples of using both approaches. The technique is analogous to integration of relational databases. It presents views but doesn't show how unstructured data could be matched.

2.1.4 OntoMapO Framework

Kiryakov and colleagues developed a framework for accessing and integrating upper level ontologies called meta-ontology (OntoMapO) [26]. It allows a user to import linguistic ontologies onto a Web server, which will then be mapped onto other ontologies. A uniform representation of the ontologies with mappings and a simple meta-ontology of property types and relation-types should be defined. Two sets of primitives are defined, InterOntologyRel and IntraOntologyRel, each of which has a number of relations that capture the correspondence of equivalent concepts of different ontologies. The authors claim that an initial prototype had been used to map parts of the CyC ontology to EuroWordNet. The framework does not however show how equivalence of concepts is measured.

2.1.5 IFF Framework

Kent introduces the Information Flow Framework (IFF) to support ontology sharing [27]. It is based on channel theory of Barwise and Seligman [28]. Kent exploits the distinction made in channel theory to formally describe the stability and dynamism of conceptual knowledge organization. There are two basic assumptions and a two-step process in this model. The framework is purely theoretical and there is no method for implementing the two-step process. There are no explicit definitions available for ontology mapping; only an implicit definition exists, which Kent refers to as the *Chu Transform* or in other words, a knowledge-sharing scenario.

2.1.6 FCA-Merge Method

Stumme and Maedche are authors of FCA-Merge [29]. It's a method for ontology merging which was based on Formal Concept Analysis [30]. The method uses concept analysis and lattice exploration with natural language techniques. Lattice concepts are derived with natural language techniques and then a knowledge engineer explores it manually. FCA-Merge provides semi-automatic guidance for the knowledge engineer to build a merged ontology. Input is given by a set of documents from which concepts and ontologies to be merged are extracted. These documents should be representative of the domain to be merged.

This step is also called the *population mechanism*. A concept lattice is then derived with the aid of lexical analysis. This helps the merging of single words (e.g. Hotel) to complex ones (e.g. Hotel Merlin). These two concepts are merged to generate a *pruned lattice*. An algorithm (i.e. TITANIC) is used for this purpose. Disambiguation via indexing and finally the construction of a merged ontology is carefully done with human interaction. The only drawback is that merging of source ontologies (SO) prevents them from being maintained independently.

2.1.7 IF-Map

Kalfoglou and Schorlemmer are the authors of IF-Map, an automatic method for ontology mapping [31]. It is also based on the channel theory of Barwise and Seligman [28]. This method provides a systematic way for ontology mapping based on

infomorphisms (i.e. transforms data while preserving its meaning). It has a four-step process: 1) ontology harvesting, 2) translation, 3) infomorphism generation and 4) display of results. Existing ontologies are first downloaded from libraries such as Ontolingua and WebOnto for step 1. Horn logic is used for step 2 and the harvested ontology formats are translated into Prolog clauses. In step 3 infomorphisms between ontologies if any, are displayed in RDF format. Results are stored in a knowledge base for future use and this completes step 4. This is done for future reference and maintenance.

2.1.8 PROMPT/SMART/PROMPT-DIFF

Noy and Musen have developed several tools for ontology mapping, alignment and versioning. SMART [32], PROMPT [33] and PROMPTDIFF [34]. All of them are available as plug-ins for the open source ontology editor, Protégé [35]. These tools perform linguistic similarity matches between concepts and then use Protégé for discovering further matches between them. They distinguish merging and alignment where merging is described as a process to create a single coherent ontology and alignment as a process that establishes links, which would help align ontologies to reuse information from one another. PROMPT guides the ontologist during the merging or alignment process. PROMPTDIFF is the latest addition to the set of tools. It uses an algorithm, which integrates different heuristic matchers for comparing ontology versions [34]. Three types of mapping levels are defined such as unchanged (nothing has changed), isomorphic (images of each other), and changed (not images of each other).

2.1.9 CHIMAERA, GLUE and CAIMAN

CHIMAERA is an interactive tool just like PROMPT where the ontologist does merging and is guided by the tool. Ontologies to be merged are analyzed by the tool and if linguistic matches are found, the merge is done automatically; otherwise the user is prompted for further action. The only difference between the two tools is the suggestions they make to the ontologist during the merging process. Doan and colleagues developed GLUE [36]. It uses machine-learning techniques to locate mappings of two or more ontologies. Similar concepts are searched using probabilistic measures. It uses multiple learning strategies and exploits information, either in data instances or in the taxonomic structure. It can also make predictions based on a content learner and a name learner. A meta-learner combines the predictions of the two learners. CAIMAN also uses machine-learning techniques and was developed by Lacher and Groh [37]. This tool is quite similar to GLUE.

2.1.10 ONION

Mitra and Wiederhold developed the ONtology composItION (ONION) system [38]. They argue that ontology merging is inefficient, not scalable and expensive. The linguistic matcher looks at all possible pairs of terms from the two ontologies and assigns a similarity score to each pair. For example, given the strings "Department of Defence" and "Defense Ministry", the match function, returns $\text{match}(\text{Defence}, \text{Defense}) = 1.0$ and $\text{match}(\text{Department}, \text{Ministry}) = 0.4$. Then, it matches the two strings, and computes similarity $(\text{"Department of Defence"}, \text{"Defense Ministry"}) = (1 + 0.4)/2 = 0.7$. The

denominator is the number of words in the string with fewer numbers of words, in this case “Defence Ministry”. The similarity score of two strings is then normalized with respect to the highest generated score in the application. If the generated similarity score is above the threshold, then the two concepts are said to match, and they generate an articulation rule: (Match “Department of Defence” “Defense Ministry”), 0.7, the last number gives the confidence measure. The algorithm fails to spot similarities when intended semantics are required. This is not the case for intended syntax.

2.2 Summary of Methodologies

There are twenty one methods that have been published in literature for semantic mediation. Fifteen of them have been highlighted previously, as they were the most relevant. All the methods have their own strengths and tackle the mapping problem in a unique way. It is therefore very difficult to conclude as which technique is the best. As mentioned previously, most techniques are theoretical models that use syntactic, string-based and machine learning matching algorithms.

Only three models actually consider semantics in their algorithm. The usage of libraries and thesaurus is only adopted by OntoMap and IF-Map respectively. Implementation of rules, conflict analysis and conflict resolution is still scarce. Works of Doan [36] and Chalupsky [39] seem to be very labor intensive and assume that the ontology engineer understands the domain, formalisms and mapping rules. There is also a lot of disparity among methods discussed in [11, 13, 14, 22-24, 32, 39, 40].

Among the methods surveyed, the use of heuristics seems to be popular. This is because they are easy to develop. However, they are easily defeasible and can fail like in the ONION example. This is due to the use of syntactic features and structural inputs in designing heuristics. Almost all the techniques did not use intended semantics. Only PROMPT and the Cooperative Framework use taxonomic features and synonym detection of concepts. However, the rigid criterion of having exact attributes in the latter makes the model less flexible.

2.3 Conclusion

Ontology mediation is similar to database schema matching or integration. Although techniques for database schema matching might be useful for mapping ontologies, there are substantial differences between the two. Unlike databases, the creation of ontologies is decentralized. Databases also do not provide formal semantics but ontologies are expected to specify explicitly intended semantics. Therefore an over-arching approach is needed to resolve heterogeneity amongst ontologies. The goal of this research is to introduce an architecture, process methodology and mapping framework for agent-based ontology integration. It proposes a consolidated approach that adopts the best aspects from the techniques cited above.

CHAPTER 3

3. The Interoperability Problem

This section focuses on the discussion of the interoperability problem. The main reasons for data heterogeneity amongst ontologies stems from four main aspects which are: 1) structural heterogeneity (difference in structures of the taxonomy tree); 2) semantic data heterogeneity (scale and representation conflict); 3) subjective mapping (conflicting instance) and 4) atomic stored data (conflicting data type value) [40].

Sources for semantic heterogeneity include differences in data-definition constructs, differences in object representations, and system-level differences in the way that atomic data (e.g., byte order for multibyte data, such as an integer) is stored in the two systems [11, 39, 40]. Semantic heterogeneity is the disagreement of meaning or interpretation of similar or related data. Examples are provided in the following sections to illustrate the heterogeneity problem.

3.1 Structural Heterogeneity Problem

Structural heterogeneity is a problem that arises when there is a mismatch between data structures of ontologies. Differences between lattice structures are common problems

faced by ontologists who attempt to match ontologies from different business entities. In order to illustrate this problem clearly, I use the following example. Imagine that there are two travel-based ontology lattices representing two separate business entities. In this example both entities are involved in the travel domain. Figure 1 illustrates that Hotel B has single and double room occupancy and the tree structure starts with “Rooms”. For the same occupancy category, the tree in Hotel A starts from “Price” instead of “Rooms”. Hotel B has two separate specifications for “Price” in its tree. The difference in the lattice structure is mainly due to design styles that differ among ontology engineers. Based on the survey conducted for ontology structures, structural differences of taxonomies are usually viewed as a common problem.

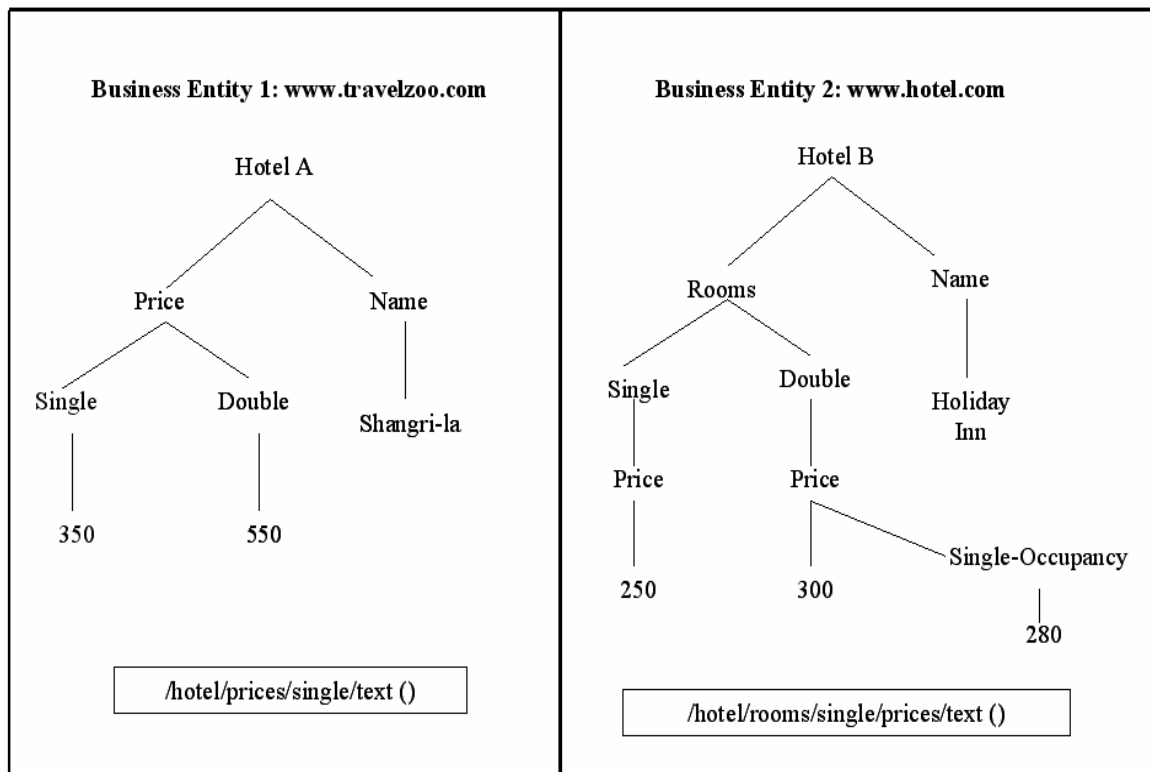


Figure 1 Structural Heterogeneity Problems

3.2 Semantic Data Heterogeneity Problem

Figure 2 illustrates the semantic data heterogeneity problem. This happens when semantically identical information is represented in different data formats or scales. This type of heterogeneity can be further divided into *scale* and *representation* conflict. For example the “Price” attribute is represented in “US Dollar” for Hotel A and in Hotel B the scale used is “Euro Dollar”. This is defined as a scale conflict. Hotel A denotes “Category” and Hotel B denotes “Class” to represent ratings or rankings of the hotel (i.e. 5 indicates a 5 star hotel). Here Hotel B has used a different rating scale (i.e. A which is in their definition is equivalent to 5 star). This is defined as a representation conflict. Suppose, Hotel B uses the label “Quote” instead of “Price” this is again another case of representation conflict.

<u>Business Entity 1 : www.travelzoo.com</u>				<u>Business Entity 2 : www.hotel.com</u>			
Hotel A – Hotel entity				Hotel B –Accommodation entity			
Name	Location	Category	Price	Name	Location	Class	Price
Shangri-la	City Center	5	100	Holiday Inn	City Center	A	100
Note : Scale & representation conflicts <ul style="list-style-type: none"> • 5 star hotel is a category 5 (object representation conflict) • Price in US dollar (scale conflict) 				Note : Scale & representation conflicts <ul style="list-style-type: none"> • Class A is a 5 star hotel (object representation conflict) • Price in Euro dollar (scale conflict) 			

Figure 2 Semantic Data Heterogeneity Problem

3.3 Subjective Mapping Problem

Figure 3 shows the subjective mapping conflict for class and category classification, which are both the third columns of entities Hotel A and Hotel B. The entity names also differ in this example. Hotel A in this example classifies only hotels in its definition. Hotel B classifies all kinds of accommodation (i.e., hotel, bungalow, apartment, villa, etc.). Hotel B has subjective classifications that include all kinds of accommodation but Hotel A refers to only one type of accommodation (i.e. hotel).

<u>Business Entity 1 : www.travelzoo.com</u>				<u>Business Entity 2 : www.hotel.com</u>			
Hotel A –Hotel entity				Hotel B–Accommodation entity			
Name	Location	Class	Price	Name	Location	Category	Price
Shangri-La	Kuala Lumpur	Hotel	600	Shangri-la	Kuala Lumpur	Hotel	450
Marriott	Singapore	Hotel	450	Marriott	Singapore	Hotel	340
Hilton	Budapest	Hotel	350	Hilton	Budapest	Hotel	230
Ritz Carlton	Hamburg	Hotel	600	IBIS	Las Vegas	Apartment	450
Holiday Inn	London	Hotel	350	Holiday Inn	London	Hotel	230
Novotel	Copenhagen	Hotel	200	Radisson	Bangkok	Villa	150
Sheraton	Jakarta	Hotel	450	Sheraton	Jakarta	Hotel	230
Crown Princess	Berlin	Hotel	200	Schulz	Manila	Bungalow	150
Ramada	Venice	Hotel	120	Ramada	Venice	Hotel	92
Note: Subjective mapping conflict for category. Hotel A's category (i.e. Class) refers to only hotel.				Note: Subjective mapping conflict for category. Hotel B's category refers to hotel, apartment, villa and bungalow.			

Figure 3 Subjective Mapping Problems

Clearly, we can say that the subjective mapping conflict between the two hotels creates potential problems when inferences are made about them.

3.4 Atomic, Inconsistency and Redundancy Problem

System-level interoperability of atomic data storage is depicted in Figure 4. As mentioned earlier, atomic data storage is related to byte order-multibyte data [41]. In this example “Price” is stored on location as an integer data type for Hotel A and as a float data type for Hotel B. We need syntactic and semantic data integration to achieve interoperability for both entities.

<u>Business Entity 1 : www.travelzoo.com</u>				<u>Business Entity 2 : www.hotel.com</u>			
Hotel A –Hotel entity				Hotel B–Accommodation entity			
Name	Location	Class	Price	Name	Location	Category	Price
Shangri-La	Kuala Lumpur	Hotel	600	Shangri-La	Kuala Lumpur	Hotel	600
Note: Storage of atomic data is integer for price data in Hotel A.				Note: Storage of atomic data is float for price data in Hotel B.			

Figure 4 System Level Interoperability of Atomic Data Storage Problem

3.5 Summary

Ontologies are becoming increasingly significant because they provide semantics for annotations in the Semantic Web. Ontology development is very distributed in nature and this has led to a large number of ontologies that overlap when addressing similar

domains. Researchers and knowledge engineers in different domain areas have developed ontologies, which have identical problems as illustrated in examples above. It would be difficult to find correspondence among these ontologies as similar concepts could have been expressed using different naming conventions or structures. For example in the domain of computer security, definitions about attacks can exist in two or more ontologies designed by different ontology engineers. Some attack definitions could be more detailed in one ontology but not detailed enough in the other. Ontology mediation via syntactic and semantic means provides a way for these ontologies to be reused. Ontology mediation covers both ontology alignment and ontology merging practices.

CHAPTER 4

4. Ontology Mediation – early efforts and limitations

4.1 Domain Ontology Approach

This approach is analogous to the federated schema approach. It is based on a shared schema and a shared vocabulary (Figure 5). The limitation of this technique is that it is not semantically rich and doesn't support inter-ontology mappings of local schemas [41]. The idea is to have a single ontology that would aggregate data of local schemas.

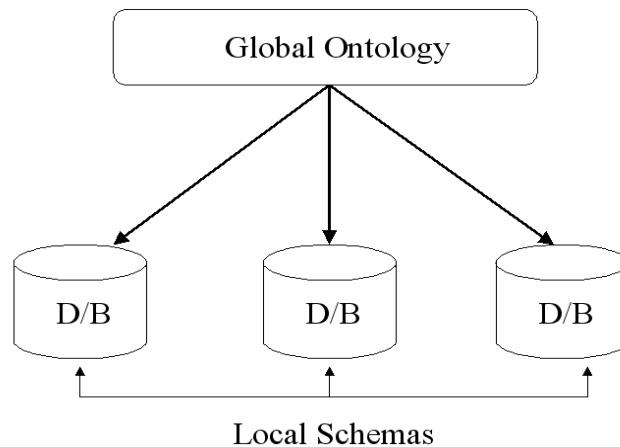


Figure 5 Domain Ontology Approach

4.2 Hybrid Approach

Figure 6 illustrates the hybrid approach for ontology mediation [41]. It's quite similar to the federated approach, as it uses a shared vocabulary. However, autonomy of the local schemas is maintained as the source data has its own ontology. Shared vocabulary is achieved via inter-ontology mapping. Inter-ontology mappings must be defined and this helps capture richer semantics. This method is promising and is quite different from conventional approaches. A simple formalism is provided here to capture only the domain knowledge pertaining to potential semantic conflicts.

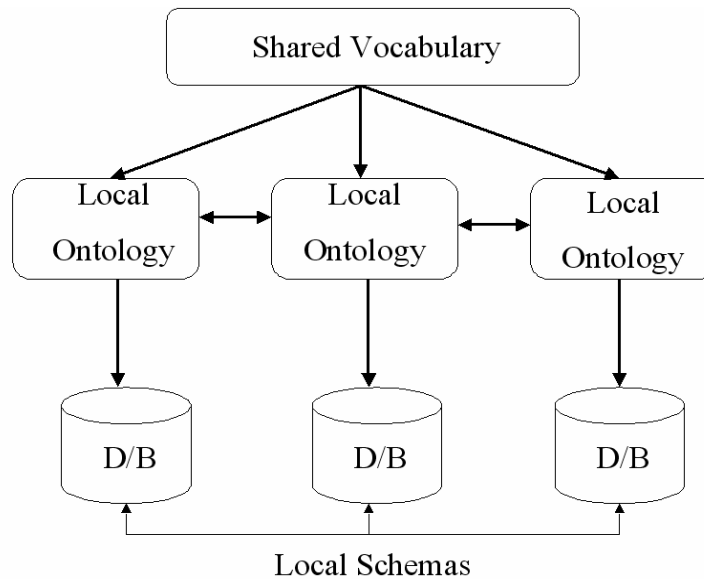


Figure 6 Hybrid Approach

The advantage of this simplified ontology is that it is not domain-specific and does not lose semantic richness. Local ontologies can maintain domain specific definitions and at the same time have the flexibility to share definitions from an upper-ontology.

The Suggested Upper Merged Ontology (SUMO) is based upon this concept, where the upper-ontology holds generic definitions that are at a high level and can cover a broad range of domains. This approach is very relevant for shared hierarchical knowledgebases.

4.3 Multiple Ontology Approach

The multiple ontology approach has been implemented in systems such as OBSERVER [41, 43]. Source data would have their own ontologies and no shared vocabulary would exist among them. Inter-ontology mapping for definitions has to be determined before data can be shared across local ontologies. In other words static mappings are required. Since a global ontology is not maintained here, the semantic richness of this method is inferior to the hybrid approach.

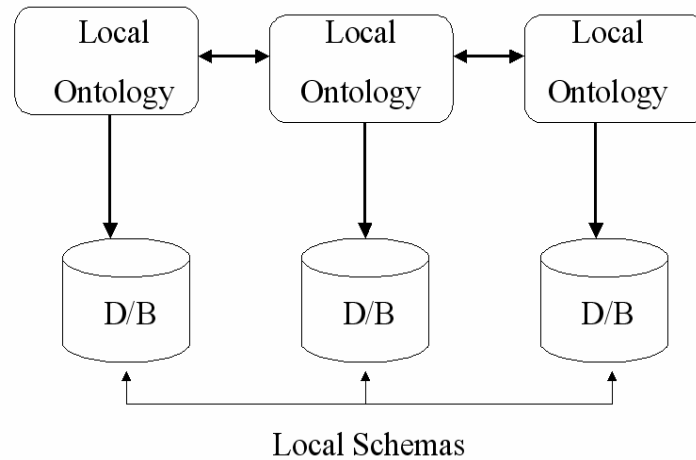


Figure 7 Multiple Ontology Approach

4.4 Need for an Overarching Approach and Thesis Goals

Most of the mediation work carried out today, is to a large extent performed manually. This leads to delay and inefficiency; therefore a semi-automated method with minimal human input is highly desirable. Current methods do not provide formal semantics to mapping data structures. Discussions in the literature review indicate numerous extant ontology mapping methods and techniques. Many of these involve small ontology sets, which could be mapped easily. They also rely heavily on string-based and structure-based similarity measures, which often times fail to produce appropriate mappings. Given that previous approaches will have difficulty scaling to address ontologies for the Semantic Web, a consolidated approach is needed to solve the heterogeneity problems among ontologies.

This thesis attempts to address these problems by introducing an end-to-end framework. It develops a new mapping framework and bridging architecture [15] to help overcome limitations of extant methods, and provides a better platform for dynamic on-the-fly mappings. Similarity score (SRS) is at the core of the proposed architecture. The uniqueness of this similarity score is that it combines both syntactic and semantic measures to match schemas unlike methods discussed in chapter 2 which only used syntactic string matches. A detailed matching algorithm for this is also presented in chapter 5.

The goal of this study is as follows:

- a. Resolve data heterogeneity problems for the Semantic Web via semantic mapping.
- b. Introduce an end-to-end approach that addresses all aspects of the semantic heterogeneity problem.
- c. Provide an over-arching approach that will consolidate best aspects of current approaches that are disjoint to create a unified model.
- d. Propose a mediation framework to map and integrate equivalent data labels via a semantic engine.
- e. Propose a workflow process, mapping framework and methodology for agent based ontology integration.
- f. Propose methods for complex mapping (1:n, m:1, m:n) to overcome limitations of binary mapping (1:1) being carried out at present.

4.5 Summary

More than twenty ontology mediation methods and techniques have been discussed in the previous chapters. Most of them can be categorized into domain, hybrid and multiple ontology approaches. The approach of this thesis closely resembles the hybrid approach. Syntactic as well as semantic concept matching strategies are used to derive a similarity scores (SRS) which are then populated into a matrix. These scores are utilized to

determine ontology mappings. The motivation for this is based on the XMapper system architecture (Figure 8) proposed by Lukasz Kurgan and colleagues [42] from the department of computer science and engineering, University of Colorado at Denver. As shown in figure 8, lattice structures of independent XML sources are analyzed. The data elements are usually structured much like in methods discussed in chapter 2. There are two structures being analyzed here, which are source XML 1 and source XML 2 from the same domain. Constraint analysis is then carried out using a feature vector for each attribute and values are determined for properties of data and the XML structure.

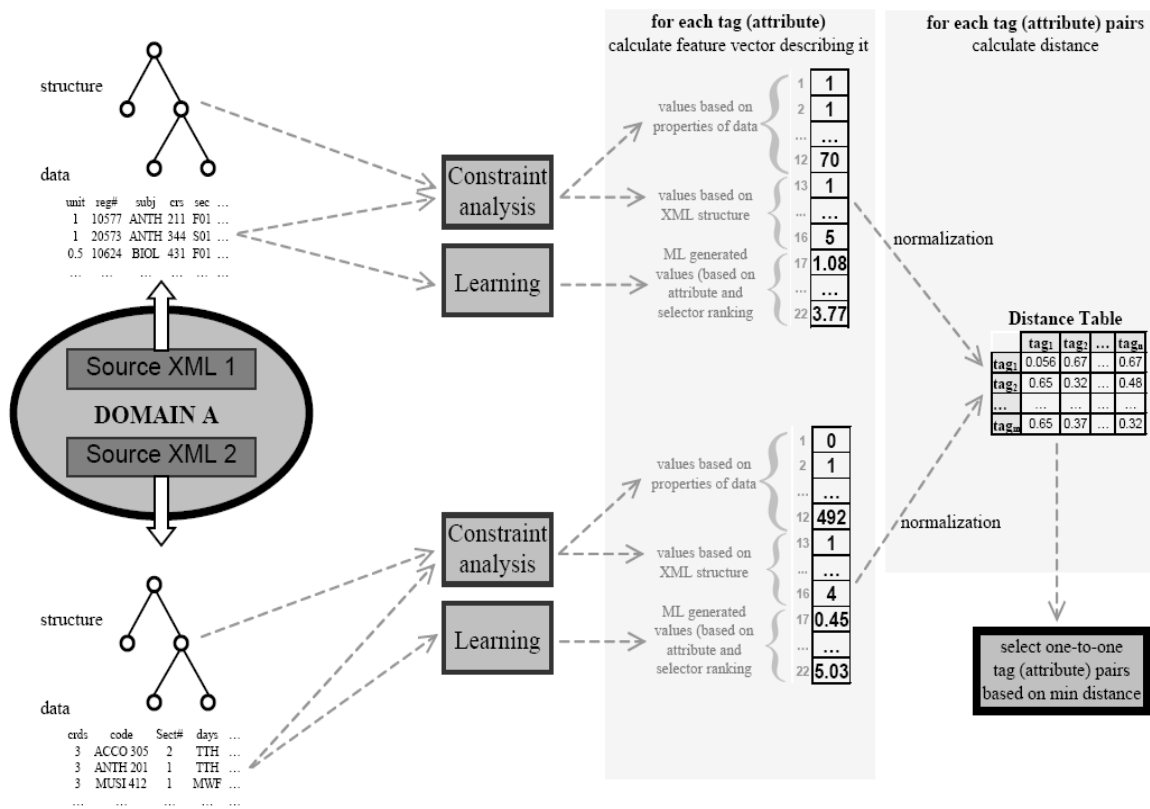


Figure 8 XMapper System Architecture

Learning of ML (Machine Learning) generated values based on attribute and selected ranking is then normalized into a distance table. Every attribute pair tag is measured for minimum distance. The distance table is then populated. This thesis uses this same principle but for ontology sources as apposed to XML sources. It provides a method for inter ontology concept mapping based on a normalized similarity score (SRS). Since similarity is easier to envision compared to distance, similarity is used instead of distance. SRS measures are discussed in greater detail in the next chapter.

CHAPTER 5

5. Ontology Mediation via Similarity Measures

In order to mediate ontological concepts with better accuracy, we first have to define concept similarity. This research adopts a broader definition of similarity which is “semantic equivalence”. The reason for this is that syntactic matching systems today do not provide accurate matches for ontology mediation. Their algorithms are based on string and substring matching which does not include concept relations.

For example if two concepts such as “car” and “automobile” were matched syntactically, the result would be a score close to zero. Although car and automobile are not the same, but they have a logical relationship. As such this research highlights the concept of relatedness being more appropriate for ontology mediation. For example two concepts may be direct opposites (e.g. hot-cold) but are still related through lexical relationships. This is an example of *antonyms*.

According to the Oxford dictionary, Martineau, a philosophy researcher was the first to define semantics as: “the study of meaning and changes of meanings” in 1887. Since the

Semantic Web is based on a shared conceptualization of an application domain, the semantics of terms and concepts should be readily negotiated. In particular the notion of “semantic equivalence” is key for measuring similarity. As such a more reliable method that uses concept relations is needed.

This research uses a six-part similarity test to check for semantic equivalence. The test involves concept relations and is defined as a function of equivalence (E), inclusiveness (IC), consistency (CN), semantic similarity (SEM) and syntactic similarity (SYN). Concepts that are disjoint (D) are negated. SEM and SYN scores are aggregated to produce a unique similarity score called Semantic Relatedness Score (SRS).

A unique feature of SRS is that it provides greater precision and reliability due to its hybrid nature compared to pure syntactic scores. SRS scores are used to populate and create a similarity matrix. The matrix presents the scores to an ontologist who would use them to match concepts. The similarity matrix idea was derived from the distance table project discussed in figure 8 previously. However only high similarity scores (SRS) are used instead of minimum (*min*) distance scores.

5.1 Introduction

Similarity measures play an important role in many applications today, such as information retrieval (IR), word sense disambiguation (WSD), word completion, spelling correction and text summary. Similarity measures are highly mathematical i.e., they

usually apply *vector analysis* for distance and *probability* analysis for similarity. Jaccard's coefficient is a probability measure, which can be used for measuring similarity and its inverse function can be used to measure distance.

$$\text{Jaccard Similarity Coefficient} - J(A, B) = P(A \cap B) / P(A \cup B) \quad (1)$$

$$\text{Jaccard Distance} = [P(A \cup B) - P(A \cap B)] / P(A \cup B) \quad (2)$$

Similarity is measured by the size of *intersection* of sets A and B divided by the size of *union* of sets A and B. Distance or dissimilarity is measured by simply subtracting the similarity value from 1 or by dividing the difference of the sizes of the union and intersection of two sets by the size of the union of the two sets.

The linguistic similarity measures between concepts are opposite to distance measures. Similarity is defined via lexical relations of *synonyms* (e.g. automobile–car) and *hypernyms* (e.g. vehicle–car). Relatedness covers a broader scope of lexical or functional relations between words like *antonyms* (e.g. day-night) [43].

There are a number of competing approaches for these measures and the focus of this study is on Semantic Relatedness Scores (SRS). In the approach developed for this thesis, similarity scores of word pairs are determined by combining syntactic (SYN) as well as semantic (SEM) measures. SRS scores were tested against Human Cognitive Responses (HCR) to validate this approach. This is discussed in greater detail in chapter 6.

5.2 Theoretical Foundation and Assumptions

Ontology mediation is a process to identify similarities of domain ontology concepts and establish mappings between them. Various research groups have introduced several methods for such mappings (see chapter 2). This study introduces a hybrid measure i.e. Semantic Relatedness Scores (SRS). Figure 9, shows a typical diagram of how schemas are mapped. Dotted lines show the semantic correspondence between the ontological concepts that needs to be mediated.

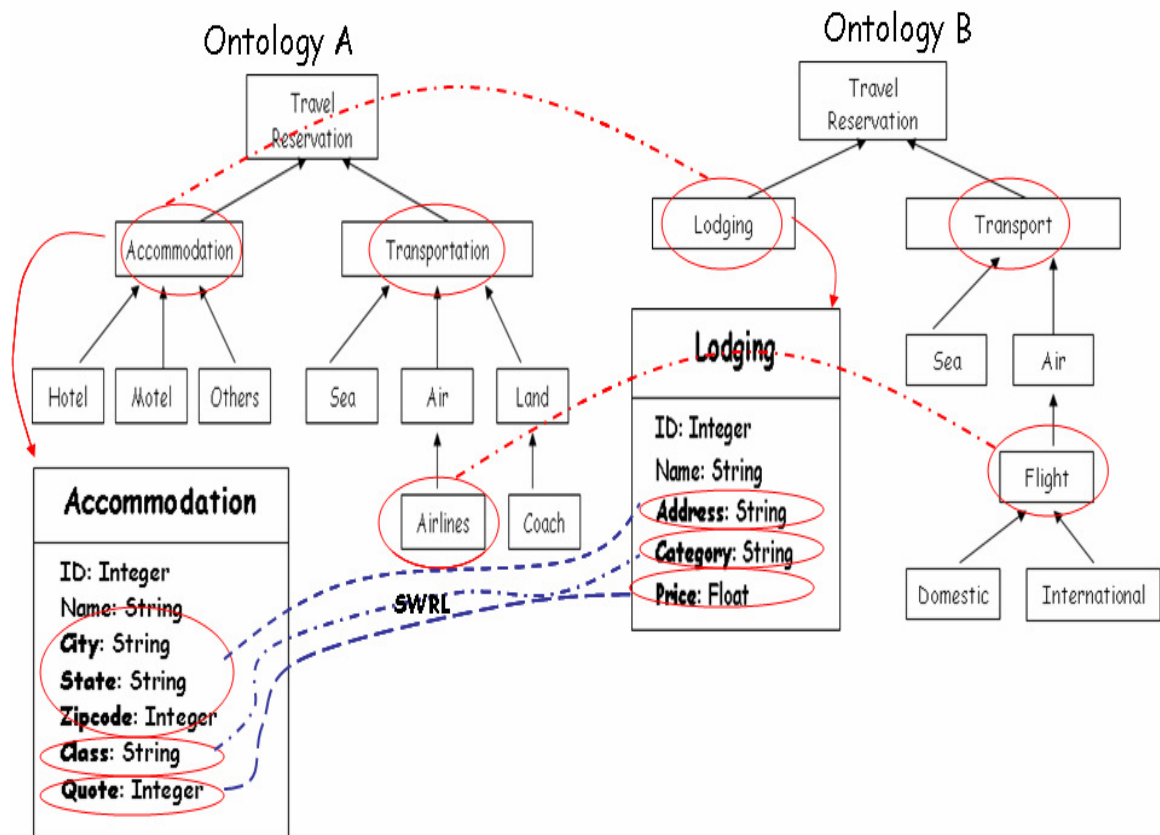


Figure 9 Achieving Semantic Correspondence via Mappings

There are two travel reservation ontologies that are being mapped in the diagram above, i.e. ontology A and ontology B. As mentioned previously, the process begins when we start to identify similarities between ontological concepts and establish mappings between them i.e. semantic correspondence. It is important to note that Semantic Relatedness Scores (SRS) and the test for relations are important to determine semantic equivalence of concepts. As mentioned earlier, the six-part test determines equivalence (E), inclusiveness (IC), consistency (CN), semantic similarity (SEM), syntactic similarity (SYN) and negates disjointed (D) concepts.

SEM and SYN scores are subsets of SRS. SRS is aggregated from a combination of linguistic and non linguistic measures. A total of thirteen measures were explored to build SRS. The thirteen measures are discussed in detail in section 5.3. Ontology mediation is a process involving several steps, each of which is discussed in chapter 6. This section focuses on the six-part similarity test. The motivation of this test is to reduce extraneous data and to provide an ontologist with meaningful data to perform matching.

Currently an ontologist has to sift through thousands of concepts that are only syntactically equivalent. This test will not only provide syntactically matched concepts but also semantically matched concepts. Since, irrelevant data is filtered out this can minimize the workload of the ontologist. The semantic mediation approach in this thesis is predicated on having well-formed ontological fragments before mapping can be done. By this we mean that concepts, relationship and constraints should satisfy a six-part test.

The definitions used for the first four out of the six-part test i.e. equivalence (E), inclusiveness (IC), disjoint (D) and consistency (CN) were adopted from the works of [22]. Additional definitions have been added to their work, i.e. semantic similarity (SEM) and syntactic similarity (SYN) to complete the six-part test. SRS is made up of the combination of SEM and SYN. The following are symbols and definitions used to describe the six-part test based on works of [22]:

- O denotes an ontology
- O_i and O_j denote source ontology (O_i) and target ontology (O_j)
- C denotes classes
- C_i and C_j denote source classes (C_i) and target classes (C_j)
- Relations (R) denotes relations between classes
- Shared conceptualization is comprised of C and R , i.e. $\Sigma \{C, R\}$ pairs, Σ denotes total number of concepts (C) and relations (R) that exist in a domain
- c denotes subclass of superclass (C)
- a denotes attributes of a class
- Superclass relation – C_i and C_j are superclasses of c_i and c_j
- Subclass relation – $c_i \in C_i (O_i)$ and c_j where $c_j \in C_j (O_j)$

Equivalence Test (E)

Let C_i and C_j be classes, where $C_i, C_j \in C$. C_i is said to be **equivalent** to C_j , if $C_i = C_j$.

Also expressed as $\exists C_i, C_j$, s.t. $C_i = C_j$, if :

- Both are **semantically equivalent**, e.g., ($C_i = \text{hotel}$ & $C_j = \text{hotel}$)
- Both are **synonyms** e.g. ($C_i = \text{hotel}$ & $C_j = \text{motel}$)
- Both define the **same attributes** (a) of their respective classes e.g. $C_i = \text{hotel}$ has (a) $a_i = \{\text{name, location, rank, price}\}$ and $C_i = \text{motel}$ has (a), $a_j = \{\text{name, location, rank, price}\}$.

Inclusiveness Test (IC)

Let C_i and C_j be classes, where $C_i, C_j \in C$. C_j is said to be **inclusive** of C_i , if $C_i \subseteq C_j$. For example if $C_i = \text{selling price}$ and $C_j = \text{price}$, then price is denoted as the **superset** of selling price. Thus ($C_j \supseteq C_i$) or ($C_i \subseteq C_j$) where, selling price is a subset of price. C_i can also be expressed as a type of C_j . This is applicable to **hyponyms**. This can also be expressed as: $\exists C_i, C_j$, s.t. $C_i \subseteq C_j$ or $C_i \supseteq C_j$.

Disjoint Test (D)

Let C_i and C_j be classes, where $C_i, C_j \in C$. C_i, C_j are said to be **disjoint**, if $C_i(a_i)$ and $C_j(a_j)$ are **disjoint**. That is they share no common attributes s.t. $C_i(a_i) \cap C_j(a_j) = \{\}$ or \emptyset . For example classes e.g. $C_i = \text{hotel}$, has $(a_i) = \{\text{name, location, rank, price}\}$ and $C_j = \text{hotel}$, has $(a_j) = \{\text{rating, zipcode, services, booking}\}$. Attributes (a) of both classes (C) do not have any overlapping properties. Also expressed as: $\exists a_i, a_j$, s.t. $a_i \cap a_j = \emptyset$.

Consistency Test (CN)

Let O denote an ontology, let $C_i(O_i)$ denote classes, C_i defined in ontology, O_i and $C_j(O_j)$ denotes classes, C_j defined in ontology, O_j . $a_i(C_i(O_i))$ denotes attributes a_i of class C_i , where C_i is a class in ontology O_i . C_i and C_j are **consistent** if, a_i^a, a_i^b, a_i^c ($a_i^a, a_i^b, a_i^c \subseteq (C_i(O_i))$) and that a_i^a, a_i^b, a_i^c have nothing in common s.t. $a_i^a \neq a_i^b \neq a_i^c$. In other words all the attributes (a_i) within the same ontology (O_i) have nothing in common among them e.g. $a_i(C_i(O_i))$. Given $(a_i) = \{\text{name, location, rank, price}\}$ all attributes should have nothing in common e.g. $\{\text{name} \neq \text{location} \neq \text{rank} \neq \text{price}\}$. Attributes of a given class i.e. $\{\text{name, location, rank, price}\}$ are subsets of that class. $\forall a_i^e, a_i^f, a_i^g$ ($a_i^e, a_i^f, a_i^g \in C_i(O_i)$) and $a_i^e \neq a_i^f \neq a_i^g$ ($e, f, g \in \mathbb{N}$). Source and target ontology classes must be consistent before they can be matched.

Syntactic Similarity Test (SYN)

Let C_i and C_j be classes, where $C_i, C_j \in C$. Let SYN denote the syntactic score. C_i is said to be **syntactically similar** to C_j , if $C_i(\text{SYN})=C_j(\text{SYN})$. Also expressed as $\exists C_i, C_j$, s.t. $C_i(\text{SYN}) = C_j(\text{SYN})$. SYN is measured based on prefix, suffix, string and substring matching. SYN scores are any value between 0 and 1. The threshold is set to 0.5. Scores below 0.5 are saved into a log.

Semantic Similarity Test (SEM)

Let C_i and C_j be classes, where $C_i, C_j \in C$. Let SEM denote the semantic score. C_i is said to be **semantically similar** to C_j , if $C_i(\text{SEM})=C_j(\text{SEM})$. Also expressed as $\exists C_i, C_j$, s.t. $C_i(\text{SEM}) = C_j(\text{SEM})$. SEM is measured based on a number of methods that are incorporated based on Natural Language Processing (NLP) and cognitive reasoning. SEM scores are any value between 0 and 1. The threshold is set to 0.5. Scores below 0.5 are saved into a log.

Mapping /Alignment

Ontology **alignment /mapping** (M) of two ontologies O_i and O_j , is carried out when all the tests above are satisfied. All six levels are tested for before data is prepared to be bridged.

Integration

Ontology **integration** (I) happens when source ontologies (SO) are merged to produce target ontologies (TO).

5.2.1 Syntactic Relatedness (SYN)

Syntactic mapping is an approach to map concepts using a linguistic matcher. The linguistic matcher is predominantly based on string match, prefix match, suffix match and substring match. Syntactic relatedness scores for pairs of concept terms are computed and assigned to each pair analyzed. Those with high scores are candidates for matching. The

structure and construct of words are used instead of meanings in this method. Syntactic relatedness provides a gross measure for similarity and requires semantics to make the output feasible.

$$\text{SimName (Cc,Co)} = 1 - (\text{Lev(CcName,CoName)}) \quad (3)$$

Syntactic matching evaluation usually applies a distance function over a pair of strings, to determine the dissimilarity between them. Levenshtein's distance (LD) measure is a good example [44]. It provides total number of character changes needed to transform one string into another. The smaller the dissimilarity, the more similar are the pair of strings, and therefore requires fewer character changes (Appendix V). SimName denotes similarity of concept names for Cc and Co. Lev denotes the (LD) measure for finding distance (d) between CcName and CoName. To find similarity, (d) is deducted from 1.

5.2.2 Semantic Relatedness (SEM)

Semantic mapping is an approach to map concepts of disparate ontologies on the basis of semantic similarity (i.e., meaning) and the degree of relatedness that exists between those concepts. Before concepts are mediated a check for their similarity must first be determined. With the aid of cognitive agents and the English language lexical database (i.e., WordNet), concept similarity can be measured. Semantic mediation helps to create a *semantic bridge* between ontologies with the aid of a lexical database. This provides rich definitions to concepts, which helps us to map them. For example if we wanted to match *price* and *quote*, for travel ontologies, a brief lookup on the database would reveal if they

were semantically related. Semantic relatedness considers all shades of meanings for a concept such as synonymy, meronymy, antonymy, functions, associations and polysemy, which is common in Natural Language Processing (NLP).

When the same word has more than one meaning (different *senses*), it causes polysemy and when it has opposite meanings it causes antonymy. Synonymy corresponds to the situation when two different words have the same meaning. Since concepts are represented in natural language by words, their shades of meanings must be analyzed for relatedness before mediation is carried out. Cognitive agents such as WordNet are used to achieve this. Unrelated concepts that do not pass the similarity test are discarded totally (e.g. *tree* and *sky*). This thesis combines semantic matching with syntactic matching to provide better results for ontology mediation.

Based on the six-part test above the following equation is generated:

$$S(f_x) \text{ where, } x = \{E, IC, CN, D, SYN, SEM\} \quad (4)$$

5.3 Matching Algorithm

A matching algorithm has been developed for the six-part test discussed previously. In this section a detailed explanation is provided for this matching algorithm. The matching algorithm runs matches and presents the results to the domain expert for final consideration. The domain expert's input is only required towards the final stage. The

idea here is to reduce the workload of domain expert by eliminating extraneous data especially in complex environments where large set of match candidates are found.

The first three tests (i.e., E, IC and CN), are iterative. This specifically addresses all nodes in a hierarchical ontology structure. Then SYN and SEM tests are applied allowing full semantic matching to be computed among classes and instances. Parameters are entered in each execution of the algorithm and the acceptance threshold is set for SRS scores. Only classes that have SRS scores higher than the specified threshold are presented to the domain expert for scrutiny. The semantic matching algorithm uses highly reliable measures such as Lin, Gloss Vector, WordNet Vector and LSA (Latent Semantic Analysis) to determine SRS which is the superset of SYN and SEM. The matching algorithm shown in figure 10, shows 13 detailed steps before the semantic matching engine produces mappings (M).

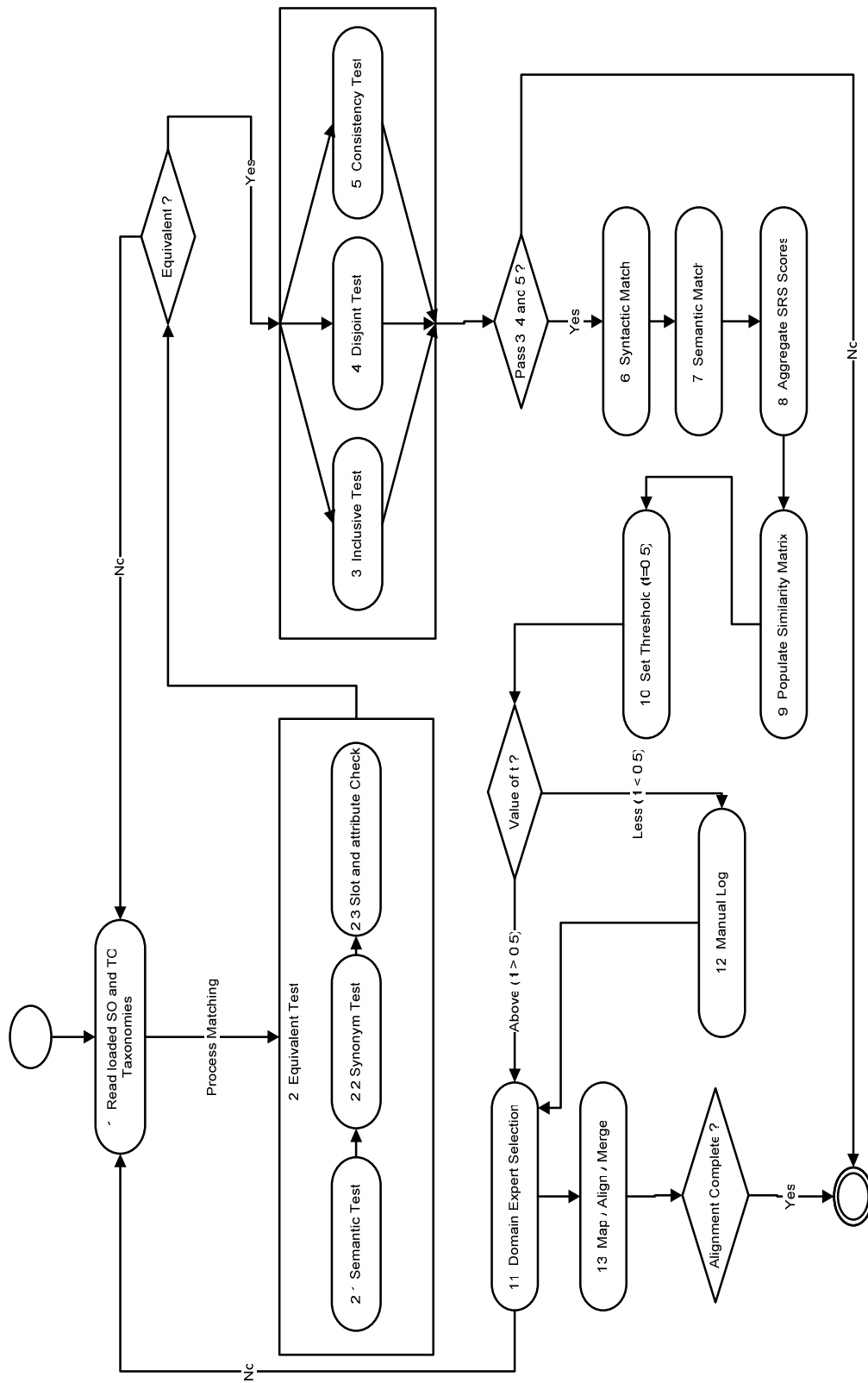


Figure 10 Matching Algorithm

The process begins when two ontologies are first loaded (i.e. O_1 and O_2) and they are identified as source ontology (SO) and target ontology (TO). The taxonomies are read and translated for beginning matching. An **equivalent test** (E) is carried out for data labels to test their similarity in terms of three parameters, 1) test for semantically equivalent data labels, 2) test for synonyms and 3) test for similar slots or attribute names. C is used to refer to classes and c refers to attributes or slots. All the nodes in a taxonomy graph can be tested for tests in step 1, step 2 and step 3.

Details of the matching algorithm steps are as follows:

- Step 1 – Read loaded SO and TO taxonomies: Semantic engine reads taxonomies of the SO and TO. Prepare for detailed matching tests of data labels, go to step 2.
- Step 2 – Equivalence Test: Test for the **equivalence** of source and target classes: Test 1) do they have semantically equivalent data labels, Test 2) are they synonyms or Test 3) do they have the same slots or attribute names. If equivalent, proceed to step 3, 4 and 5. Else go to step 1.
- Step 3 – Inclusive Test: Source and target classes or concepts (C) are **inclusive** if, the attribute (c) of one is inclusive in the other. In other words *selling price* (c_i) is inclusive in *price* (c_j), this is applicable to *hyponyms*. If inclusive, proceed to step 6.
- Step 4 – Disjoint Test: Source and target classes or concepts (C) are **disjoint** if, the intersection of their two attribute sets (c), c_i and c_j results in an empty set $\{\}$ or \emptyset . If match test is not disjoint, proceed to step 6.

- Step 5 – Consistency Test: Source and target classes or concepts (C) are **consistent** if, all the attributes or slots (i.e. c_1 and c_2) in the class, have nothing in common s.t. $c_1 \cap c_2 = \{\}$. All slots must belong to class that is being tested. This can be configured with RacerPro. If consistent, proceed to step 6.
- Step 6 – Syntactic Match: Syntactic match similarity scores based on concept prefix, suffix, substring matches are calculated. This calculation is performed for every concept in the source and target ontology. Go to step 7.
- Step 7 – Semantic Match: Semantic match similarity scores based on cognitive measures such as LSA, LC, RS, JC, LN, WordNet and HS are among others used. This calculation is done for every concept in the source and target ontology. Go to step 8.
- Step 8 – Aggregate both similarity scores: Similarity inputs from step 6 and 7 are aggregated, to produce SRS. Go to step 9.
- Step 9 –Populate similarity matrix: The aggregated values (SRS) from step 8 of candidate labels are populated into the similarity matrix. Multiple matches are carried out. Values are to be verified against the threshold. Go to step 10.
- Step 10 – Set threshold: Threshold value (t) is set based on scale used. For a scale between, 0 and 1 the threshold value is usually 0.5 ($t > 0.5$). Those below threshold are logged in file in step 12. If greater than the threshold value, go to step 11.
- Step 11 – Domain Expert Selection: At this stage, candidates from step 10 are presented to domain expert by the system. Input from step 12 is accepted at the discretion of the domain expert.
- Step 12 – Manual Log: Selection is made manually only for those values below threshold. The domain expert uses his own cognitive judgment. Go to step 13.

- Step 13 – Mapping/Alignment/ Merge: All the candidates for mapping, alignment or merge (i.e. integration) chosen from step 11 and 12 are processed. End.

5.4 Benefits of Combining Syntactic and Semantic Measures

In order to achieve the Semantic Web dream, both syntactic and semantic interoperability aspects must be given importance. The distinction between syntax and semantics is that *syntax* refers to form and structure but *semantics*, as mentioned earlier, is the representation of meaning. A given word could have multiple meanings, which are referred to as word senses. Grammatical rules provide *syntax*, while *semantics* provides meaning. Semantic matching is an approach to map concepts based on the level of similarity of meanings between those concepts. Syntactic matching does not consider semantic similarity between concepts and is largely based on rules.

Syntactic integration defines rules in terms of class and attributes names and does not take into account the structure of the ontology. Such integration is usually conceptually blind but easier to implement. Syntactic matches provide a gross match for concepts. This saves the ontology engineer a lot of time. The main benefit of combining both syntactic and semantic measures is getting better results.

5.5 Measuring Similarity and Similarity Measures

We distinguish similarity of entity classes and similarity of entity instances. Entity classes refer to concepts in the real world but entity instances refer to physical objects in

the real world. Since this study focuses on entity classes, it does not address the similarity assessment among attribute values of specific instances of a class. For example, when assessing the similarity between a *hotel* and an *accommodation* entity, this study considers attribute concepts of those entity classes (e.g., name, location, class and price) but disregards the similarity assessment among values associated with instances (e.g. Shangri-La, Paris, Hotel, \$600). Various approaches for computing semantic relatedness of words or concepts have been proposed, e.g., dictionary-based [45], ontology-based [45, 46], information-based [47-49] or distributional [46]. The knowledge sources used for computing relatedness can be as different as dictionaries, ontologies or large corpora. Budanitsky and Hirst [47], explain three prevalent approaches for evaluating SR measures: *mathematical analysis*, *application specific evaluation* and *comparison with human judgments*. Mathematical analysis can assess a measure with respect to some formal properties; however, mathematical analysis cannot tell us whether a measure closely resembles human judgments. The following sections describe the existing approaches and measures in detail.

5.5.1 Leacock-Chodorow Measure (LC)

Leacock-Chodorow [48] uses the length of the shortest path $len(c1, c2)$ of two synonym sets (*synsets*) to measure similarity. The method counts up the number of links between the two *synsets*. The shorter the length of the path, the more related they are. The measure was found to have performed well for a medical taxonomy called MeSH (<http://www.nlm.nih.gov/mesh/>). However it is limited to *is-a* links and scales the path

length by the maximum depth D of the taxonomy of noun hierarchies in WordNet. The following formula is used to compute semantic relatedness:

$$\text{sim}_{\text{LC}}(c1, c2) = \frac{-\log \text{len}(c1, c2)}{2D} \quad (5)$$

Where $c1, c2$ are synsets, sim denotes similarity, LC denotes Leacock-Chodorow, D denotes maximum depth of noun hierarchies in WordNet and len denotes length of path.

5.5.2 Resnik Measure (RS)

Resnik [49] was the first similarity approach to include ontology and corpus. Similarity between two concepts is defined as information content of their lowest super-ordinate (most specific common subsumer) i.e., $\text{lso}(c1, c2)$. The variable p denotes the probability of encountering an instance of a synset c in the corpus [50, 51]. The formula below is used to compute similarity:

$$\text{sim}_{\text{R}}(c1, c2) = -\log p(\text{lso}(c1, c2)) \quad (6)$$

Where $c1, c2$ are synsets, R denotes Resnik, lso denotes lowest super-ordinate, and p denotes the probability of encountering an instance.

5.5.3 Jiang-Conrath Measure (JC)

Jiang-Conrath [52] uses information content in the form of the probability of encountering an instance of a child-synset given the instance of a parent synset.

Information content of two nodes as well as their most specific subsumer is important. Semantic distance is measured here instead of semantic similarity. The formula below is used to compute distance:

$$\text{dist}_{\text{JC}}(c1, c2) = 2\log(p(\text{Iso}(c1, c2))) - (\log(p(c1)) + \log(p(c2))) \quad (7)$$

Where $c1, c2$ are synsets, dist denotes distance, JC denotes Jiang-Conrath, Iso denotes lowest super-ordinate and p denotes the probability of encountering an instance.

5.5.4 Lin Measure (LN)

Lin [53] uses the theory of similarity between arbitrary objects. This measure uses the same elements such as dist_{JC} with slight changes. The formula is as follows:

$$\text{sim}_{\text{L}}(c1, c2) = \frac{2 \times \log p(\text{Iso len}(c1, c2))}{\log p(c1) + \log p(c2)} \quad (8)$$

Where $c1, c2$ are synsets, sim denotes similarity, L denotes Lin, Iso denotes lowest super-ordinate, len denotes length of path and p denotes the probability of encountering an instance.

5.5.5 Hirst-St.Onge Measure (HS)

Hirst-St.Onge [54] assumed that two lexicalized concepts are semantically close if, a path that is not too long and does not change often, connects their WordNet synsets. They measure semantic similarity with the following formula:

$$\text{rel}_{\text{HS}}(c1, c2) = C - \text{path length} - k \times d \quad (9)$$

Where rel denotes relation, HS denote Hirst-St. Onge, d is the number of changes of direction in the path of synsets, C and k are constants. If a path does not exist then $rel_{HS}(c1, c2) = 0$ and synsets are deemed unrelated.

5.5.6 PMI Measure (PM)

Turney [55] computes similarity of word pairs based on this algorithm, also referred to as PMI-IR (Pointwise Mutual Information) and IR (Information Retrieval). It is a successful measure for approximating human semantics. A test match for 80 synonyms on TOEFL (Test of English as a Foreign Language) and 50 synonyms on ESL (English as a Second Language) produced higher scores compared to LSA (Latent Semantic Analysis) and LSI (Latent Semantic Indexing). The following formula is used to measure similarity:

$$PMI(c1, c2) = \log_2 \frac{P(c1, c2)}{P(c1) \times P(c2)} \quad (10)$$

It is based on the probability (P) of finding two concepts of interest ($c1$ and $c2$) within the same text window versus the probabilities (P) of finding the concepts separately. $P(c1, c2)$ is the probability of finding both $c1$ and $c2$ in the same window. $P(c1)$ and $P(c2)$ are probabilities of finding concepts $c1$ and $c2$ separately.

5.5.7 NSS Measure (NS)

Cilibrasi, R. and Vitanyi [56] introduce Normalized Search Similarity (NSS) which is adapted from Normalized Google Distance (NGD). Cilibrasi, R. and Vitanyi [57]

measure similarity between two concepts using probability of co-occurrences as demonstrated by the following equation:

$$NGD(c1, c2) = \frac{\max \{\log f(c1), \log f(c2)\} - \log f(c1, c2)}{\log M - \min \{\log f(c1), \log f(c2)\}} \quad (11)$$

M is the number of searchable Google pages, and $f(x)$ is the number of pages that Google search returns for searching x . NGD is based on the Google search engine. The equation may also be used with other text corpora such as Google, Wikipedia, New York Times, Project Gutenberg, Google groups and Enron E-mail corpus.

5.5.8 GLSA, LSA and SA Measure (SA)

Landauer and Dumais [58] introduce LSA (Latent Semantic Analysis). It uses Singular Value Decomposition (SVD) to analyze relationships among concepts in a collection of text. It is a fully automatic computational technique for representing the meaning of text. A passage is viewed as a linear equation and it's meaning is a sum of words i.e., $m(\text{passage}) = m(\text{word}_1) + m(\text{word}_2) + m(\text{word}_n)$. Eigenvalue is used for ordering the vector and cosine values are used to represent similarity:

$$\cos \theta_{xy} = x \cdot y / |x| |y| \quad (12)$$

LSA provides better results than keyword matching; for example, *doctor-doctor* match gives a 1.0 score for both LSA and keyword match. However, *doctor-physician* results in a 0.8 score for LSA and 0 score for keyword match. This is why LSA is better. GLSA is

Generalized LSA computes term vectors for vocabulary V of document collection C using corpus W [59]. Anderson and Pirolli [60] introduced Spreading Activation (SA), which uses a semantic network to model human memory using a Bayesian analysis. The following is their formula to measure similarity:

$$SA(w_1, w_2) = \log \frac{P(X=1|Y=1)}{P(X=1|Y=0)} \quad (13)$$

5.5.9 WordNet ::Similarity Measure (WN)

The similarity measure program is an open-source Perl module developed at the University of Minnesota. It allows the user to measure the semantic similarity between a pair of concepts. The system provides six measures of similarity and three measures of relatedness based on the WordNet lexical database. The measures of similarity are based on WordNet *is-a* hierarchy. Specifically, measures used are Resnik (RS), Lin (LN), Jiang-Conrath (JC), Leacock-Chodorow (LC), Hirst-St.Onge (HS), Wu-Palmer (WP), Banerjee-Pedersen (BP), and Patwardhan-Pedersen (PP).

5.5.10 Gloss Vector (GV)

The Gloss Vector (GV) measure forms a second-order co-occurrence vectors from the glosses of concept definitions. It primarily uses WordNet definitions to measure similarity or relatedness of two or more concepts. GV determines similarity of two concepts by determining the cosine of the angle between their gloss vectors.

It augments glosses of concepts with glosses of adjacent concepts as defined by WordNet relations to resolve data sparsity problems due to extremely short glosses.

5.6 Similarity Relatedness Scores (SRS) and Similarity Function

Similarity (s) is the strength of relatedness existing between concepts. A similarity score can be normalized between 0 and 1. Distance (d) however is the inverse of similarity ($d=1-s$). It is also called dissimilarity, where discrepancies between concepts are given importance. If distance (d) is measured first, as in the case in most situations, one could measure similarity (s) with this expression ($s=1-d$). Intuitively speaking, two concepts like *price* and *price* can be said to have a similarity score of 1 ($s=1$), while concepts like *tree* and *sky* would have a score of 0 ($s=0$), because there is no significance relationship between the two. In summary if $s=0$, then $d=1$ and if $d=0$, $s=1$.

Similarity ($s=1$) means exact similarity like *price* and *price*. However, *price* and *quote* are not precisely similar, and as such, the value for similarity would have to be greater than 0 but less than 1 (e.g. $0 < s < 1$). Exact dissimilarity would be concepts like *tree* and *sky* as mentioned earlier with $s=0$. If similarity has not been normalized (i.e. s ranges from -1 to $+1$) then $s = -1$ would mean $d=1$ and $s=1$ when $d=0$. In many cases measuring distance or dissimilarity (d) is easier than measuring similarity (s). This is why techniques in the past have always used distance tables, graph theory and nearest neighbor for their distance measures. (s) and (d) can be determined between two concepts based on feature variables and its scales.

As discussed previously, measures of similarity have been researched widely in the areas of cognitive sciences, databases, natural language processing (NLP) and artificial intelligence (AI). A popular usage of these measures is in word sense disambiguation, information retrieval and malapropism detection. SRS introduced in this thesis, is a similarity score that is a function of the above measures. Semantic similarity can thus be represented by the following function:

$$\text{SRS} = f_x \{ \text{LC, RS, JC, LN, HS, PM, NS, LSA, WN, GV, SYN} \} \quad (14)$$

Although thirteen measures were explored to build the SRS function above, empirical tests show that only four out of the thirteen measures provided the highest degree of relevance, precision and reliability. Empirical data based on a study done in Princeton by Miller and Charles [1] were used to test the revised SRS function. Results also showed a higher degree of correlation i.e. 92% when compared with human cognitive evaluation of sixty concept words. This makes the revised SRS function more reliable. The four measures used in combination with SYN were Lin (LN), Gloss Vector (GV), WordNet (WN) and LSA. Lin (LN) and Gloss Vector (GV) measures were obtained through an API service supported by Ted Pedersen and Jason Michelizzi from University of Minnesota¹⁰. WordNet and LSA measures were obtained through an API service provided by Rensselaer's MSR server¹¹. Scores were aggregated to derive SRS. Thus the revised SRS function is represented as:

¹⁰ Ted Pedersen and Jason Michelizzi - <http://marimba.d.umn.edu/cgi-bin/similarity.cgi>

¹¹ Rensselaer MSR Server - <http://cwl-projects.cogsci.rpi.edu/msr/>

$$\text{SRS} = f_x \{ \text{LN}, \text{LSA}, \text{WN}, \text{GV}, \text{SYN} \} \quad (15)$$

SRS is unique because it adopts both cognitive and syntactic measures to calculate similarity between concepts, making it more reliable. Current research on ontology mediation, focuses mainly on binary mappings (1:1) and does not use cognitive measures to determine semantic similarity for concept matching. In most research only syntactic matching is carried out. One of the contributions of this thesis, is to including multiple mappings (1:n, n:1 and m:n), and provide SRS scores which is a composite measure for similarity. The scores are used to populate a similarity matrix, which is a major piece of the proposed mediation framework and architecture.

5.6.1 Similarity Matrix and Binary Mappings

The SRS function includes all cognitive, semantic and syntactic measures discussed above into a similarity matrix. The matrix of figure 11 shows similarity scores produced via the SRS function for concept names such as *price*, *quote*, *cost*, *amount*, *damage* and *value*. A pair-wise measure for *price-price*, *price-quote*, *price-cost*, *price-amount*, *price-damage* and *price-value* is given. The first row is matched with the subsequent columns. The results are as follows:

Document	price	quote	cost	amount	damage	value
price	1	0.20	0.46	0.19	0.03	0.26
quote	0.20	1	0.16	0.03	0.01	0.15
cost	0.46	0.16	1	0.26	0.16	0.37
amount	0.19	0.03	0.26	1	0.16	0.26
damage	0.03	0.01	0.16	0.16	1	0.11
value	0.26	0.15	0.37	0.26	0.11	1

Figure 11 Similarity Matrix

Binary Mappings (1:1)

Literature shows that binary mappings are popular among ontology engineers. However most methods described are based on syntactic matches only. In this section, binary mappings are shown for the mediation of two ontologies O_1 (Ont 1) and O_2 (Ont 2). The ontologies belong to separate domains i.e., A and B. The SRS function is used to compute similarity scores by matching terms $(t_1)O_1$ with $(t_1)O_2$ and $(t_2)O_1$ with $(t_2)O_2$ and populates the similarity matrix. Figure 12 shows $(t_1)O_1$ –Quote is being matched with $(t_1)O_2$ –Price and $(t_2)O_1$ –Value is being matched with $(t_2)O_2$ –Quote. Unlike methods discussed in the literature, SRS is also extensible beyond binary mappings into multiple mappings. This is discussed in the next section.

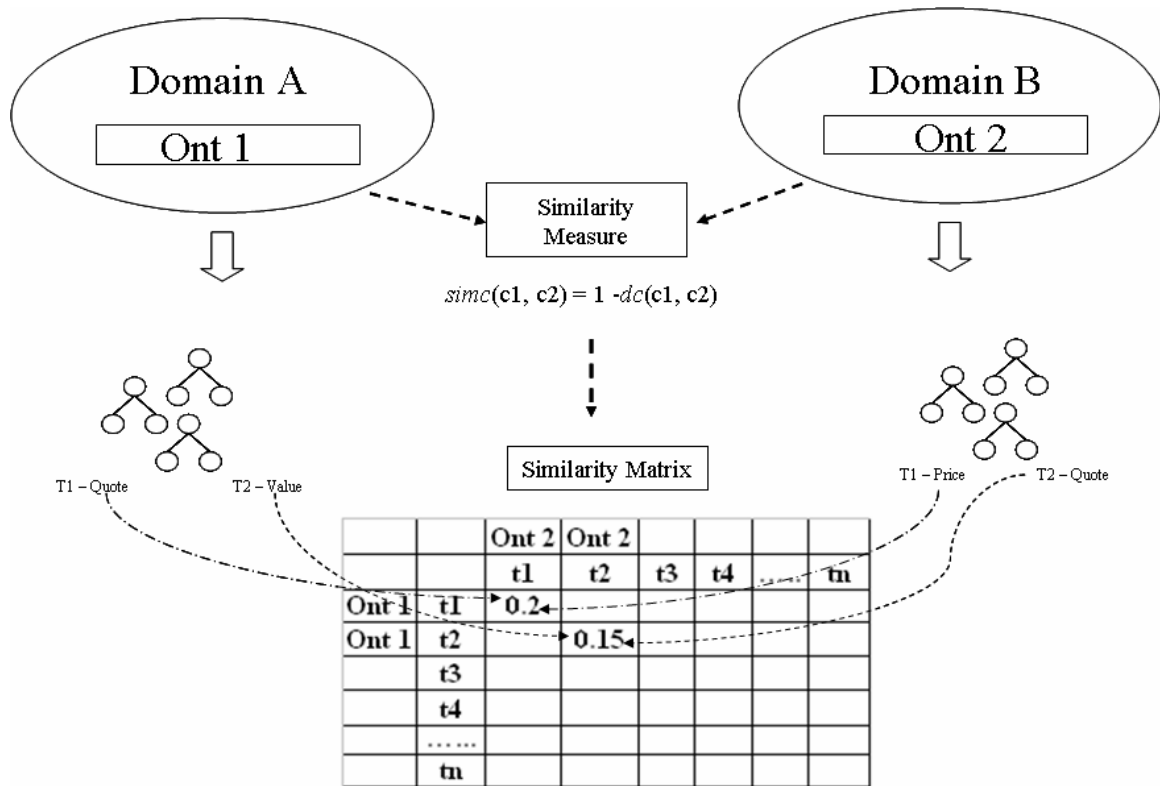


Figure 12 Similarity Measures for Binary Mappings

5.6.2 Similarity Matrix and Multiple Mappings

Multiple Mappings (1:n, m:1, m:n)

As mentioned earlier, literature shows that binary mappings are common among ontology engineers. It is usually easier to measure between two data labels. Binary mappings (1:1) are also easier to implement. However, the main problem with such mappings is that they are based on syntactic matching only. This is not flexible or scalable enough for the Semantic Web. As such multiple mappings between ontologies i.e. $O_1-O_2-O_3-O_4$ must be addressed. In this section, an example of multiple mappings (1:n, m:n, m:1) is

demonstrated. Figure 13, shows four ontologies, which are being matched. Multiple mappings results in the mapping of $(T_1)O_1$ -Price with $(T_1)O_2$ -Quote and $(T_3)O_3$ -toll and $(T_1)O_2$ -Quote. Mappings can be extended to include any concept (T) from any ontology (O). This type of mapping is sometime referred to as complex mappings. The benefit of complex mappings is that it can accommodate more ontologies for mediation purposes. This results in more flexibility and scalability.

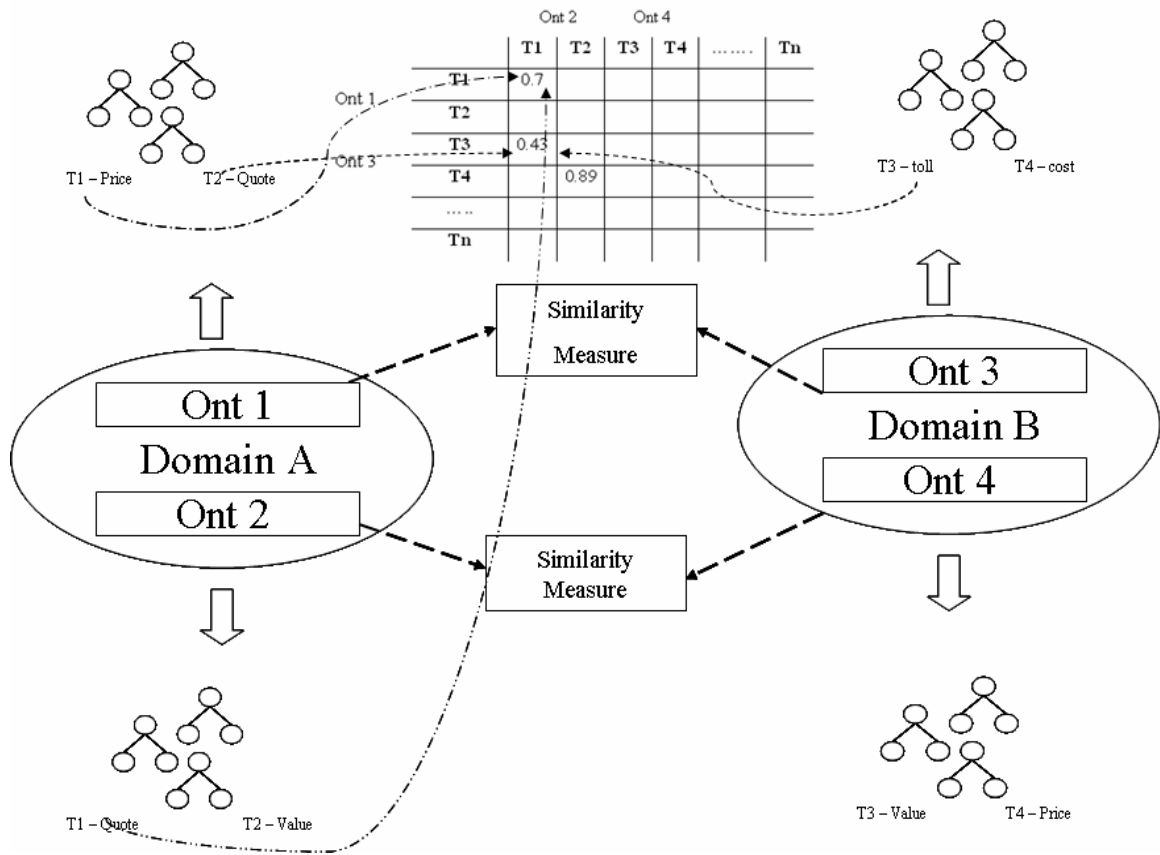


Figure 13 Similarity Measures for Multiple Mapping

5.7 Summary

The main contribution of this chapter is the similarity measure i.e. SRS. This method is unique because it adopts computer science as well as cognitive science measures for computing similarity scores. Another important contribution is the extension of binary (1:1) ontology mapping methods to complex (1:n, m:1,m:n) methods. In order to realize the dream of the Semantic Web more scalable and flexible methods are required. Complex mapping techniques proposed in this chapter helps us to get closer to this goal. The ontology mediation framework focusing on the process methodology, mediation tools and mapping phases is discussed in detail in the following chapter.

CHAPTER 6

6. Ontology Mediation Framework

6.1 Introduction

Ontology mediation is not a trivial task and it involves several processes. Previous chapters have illustrated this with regards to the SRS scores and the matching algorithm. Understanding the whole process flow of ontology mediation with reference to all processes is important and this is the goal of this chapter. This chapter introduces three important aspects of the ontology mediation framework: 1) process methodology, 2) prototyping tools and 3) mapping phases.

There is only one framework that is theoretically feasible, which is the MAFRA framework [11]. However, the drawback here is that the MAFRA framework does not provide a scientific methodology to support it, in contrast to the contribution of this thesis with the introduction of SRS. The goal of this thesis is to provide a guiding principle that is both theoretically and practically feasible for ontologists to master and deploy in their day-to-day tasks. This is by itself a major contribution. In order to provide a proof-of-

concept, a detailed wine ontology example is given in this chapter where two disparate wine ontologies are mediated using steps proposed in the process methodology. The following section describes the process methodology.

6.2 Process Methodology

The proposed process methodology for ontology mediation (see figure 14) has six important steps:

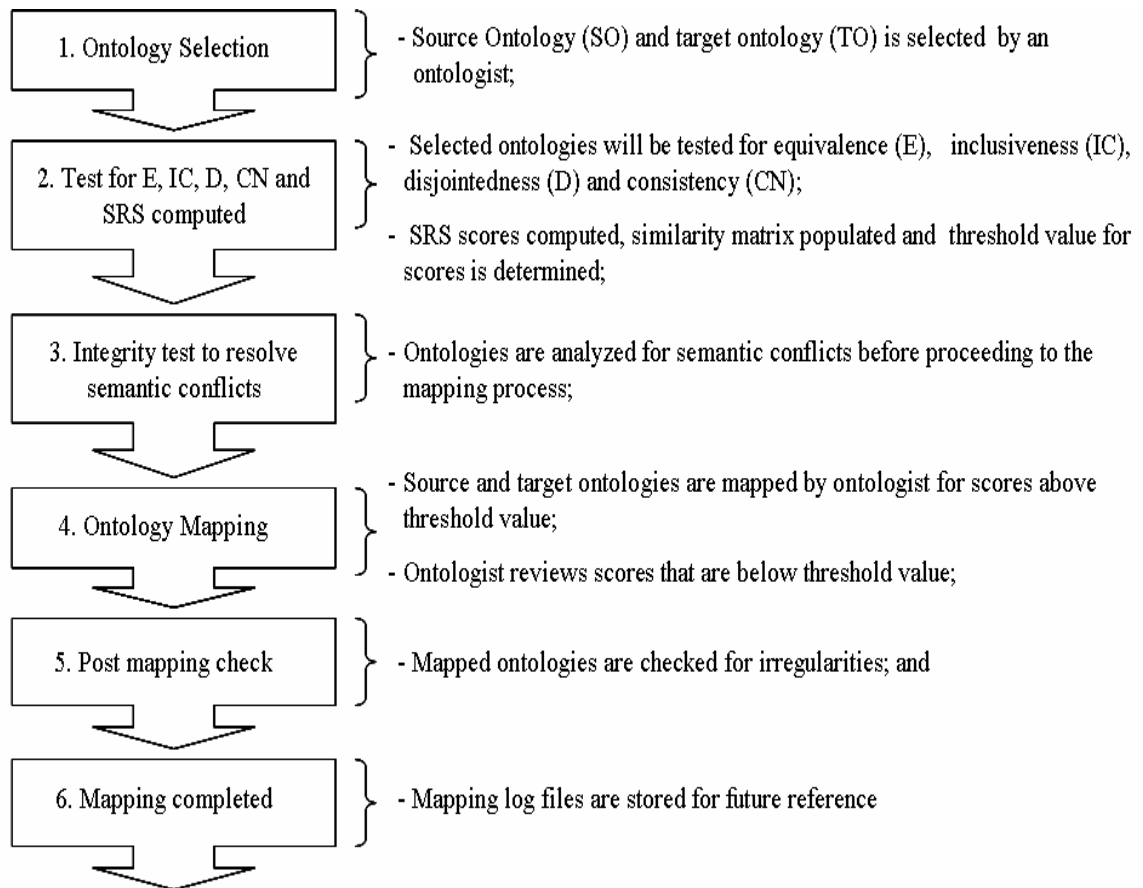


Figure 14 Process Methodology for Ontology Mediation

- Step 1 – In this step an ontologist selects the source ontology (SO) and the target ontology (TO). The source ontology (SO) is really a domain specific ontology that has been developed locally and usually has very detailed definitions of concepts. The target ontology (TO) is an ontology with which the ontologist wishes to map the SO. The TO is usually referred to as the upper ontology which is more general in nature. For example, a white wine retailer who only sells white wine would maintain a very specific white wine ontology in his local ontology. For more general white wine or red wine concepts, he would usually borrow them from an upper ontology. In this case his local white wine ontology would be the SO and the upper ontology would be the TO.

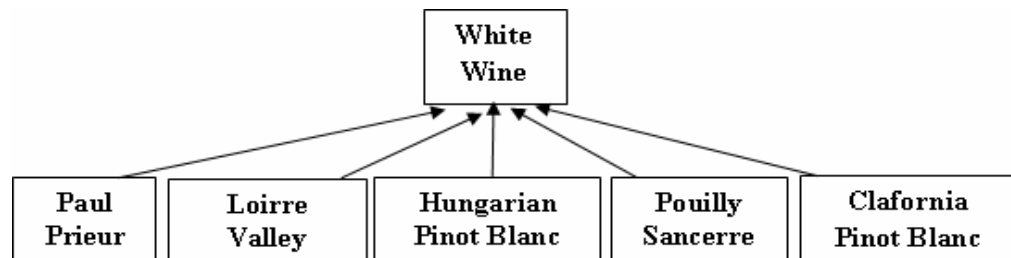


Figure 15 Local ontology classes for white wine

Figure 15, illustrates the locally developed white wine ontology. The five subclasses of white wines that have been defined locally are Paul Prieur, Loirre Valley Sancerre, Hungarian Pinot Blanc, Pouilly Sancerre and Claforia Pinot Blanc. The subclasses are defined as “is-a” relations here. The superclass here is shared from an upper ontology shown in Figure 16. Figure 16, illustrates a how a

local white wine ontology (SO) shares concepts from the upper ontology (TO).

The dotted-line boundary is used to distinguish them.

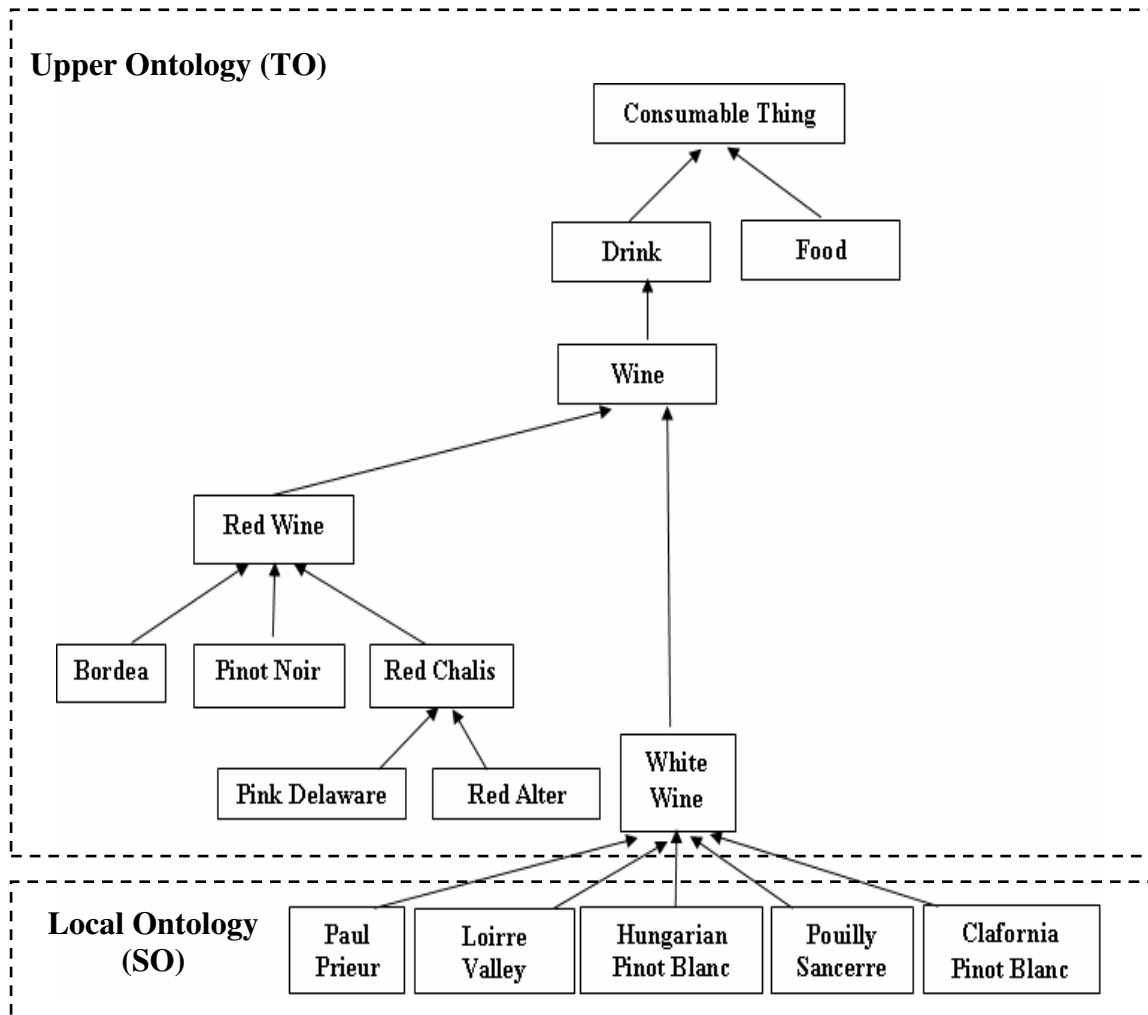


Figure 16 Concepts shared from upper ontology

- In step 2, SO and TO ontologies are tested for equivalence (E), inclusiveness (IC), consistency (CN) and disjointedness (D) as developed previously in our matching algorithm. These tests ensure that the SO and TO have similar concepts to be matched and filters mismatched data formats that have been expressed in the SO

and TO. Once this is achieved, the filtered SO and TO concepts is ready to be syntactically (SYN) and semantically (SEM) matched. This hybrid matching approach is accomplished via the SRS scores discussed in the previous chapter. SRS scores are then generated and the similarity matrix is created for the perusal of the ontologist. Only scores above the threshold (i.e., $t > 0.5$) is populated into the matrix. Lower scores are generated and logged into a file which would be read by the ontologist at a later stage.

- In step 3, an integrity test is performed to ensure that there are no semantic conflicts between SO and TO. First the SO and TO are exported into the OWL (Web Ontology Language) format. Then a reasoning engine can be deployed to analyze the OWL structures to ensure that there are no conflicting concepts. For example, figure 17 shows a French Pinot Blanc concept which has already been defined in the shared upper ontology. Suppose the local ontologist attempts to update the SO with the French Pinot Blanc concept, then the integrity test would automatically highlight a conflict and instruct the ontologist to merge the SO French Pinot Blanc concept with that of the TO, and thus minimize potential errors. Step 3 is crucial before actual mapping can be done in step 4.

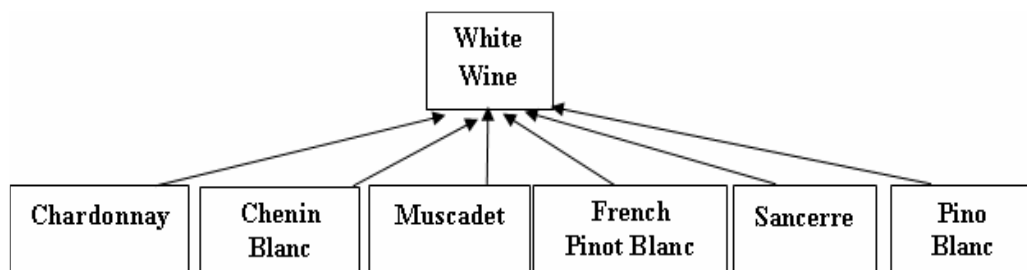


Figure 17 Updated classes in shared ontology

- In step 4, SO and TO are mapped and only SRS scores that are above the threshold are used. Scores below the threshold are analyzed by the ontologist manually. Once the ontologies are mapped they are checked for irregularities in step 5.
- In step 5, irregularities are checked via a post-consistency check. This is performed to ensure that the SO sets of data, concepts and instances remain consistent with TO. This feature also serves as an audit feature in the integration process.
- In step 6, the mapped ontologies are published in the public domain for other wine distributors and retailers to adopt. A log report is created with annotation of the creation date and version information. Every time the SO or TO is upgraded the version information also accompanies it. We believe that during an upgrade process the version information would provide invaluable information as to the differences that exist between the old and new version of the shared ontology. This allows the ontologist to accept new updates or roll back to a previous version.

6.3 Prototyping Tools

The process methodology presented above shows the logical process involved in the proposed mediation framework. For proof-of-concept and to support the physical implementation of this workflow, several tools have been adopted. Figure 18 shows the various tools that have been adopted for each step in the process methodology.

Steps/Process	Tools adopted for proof-of-concept at each step
1	Protégé 3.3.1 –Ontology editor
2	RacerPro – DL OWL reasoner
3	RacerPro, JENA and JESS
4	PROMPT plug-in for Protégé 3.3.1
5	PROMPT plug-in for Protégé 3.3.1
6	PROMPT plug-in for Protégé 3.3.1

Figure 18 Prototyping Tools for the Process Methodology

- In step 1 – Protégé 3.3.1 was used to create both the source (SO) and target (TO) wine ontologies. Protégé is a open source ontology editor developed at Stanford University. Although other editors were available, this tool was chosen because it had the best compatibility to support other plug-ins which had to be configured for the subsequent steps.
- In step 2 and 3 – RacerPro, a reasoning engine developed by Racer Systems GmbH & Co. KG, was adopted. It is a description-logic-based reasoning engine that also supports the Semantic Web Rule Language (SWRL) [61]. Apart from checking consistency, integrity and possible conflicts, RacerPro provides an application programming interface (API) that reads OWL data and reasons assertion boxes (ABox) and terminological boxes (TBox) to draw inferences with given information. JENA was used to test for reasoning of RDF schemas.

JENA is a Java framework that provides an API for RDF [62] and RDFS schema [63]. It also includes a rule-based inference engine. JENA was developed by Hewlett Packard (HP). This was very useful for triggering rules, which are explained in chapter 9. For executing rules another plug-in JESS 7.0 was adopted, which is a

scripting environment for the Java platform. JESS [63] was developed by Ernest Friedman-Hill of Sandia National Laboratory and is a superset of the CLIPS (C Language Integrated Production System) programming language.

- In step 4, 5 and 6 – PROMPT was used to map source (SO) and target (TO) wine ontologies. PROMPT is an open source plug-in which was configured in Protégé for this purpose. It provides a list of matches and displays them to the ontologist via a use interface. The ontologist would then select matches by clicking on the highlighted items that are to be clicked.

6.4. Mapping Phases

This section, presents the mapping phases for the process methodology described earlier. The six steps of the process methodology is divided and organized into four main mapping phases (I-IV). The four phases represent crucial functionalities that exist in a system, providing semi-automatic support throughout the entire ontology mapping process. The following are mapping phases of the proposed mediation framework:

6.4.1 Phase I – Semantic and Syntactic Agreement

Source (SO) and target ontology (TO) is identified by the ontologist who wishes to merge, update or consolidate his SO with an existing upper ontology. This phase also focuses on examining syntax, structure and semantics that cause data heterogeneity

between SO and TO. Consistency and integrity issues are resolved using linguistic and non-linguistic agents which perform matches using WordNET (WN), Gloss Vector (GV), Latent Semantic Analysis (LSA) and Lin (LN) measures.

6.4.2 Phase II - Affinity Measurement (SRS)

This section presents affinity measurement via SRS scores. The SRS score is a hybrid measure that combines several measures including WordNET (WN), Gloss Vector (GV), Latent Semantic Analysis (LSA), Lin (LN) and syntactic matching (SYN). It has been empirically tested and is proven to be highly reliable. As such it is used to find the appropriate agreement between SO and TO.

6.4.3 Phase III - Semantic Bridge

This phase presents a semantic bridge that is formed logically via a two-part approach. The first part is based on SRS scores and the second part is based on SWRL rules. SWRL is useful for fuzzy definitions that can exist within the ontologies. It is also useful for ontological data that is used over and over again. As such, SWRL rules would have to be predefined ahead of time. This improves the productivity as rules can be fired on-the-fly without any intervention from the ontologist. This is clearly explained in chapter 9. SRS and SWRL are complementary in semantically bridging ontologies. This semantic bridge is flexible and can support binary (1:1) as well as complex mappings (1:n, n:1, m:n). Table 1 summarizes the important tasks performed in each phase of the mapping process.

Table 1 Mapping Phases

Phase	Task
Semantic and syntactic agreement	<ul style="list-style-type: none">▪ Identify SO and TO▪ Resolve data heterogeneity for syntax, structure and semantics▪ Check consistency and integrity▪ Use WordNet agent to eliminate ambiguity
Affinity measurement (SRS)	<ul style="list-style-type: none">▪ SRS function is used for similarity measurement▪ Linguistic and lexical similarity is performed▪ Synonyms and hyponyms are analyzed
Semantic bridge	<ul style="list-style-type: none">▪ Source and target ontology is bridged based on similarity▪ Semi-automatic and dynamic mappings performed▪ 1:1, 1:n, n:1 and m:n mappings is performed▪ complete bridging
Semantic consistency and integrity checking	<ul style="list-style-type: none">▪ Ensure results are consistent▪ Check integrity of data▪ Run again if there are errors otherwise end

6.4.4 Phase IV - Semantic Consistency and Integrity Checking

This phase performs checking consistency and integrity of data labels to be matched. If any errors are found they are corrected immediately and the checks are run again. This is repeated until all errors are resolved. Failure to check data labels would result in errors during mapping, and most importantly, would result in wrong information being published in the public domain. This is an important process in both steps 2 and 5 in the process methodology discussed earlier.

6.5. Mediation Architecture Phases and Tasks

There are 4 stages in the mediation phase of the ontology mediation architecture. They are carried out in accordance of the mapping phases in table 1. To illustrate how the

mediation process works, we use the same wine ontology example from our earlier discussion (figure 16).

Table 2 Mediation Architecture Phases and Tasks

Stages	Activity
Stage I	Select the SO (wine v1.owl) and TO (wine v2.owl) their classes, subclasses, slots (attributes) and instances are determined
Stage II	Check for consistency with RACERPro
Stage III	Configure PROMPT (Map), JESS, JENA, SWRL to merge classes, subclasses, slots and instances
Stage IV	Implement post consistency checks -RACERPro

Table 2 shows the four stages involved for implementation. In stage I, the wine ontologies are selected for mapping, i.e., wine v1 and wine v2. Wine v1 is designated as the SO and wine v2 is determined as the TO. PROMPT (see figure 20) does not distinguish them and the order is irrelevant here. Then classes, subclass, attributes (i.e., slots values) and instances for SO and TO are determined before matching is done. In stage II, the ontologies are checked for consistency and integrity using RacerPro. Figure 19, shows the result of an integrity check carried out by RacerPro.

The interface shows the amount of time taken by the synchronizer reasoner to check for concept consistency. Once the ontologies are consistent, we can then proceed to stage III. In this stage several tools are used for mapping data labels. To merge data labels we use PROMPT and for implementing rules we can use SWRL rules. The rules however are executed using JESS, and JENA is used to read the RDF structure.

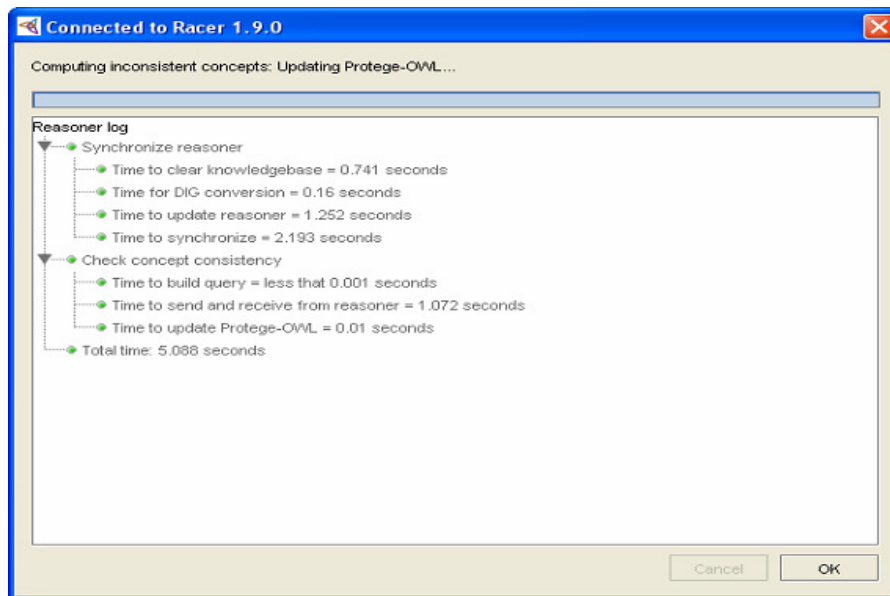


Figure 19 Stage II - Reasoning Phase

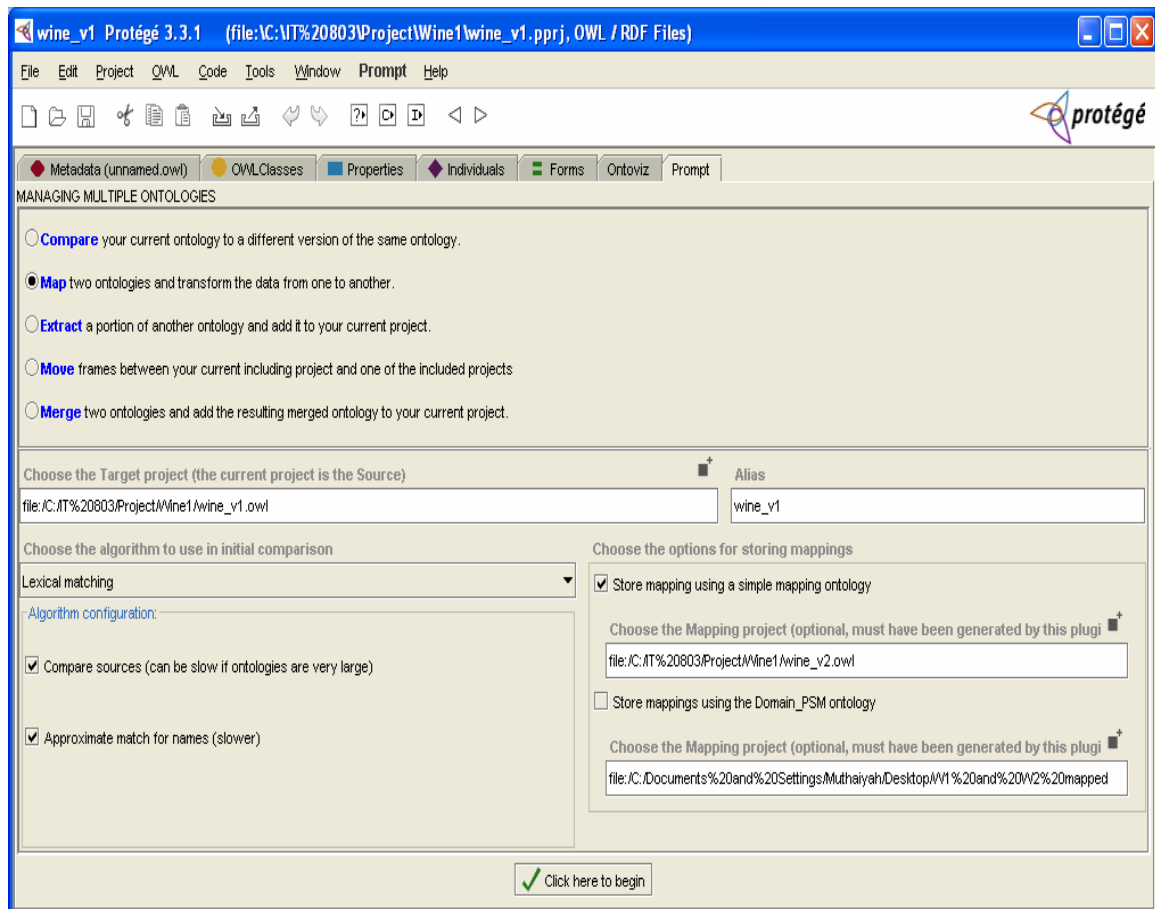


Figure 20 Stage III-Mapping in PROMPT

6.6 Summary

In conclusion, ontology mediation is not a trivial task and involves a lot of processes, phases and tasks. This chapter highlights the ontology mediation framework with reference to a six-step process methodology, mapping phases and implementation stages. To demonstrate how the mediation process works, several tools have been used such as PROMPT, JESS, JENA and RacerPro together with SWRL rules. The proposed framework places considerable importance to consistency and integrity checks between ontologies before and after mapping, unlike other mediation techniques in literature. This is by itself a unique feature of the proposed mediation framework. It uses combines semantic and syntactical matches for determining the SRS scores which gives this framework an edge over existing methods. In the following chapter a more detailed example is provided for the mediation of disparate security policies between virtual organizations (VO) and real organizations (RO) to further illustrate the power of this framework.

CHAPTER 7

7. Mediation Architecture

In this section, the ontology mediation architecture is presented. Ontology mediation is presented here as a process where several components from different layers would work together to resolve semantic differences. Most mediation methods expressed in chapter 4 are string-based. They use prefix, suffix, substring and structure-based similarity measures, which often times fail to produce accurate mappings as they are syntactic. This thesis introduces a bridging framework that combines syntactic and semantic matching (i.e., SRS) to overcome this limitation. SRS produces more accurate matches compared to purely syntactic scores and empirical evidence for this is presented in chapter 8.

Semantic Mediation Architecture (SMA) is a mapping framework that overcomes limitations of existing methods to provide a better platform for dynamic on-the-fly mappings. SRS scores are used for data label matching via the mapping agent (MA) which resides within the semantic engine. This is the core feature of this four layer architecture which creates the semantic bridge between heterogeneous ontological data sources.

There are two ways in which this architecture operates: top down and bottom up. The architecture enables on-the-fly translations via a semantic engine. This process is similar to a real-time interpretation or translation service. Different data label sources are matched using similarity scores via mapping agents to produce a semantic bridge. Figure 21, illustrates the proposed Semantic Mediation Architecture (SMA). Several agent-based systems are deployed to carry out dynamic on-the-fly translations for mediating disparate ontologies. The mapping agent (MA) performs concept mapping using the SRS scores and presents them to broker agents (BA) and search agents (SA). The function of the ontology agent (OA) is to transcribe and lift existing structured, unstructured and semi structured schemas into an ontological format (i.e., OWL). This is done to overcome inherent obstacles that exist within those formats such as inheritance and polymorphism. Ontological formats not only support these features but are also backed up with formal logics that allow it to infer new facts that may not have been explicitly specified.

The SA communicates with the BA and presents all the matched concepts for the perusal of the ontologist or domain expert. Figure 21, highlights how BA, MA, SA and OA work collaboratively to fulfill this goal. This architecture is also directly applicable to solve problems in the area of ontology evolution and shared hierarchical knowledge bases. A prototype has been implemented using the Java Agent Development Framework (JADE). There are two parts to how this architecture works. The bottom up process is where structured, unstructured and semi structured data are transcribed by ontology agents into the semantic engine.

The semantic engine takes the data labels of respective data sources and matches them for concept and attribute similarity using cognitive agents such as WordNet which is a component of the SRS scores. The process methodology and reasoning aspects discussed earlier in chapter 6 are applied here. The top down process collects query information from SA and BA and presents them to the MA in the semantic engine. The ontologist does not have to invest much time searching and then refining their search.

7.1 Mediation Architecture Layers

The layers that make up the mediation architecture are discussed in this section. There are four layers all together: user layer, search layer, semantic layer and data layer (see figure 21). Each of the layers is discussed in detail in the following sections.

7.1.1 User Layer

The user layer has three components: the domain expert, GUI and search agent. The GUI provides an interface for search queries posed by the user. It connects to search agents (SA) and broker agents (BA), which format and send the search to the semantic engine. The search agent (SA) probes user query composition and propagates it to seek relevant data. It determines relevant data sources, sends request to broker and decomposes the query into sub-queries for each source.

7.1.2 Search Layer

The search layer has two components, the broker agent (BA) and the semantic engine. The semantic engine consists of reasoning engine, a WordNet agent and a mapping agent. The broker provides discovery service to discover relevant information resources with respect to a target request, based on ontology descriptions. The semantic engine receives the query from broker and performs matchmaking of concepts. It extracts data from ontology agents (OA), reasons about them and checks for both consistency and integrity. It does both binary (1:1) and complex mappings. Next it updates the integrated ontology and updates a knowledge base. The integrated ontology is versioned and archived for future use.

7.1.3 Semantic Layer

In the semantic layer, ontology agents transcribe the different sources of data. With assume that there are structured, unstructured and semi structured data in the Semantic Web. This is how it would work; an automatic indexing tool will analyze unstructured data (i.e., text of documents) and assign indexed terms for a shared ontology. This tool then creates an ontological representation for the unstructured data. The data is then passed on to the semantic engine for processing.

7.1.4 Data Layer

The data layer is the host to data sources. Various data sources can exist in this context such as structured data (e.g. databases), semi-structured data (e.g. HTML files) and unstructured data (e.g. multimedia files and images).

explicitly defined and others may be fuzzy. As such, VO that represent them in the Semantic Web will not be able to mediate between specified security policy schemas accurately. One such case would be an authorization policy that is not semantically equivalent [64]. Therefore, semantic mediation would be necessary to create homogeneous security policies for the Semantic Web environment that can be referred to by all entities, irrespective of their actual security policies.

We propose a security policy reference ontology for this, and show how to create it via ontology mediation. This section highlights ontology mediation to map security policies of virtual organizations (VO) with security policies of real organizations (RO). The goal is to develop a common domain model for security policy via semantic mapping. This helps to mitigate interoperability problems that exist due to data heterogeneity problems among security policies of various (VO) and (RO).

Only one aspect of security policy ontology mediation is discussed in this section to illustrate this process: authorization and authentication. Ontology mediation of other security policies (i.e., integrity, repudiation and confidentiality) can be dealt with a similar manner. We will now use the following example (see figure 22) to demonstrate how structural heterogeneity problems can exist for security policies amongst RO and VO. Assume that the VO labeled it's security policy as "authentication" and RO has its own authentication policy, which it labels as "authorization". Although the policies appear to be similar, structurally they are different.

In the authentication policy structure, the VO has “UserPassword” and the RO has “Token”. However the other classifications in the tree remain the same. The next example in table 3 shows semantic heterogeneity that exists for the same entities. From the diagram, one would assume that the PKI data definitions of both entities are the same. However this is not always true and it is clear from the example below that the PKI data definitions are different in the final policy.

The X.509 certificate specifies the association between a public key and a set of attributes such as subject name, version, issuer name, serial number and validity interval. The OASIS¹² specification describes the use of the X.509 authentication framework in greater detail and SOAP¹³ Message Security specification (WS-Security) describes procedures for message exchange. In order to achieve interoperability and to resolve the heterogeneity issues of security policies below, semantic integration is needed. This is because syntactic mapping of PKI definitions alone would result in a semantic loss and the goal of this study is to preserve both syntactic and semantic mappings. The goal here is to specify a Security Policy Domain Model (SPDM), via ontology mediation. SPDM will act as a common security policy for both entities by inheriting existing attributes and sometimes adding or modifying attributes between them so that the policies become commonly applicable to both entities without compromising existing security standards.

¹² Organization for the Advancement of Structured Information Standards (<http://www.oasis-open.org/who/>)

¹³ Simple Object Access Protocol

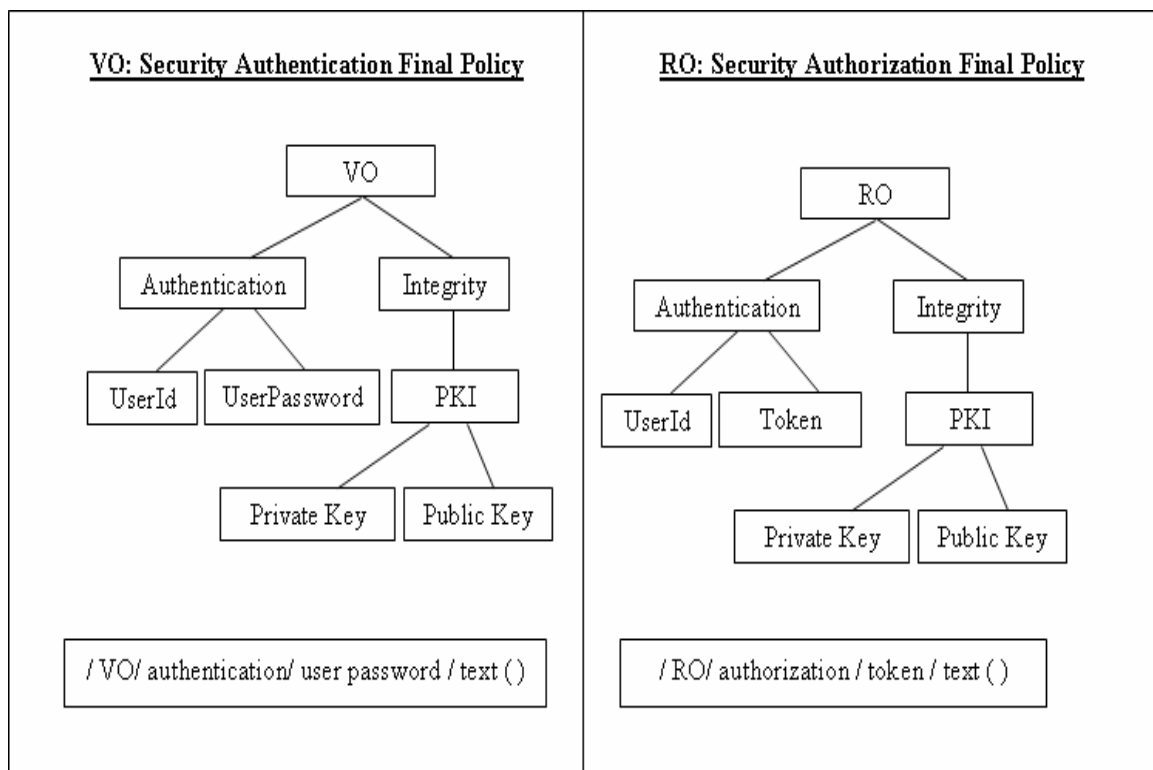


Figure 22 Security Policies –Structural Heterogeneity Problem

Table 3 Semantic Data Heterogeneity Problems

<u>VO: Security Authentication Final Policy</u>				<u>RO: Security Authentication Final Policy</u>			
PKI – X.509 Certificate				PKI – X.509 Certificate			
Name	Identifier	Category	Version	Name	Identifier	Category	Version
Final Policy	UserId	Certificate	3	Final Policy	UserId	Token	3
Note: Object representation conflicts				Note: Object representation conflicts			

Explicit differences in security policies can exist among organizations that transact over the Semantic Web. It's clear from the examples provided that RO and VO have differences in their security policy design.

SPDM fully integrates and eliminates semantic differences that exist at the object and attribute levels of the security policy data structures. The assumption here is that only two entities exist, i.e. RO and VO. In earlier examples, authorization security policy is specified for RO and VO. Having identified the structural and semantic differences we can proceed to carry out reasoning regarding those policies. Reasoning to check for consistency in both of the ontologies can be performed using RacerPro, as discussed previously in chapter 6.

Inconsistencies are resolved to produce a consistent SPDM, because mapping cannot be produced on inconsistent ontologies. The tool also understands the ontology to be mapped and sends a message to the reasoning server via http GET and POST messaging. This is a client-server Java-enabled engine that produces a summary report after the reasoning process is complete. Although this process is both very time both cumbersome and time consuming, it is crucial to obtain consistent mappings. The analysis provides us an overview of the similarities and differences in the structure and semantics of those security policies. Next, mapping and consolidating of the policies are done.

7.3 Understanding Security Policy

The term “security policy” has many different meanings and can be interpreted in many different ways. A security policy is a statement of what is allowed, and what is not allowed [65]. The existence of various interpretations is rooted in two observations. Firstly, security policy is a context-dependent notion (e.g., computer security policy,

information security policy, etc.) and secondly, even in the same context, specific kinds of security policies have been developed to meet specific needs (e.g., confidentiality security policies in military environments, etc.).

To effectively manage security policies we must be able to produce compatible policy representations. The existence of a large number of representation methods leads to the conclusion that security policies, even if semantically compliant, can be represented in ways that differ substantially in terms of formalism, structure, and hierarchy. This raises obstacles to their reconciliation. Multiple interacting security policies require semantics to be managed and manipulated. The security policy semantics ontology is an efficient means for achieving this. Figure 23, shows the six steps of the SPDM development lifecycle to produce the SPDM [15]. The steps that are illustrated here are analogous to the process methodology discussed previously in chapter 6.

The security policy ontology is created for RO and VO first (step 1) and then exported to OWL. We use Protégé to build and export the OWL and perform reasoning with RacerPro. The reasoning process in RacerPro is made up of three sub processes: 1) checking for consistency in the ontologies to be mapped, 2) classifying their taxonomy and 3) computing their inferences. Figure 23, summarizes those processes in (steps 2-3). The reasoner will eliminate all inconsistencies and align any new additions that are inserted into the ontology. Upon successful elimination of inconsistencies in (steps 2-3) we begin mapping ontologies using the PROMPT plug-in (step 4). Merge functions (step 5) are carried out if ontologies can be combined and lastly a mapping log is generated

(step 6). In the dynamic semantic web environment the policy mapping would be done on-the-fly seamlessly. When the VO and RO agree on a common security policy for processing a transaction, a new contact agreement is formed. Since the Semantic Web is a dynamic environment with multiple new entrants coming into the virtual supply chain, there is a need to dynamically map local and global policies. The wrapper, as explained earlier solves this problem. Mapped policies would also represent virtual temporary contracts (VTC), which can be stored in a knowledge base for future use.

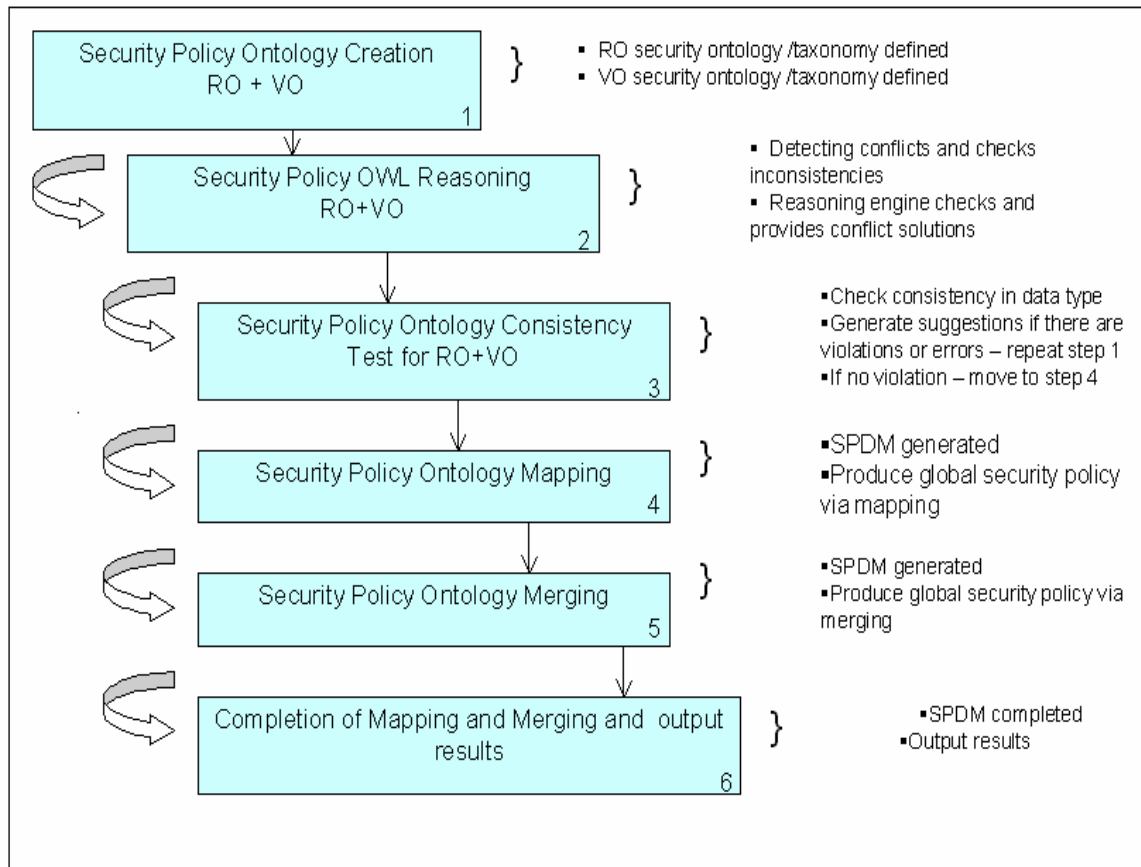


Figure 23 SPDM Process Methodology

The methodology for semantic mapping in the semantic web would be to have a WS-Security ontology mapping system, which has its own rule ontology and rule parser. However, aligning and mapping the security policies of the constant influx of new entrants is not a trivial task. To demonstrate this, we employ different plug-ins in Protégé such as Algernon¹⁴ and PAL¹⁵. Also a merge¹⁶ function was carried out with the PROMPT plug-in. PROMPT has 4 divisions (i.e. merge, extract, move frame and compare). RacerPro was used to test the overall consistency of SPDM.

7.4 Mapping with Protégé and reasoning with RacerPro

There are many aspects of security policy that need to be reconciled between the global and local environments. A complete taxonomy is needed so that the reconciliation can be performed quickly. Literature indicates that a comprehensive ontology for security policy does not exist. Therefore one had to be created for the VO and the RO [64, 65]. Security policies can be divided into authentication, authorization, integrity, access control, non-repudiation and confidentiality policies individually or a combined version of all. This chapter focuses on the authorization security policy for SPDM. Table 4, shows two scenarios, A and B. The “final policy” data labels for authentication in scenario A are “UserId” and “UserPassword” and authorization data labels are “UserId” and “Token”. SPDM shows the mapped data labels, which are “UserId” and “UserPassword”.

¹⁴ Protégé plug-in is used for forward and backward chaining rules.

¹⁵ Protégé plug-in is used for expressing constraints.

¹⁶ Merging function allows projects to be merged after resolving the common concepts between them.

Table 4 Merging two SPRO Policy Data Elements

	Security Policy – VO (Authentication) <i>Before Mapping</i>	Security Policy – RO (Authorization) <i>Before Mapping</i>	Compatible Security Policy – SPDM = VO+RO (Authorization) <i>After Mapping</i>
SCENARIO A	A=Identifier +Password <? xml version="1.0" ?> <Final_Policy > <Entity> <Type/> <Identifier>UserId</Identifier> <Password> UserPassword </Password>	A=Identification with token ID <?xml version="1.0" ?> <Final_Policy > <Entity> <Type/> <Identifier>UserId</Identifier> <TokenId>Token</TokenId>	A=Password ticket <? xml version="1.0" ?> <Final_Policy > <Entity> <Type/> <Identifier>UserId</Identifier> <Password Ticket> UserPassword </Password Ticket>
SCENARIO B	B=X.509 Certificate <?xml version="1.0" ?> <Final_Policy > <Entity> <Type/> <Identifier>UserId</Identifier> <Certificate>X.509</Certificate>	B=X.509 token <?xml version="1.0" ?> <Final_Policy > <Entity> <Type/> <Identifier>UserId</Identifier> <Token>X.509</Token>	B=X.509 Certificate & token <?xml version="1.0" ?> <Final_Policy > <Entity> <Type/> <Identifier>UserId</Identifier> <CertificateToken>X.509 </CertificateToken>

The identifier remains as “UserId” but Password and TokenId becomes “Password Ticket”. In scenario B, the “final policy” data labels for authentication are “UserId” and “X.509” (Certificate) before mapping. After mapping SPDM shows the new data labels as “UserId” and “X.509” (Certificate Token). Merge is complete without any discrepancies and SPDM is the common global policy for authorization, which is agreed by VO and RO. Heterogeneity in the data labels above, were reconciled to arrive at the common policy (SPDM). The scenario above is mapped into XML and in order to maintain consistency we use RacerPro to do reasoning.

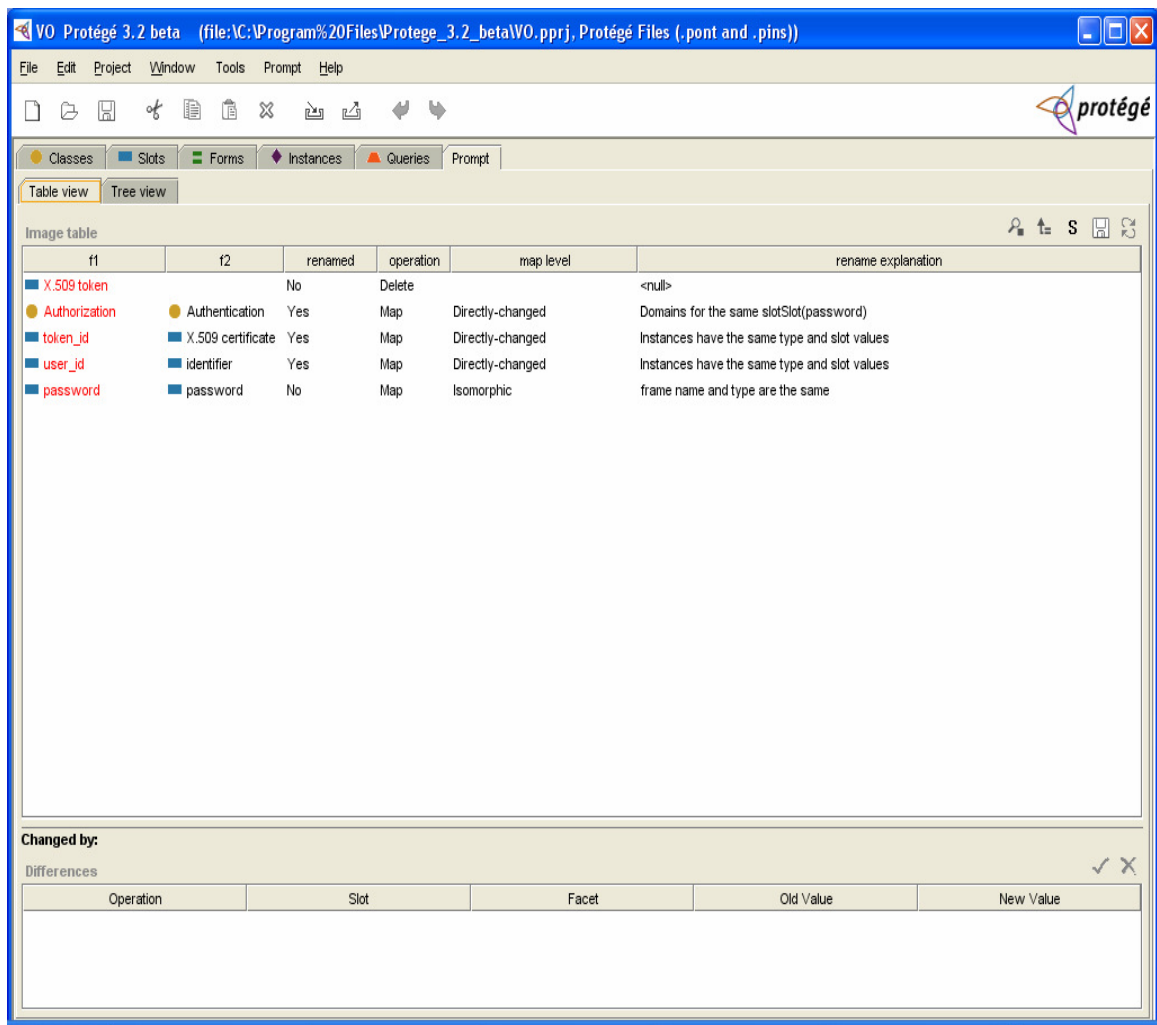
Typically RacerPro would produce statistics for tests carried out and produces a log report (i.e. Mlog) (see appendix III), which will address the following:

- Total number of generated suggestions
- Number of generated suggestions that were followed by the user
- Total number of conflicts detected
- Number of conflict solutions used
- Total number of KB operations

First, ontologies for Scenarios A and B have to be created in Protégé. Each ontology must then be exported to OWL prior to reasoning. After that reasoning is done separately for A and for B. Notice that the reasoning results are error free which means mapping can be carried out from this point. If there were errors, they would have to be corrected before mapping can be done. We do this to ensure that the data is consistent before mapping. Errors are usually related to inconsistent classes, concepts, slots and attributes. For instance if the authorization policy for scenario A had inconsistent attributes, the user would be prompted to correct those errors. These errors must be amended before proceeding to the next step, which is mapping.

Appendix III, shows the successful reasoning (i.e., number of conflicts detected =0) performed for Scenario A (RO and VO). Reasoning is also done for Scenario B (RO and VO) and after successful reasoning we perform mapping. Mlog (mapping log) outputs in (appendix III) also show the number of conflicts detected which in this case is zero. SPDM (Figure 23, step 6) would be the end result. Figure 24, shows the mapping

suggestions provided by PROMPT to the ontologist. Isomorphic concepts are concepts that have similar names and frames such as “password”. Such concepts are usually recommended to be directly mapped. Concepts that have similar instance type and slot values are usually recommended to be renamed and directly mapped. Isomorphic concepts are not renamed as they have similar names. Successful mapping with PROMPT and merge for scenario A is shown in appendix III.



f1	f2	renamed	operation	map level	rename explanation
X.509 token		No	Delete		<null>
Authorization	Authentication	Yes	Map	Directly-changed	Domains for the same slotSlot(password)
token_id	X.509 certificate	Yes	Map	Directly-changed	Instances have the same type and slot values
user_id	identifier	Yes	Map	Directly-changed	Instances have the same type and slot values
password	password	No	Map	Isomorphic	frame name and type are the same

Changed by:				
Differences				
Operation	Slot	Facet	Old Value	New Value

Figure 24 Mapping results for scenario A

7.5 Summary

Ontologies raise the level of specification of knowledge by incorporating semantics into data representations and promoting their exchange in an explicitly understandable form. Stand-alone ontologies provide zero interoperability therefore we need accurate mappings. Ontology mediation via semantic data integration carried out in this chapter gives an insight into the benefits of ontology mapping in the area of security. This is significant for building a common information model and trusted domain for federated services. VO represents RO for carrying out transactions to fulfill demands from user. Policies of RO and VO of agent ontologies may differ greatly and so we need to look into implementing security taxonomy, which can be common to all.

CHAPTER 8

8. Empirical Evaluation of SRS

Concept similarity is the measure of linguistic similarity between concepts. There are many ways to evaluate and justify the accuracy of similarity measures but the best evaluation would be to compare them to human cognitive responses or human evaluation. In today's world of ontology mapping, input from a human being (i.e., domain expert) is still required. The matching algorithm only presents potential candidates that can be mapped but the domain expert will use his cognitive judgment to select the data labels to be matched. As such the validity of similarity measures is best ascertained by how close it correlates to responses provided by domain experts. By matching SRS scores to scores provided by human subjects, we can determine how effective SRS is.

PROMPT [33] for example, provides results for matching and gives suggestions but the ontologist still decides which concepts are to be chosen for actual mapping. Most semi-automated mapping systems use this approach. Since human input is still much required and if SRS mimics to a certain extent human responses, then its validity can be justified. As such a comparison of SRS with human cognitive response (HCR) is needed. The following sections describe the experiments that were carried out with human subjects and responses received. In view of evaluating SRS a two-part survey was conducted.

Since SRS combines Lin (LN), Gloss Vector (GV), WordNet (WN) and LSA measures, it works well in taxonomies and ontological structures. Currently, these measures only support “is-a” relations and WordNet (WN) is limited to the number of definitions that is available in it. As such other measures that were used in SRS helped to produce a more reliable similarity score.

8.1 Experiment Design

An experiment was conducted via surveys (Appendix I & II) to compare SRS results to results of human cognitive responses (HCR). Questionnaires were distributed to a carefully chosen group of individuals in order to get the best results. English teachers of a local high school were chosen as respondents. The reason for this was because sample experts were required to simulate real-life situation where domain experts would usually use their cognitive responses to match data labels. The main goal here is to validate SRS scores. These teachers have been formally trained to teach English as a Second Language (TESL) and are very proficient in the English language. This in the real world is equivalent to a domain expert’s judgment or opinion when they choose and select the data labels to be matched. Fifty survey forms were distributed via e-mail, fax, regular mail and face-to-face interviews.

Fifty completed survey responses were received, thus this gives the study a 100% response rate. However, twelve survey responses had to be eliminated from the analysis as the respondents did not correctly follow the instructions given in the survey form.

Thus, the balance of thirty-eight responses was used for data analysis. The survey form was divided into two parts. Part 1 focused on testing human judgment for similar word-pairs. The higher a respondent's score would mean the higher the similarity of the selected words or concepts, in their expert opinion. This was to test the hypothesis if SRS scores would match human cognitive judgment and if so to what extent it does. Part 2 focused on testing whether or not a subject's response changed if the context of the search was modified. This was based on the hypothesis that a search word used is largely associated with the context of search. Further details of the questionnaire are explained in the next section.

8.2 Survey Construct-Part I

Part I of the questionnaire has a thirty word-pair list, which was used to judge semantic relatedness of word pairs. The respondents were asked to rank the word pairs on a scale of 0 to 5, for all thirty word-pairs; 0 for unrelated word pairs and 5 for highly related word pairs based on their cognitive similarity judgment. Respondents were allowed to use any number on a scale of 0 to 5 including decimal numbers and the survey was not based on a Likert scale.

Respondents were requested not to assign the same rank twice for the same word category. For instance if "lad" appeared twice for a word pair then they should not give the same rank for another instance which has "lad" in it. This was to ensure that their previous answers did not have an effect on the new answers and also to prevent bias

answers. The benchmark for such an experiment was inspired by previous work conducted by Rubenstein and Goodenough who used sixty-five word pairs in their research [66]. Some years later Miller and Charles repeated the same experiment but with a subset of thirty word-pairs [1] out of sixty-five.

This research uses the same subset list of thirty word-pairs used by Miller and Charles. The reason for this is that these word pairs had already been tested for all degrees of semantic relatedness and there has been no major discrepancy in the results of those two studies. This shows that not much has changed in the way humans perceive similarity, and allows us to repeat this experiment with confidence. Out of thirty, ten are highly related word-pairs (which should yield scores between 3 and 5), ten intermediately related pairs (which should yield scores between 1 and 3) and ten unrelated pairs (which should yield scores between 0 and 1). Rubenstein and Goodenough used fifteen subjects in their experiment; Miller and Charles used thirty-eight subjects and this research resulted with the analysis of responses of thirty-eight subjects as well. Appendix I, lists the word-pairs used in this experiment.

8.3 Survey Construct -Part II

Part 2 has five distinctly different sections (i.e., A, B, C, D and E). Each part describes a situation or context with a set of entity sets (i.e., words). Subjects were asked to rank, according to their judgments of similarity all eleven entity sets provided in each part. Taking into consideration that some of the respondents might not be familiar with the

entity sets, a list of their definitions was provided in the survey form. A total of 23 definitions related to travel and accommodation were provided. The primary goal here was to test if human responses changed when there was a change in context used for determining similarity. Unlike Part I, this section highlights the importance of context in the process of mapping concepts. The hypothesis here is that similarity judgment scores do change if the context changes. Subjects were asked to rank the entity sets between (1 and 11) based on their similarity judgment and the given context. Each section had eleven entity sets and this resulted in a total of fifty-five responses.

Sections A, B and C prompt users to judge the same set of entity sets, but using different contextual information. Section A represents the default case (i.e. similarity of a “place to stay”) with no explicit contextual information. Sections B and C provide specific context (i.e. “place to stay” for a meeting, seminar or conference) for the former and (“place to stay” if someone was only keen on staying first class) for the latter. Section D uses the same context as C but lists different entity sets. Section E uses a set of outdoor activities, for its contextual information. The hypothesis here is that semantic and syntactic measures alone do not suffice for concept matching; context must be determined as well. Contextual information is described as a natural-language statement and context specification as part of concept mapping is future work, which I would like to explore. Automatic extraction of contextual information from natural-language statements would be necessary for measuring similarity in this way.

8.4 Data Analysis and Hypothesis Testing

Data analysis and hypothesis testing was done using Microsoft Excel and SPSS version 13 [67].

8.4.1 Data Analysis – Survey Part I

This experiment was carried out to test whether or not the 30 word pairs used for human judgment (HCR) positively correlated with SRS scores.

Table 5 Word pairs, SRS and HCR scores

Word Pairs	SRS Scores	SRS Rank	HCR Rank
car - automobile	1	5	4
gem - jewel	0.042	0.21	1
journey - voyage	0.806	4.03	4
boy - lad	0.786	3.93	4
coast - shore	0.643	3.215	3.5
asylum - madhouse	0.769	3.845	4
magician - wizard	1	5	4
midday - noon	1	5	5
furnace - stove	0.576	2.88	3
food - fruit	0.297	1.485	1.5
bird - cock	0.655	3.275	3
bird - crane	0.358	1.79	2
tool - implement	0.354	1.77	2
brother - monk	0.429	2.145	1
lad - brother	0.424	2.12	1
crane - implement	0.159	0.795	0.5
journey - car	0.417	2.085	1
monk - oracle	0.129	0.645	2
cemetery - woodland	0.062	0.31	1
food - rooster	0.131	0.655	2
coast - hill	0.237	1.185	1
forest - graveyard	0.09	0.45	0.6
shore - woodland	0.107	0.535	0.5
monk - slave	0.251	1.255	1

coast - forest	0.165	0.825	0
lad - wizard	0.052	0.26	0
chord - smile	0.076	0.38	0.3
glass - magician	0.066	0.33	0
rooster - voyage	0.041	0.205	0
noon - string	0.094	0.47	0.5

Table 6 Case Summary Output for SRS and HCR scores

Word Pairs			SRSRank	HCR Rank
asylum - madhouse	1		3.845	4
	Total	N	1	1
bird - cock	1		3.275	3
	Total	N	1	1
bird - crane	1		1.790	2
	Total	N	1	1
boy - lad	1		3.930	4
	Total	N	1	1
brother - monk	1		2.145	1
	Total	N	1	1
car - automobile	1		5.000	4
	Total	N	1	1
cemetery - woodland	1		.310	1
	Total	N	1	1
chord - smile	1		.380	0
	Total	N	1	1
coast - forest	1		.825	0
	Total	N	1	1
coast - hill	1		1.185	1
	Total	N	1	1
coast - shore	1		3.215	4
	Total	N	1	1
crane - implement	1		.795	1
	Total	N	1	1
food - fruit	1		1.485	2
	Total	N	1	1
food - rooster	1		.655	2
	Total	N	1	1
forest - graveyard	1		.450	1
	Total	N	1	1
furnace - stove	1		2.880	3

	Total	N	1	1
gem - jewel	1		.210	1
	Total	N	1	1
glass - magician	1		.330	0
	Total	N	1	1
journey - car	1		2.085	1
	Total	N	1	1
journey - voyage	1		4.030	4
	Total	N	1	1
lad - brother	1		2.120	1
	Total	N	1	1
lad - wizard	1		.260	0
	Total	N	1	1
magician - wizard	1		5.000	4
	Total	N	1	1
midday - noon	1		5.000	5
	Total	N	1	1
monk - oracle	1		.645	2
	Total	N	1	1
monk - slave	1		1.255	1
	Total	N	1	1
noon - string	1		.470	1
	Total	N	1	1
rooster - voyage	1		.205	0
	Total	N	1	1
shore - woodland	1		.535	1
	Total	N	1	1
tool - implement	1		1.770	2
	Total	N	1	1
Total	N		30	30

The SRS scores obtained, were first adjusted (SRS Rank) to reflect the same scales used by HCR. All thirty-eight HCR Rank results for the total of thirty word-pairs were averaged and matched. Table 6, shows all the thirty word-pairs (N) that have been tested for this experiment. Scores for SRS Rank refers to SRS scores that have been normalized. The averaged SRS and HCR results are shown and the minimum score was 0.21 and the highest was 5 for SRS Rank. HCR Rank has a minimum score of 0 and maximum of 5

for the same word-pairs. The reason for this is because syntactic match uses prefixes and suffixes for matching, which is not how we humans cognitively rank similarity. Number of word pairs shows all thirty word-pairs for HCR and SRS that have been ranked and entered and no data was missing (Valid N). The mean score is shown as 1.87 for SRS Rank and 1.78 for HCR. HCR has a lower mean because of the many zero scores. Standard deviation scores show not too much disparity exists between ranks, SRS Rank with 1.60 and HCR Rank with 1.52. This indicates close proximity to the mean exists for both scores and this shows a healthy dispersion.

Table 7 Descriptive Statistics on SRS and HCR scores

	N	Minimum	Maximum	Mean	Std. Deviation
SRS Rank	30	.21	5.000	1.87	1.60
HCR Rank	30	0	5	1.78	1.52
Valid N (listwise)	30				

Histograms below present the descriptive statistics of SRS Rank and HCR Rank scores. The highest frequency for SRS Rank scores is 8, which represents ranks between (0-0.5). This shows that there is a high number of dissimilar word-pairs. Lowest frequency, 1 represents SRS Ranks between (2.5-3.0) and (4-4.5) and this means that a lower number of similar words exists. HCR rank has the highest frequency 7, for ranks (1-1.5) and the lowest 0 for ranks (2.5-3). This indicates that more word-pairs exist for lower similarity scores and no word-pairs were found for average similarity scores. SRSRank has a more normalized output compared to HCRRank due to this reason (see figure 25 and 26).

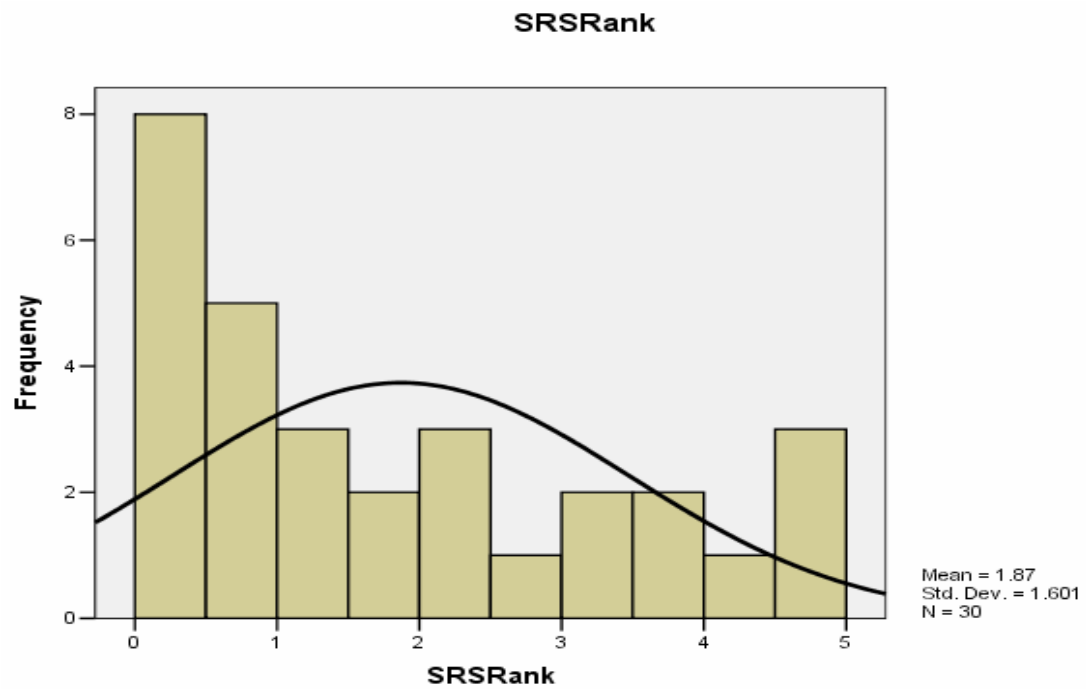


Figure 25 Case Summary Output for SRS scores

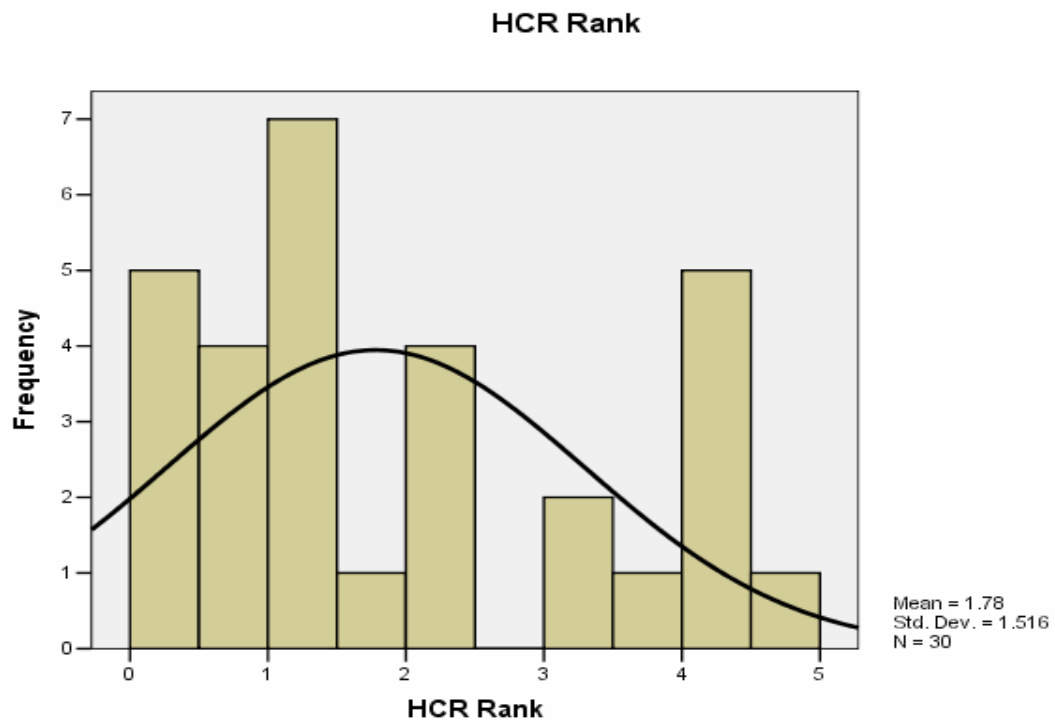


Figure 26 Case Summary Output for HCR scores

Table 8 shows case processing summary for all the 30 word pairs (N) that have been included for this experiment. Included case shows 100%. This means that all cells were populated and responses were recorded. Excluded case shows that none of the ranks were left out. SRS and HCR ranks with zero exclusion shows that none of the data were missing from the word pair table.

Table 8 Case Processing Summary

	Cases					
	Included		Excluded		Total	
	N	Percent	N	Percent	N	Percent
SRS Rank * Word Pairs	30	100.0%	0	0.0%	30	100.0%
HCR Rank * Word Pairs	30	100.0%	0	0.0%	30	100.0%

8.4.2 Reliability Analysis

Reliability analysis used here is the Cronbach's Alpha model (Cronbach, 1951), which shows the internal consistency, based on the average inter-item correlation of HCR Rank and SRS Rank. Reliability is defined as the proportion of variances of the survey. The computation of Cronbach's alpha is given below. It is based on the number of items in survey (k) and the ratio of the average inter-item covariance to the average item variance. Table 9, shows a positive correlation coefficient of 0.957 or (95.7%). This indicates a strong relationship between SRS scores and the human evaluation scores (HCR).

$$\text{Cronbach's alpha } \alpha = \frac{k (\text{cov} / \text{var})}{1 + (k-1) (\text{cov} / \text{var})}$$

Table 9 Reliability Analysis

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
0.957	0.958	2

8.4.3 Reliability Function (Precision and Relevance Test)

Precision, recall and the F-measure are currently standard test measures for IR systems. However, only the precision measure is appropriate for this study. A new test called the reliability test is introduced in this thesis for validating SRS scores. The reliability test is a function of precision and relevance, i.e., Reliability (REL) = {precision and relevance}. Precision is denoted as (P_s) and relevance as (R_L), thus the function for reliability can be denoted as:

$$REL = \{P_s \text{ and } R_L\} \quad (16)$$

Precision (P_s) and relevance (R_L) is measured as:

$$P_s = \frac{\text{number of correct responses}}{\text{total number of responses}} \quad (17)$$

$$R_L = \frac{\text{number of relevant responses}}{\text{total number of responses}} \quad (18)$$

There are two parts to reliability: precision and relevance. Any semi-automated ontology mediation system is meant to reduce the workload of the ontologist. The ontologist must be provided with reliable information before using his own discretion to finally decide on the concepts to be merged. The hypothesis here is that SRS scores, which include

syntactic and semantic measures, are more reliable to an ontologist compared to using syntactic measures only. The null and alternate hypothesis is stated as:

(H_0) : *SRS Rank scores are less reliable than SYN scores*

(H_1) : *SRS Rank scores are more reliable than SYN scores*

8.4.4 Hypothesis Testing - Reliability Function

8.4.4.1 Precision (P_s)

Considering that there are 30 word-pairs (Part I of survey), in order to calculate precision (P_s) all the SRS Rank responses were first converted into integers. Then, HCR responses were matched against with. The idea was to compare exact matches only. Out of 30 pairs, there were 12 exact matches. Although there were some that were really close, but because they were not exact matches, those scores were not considered for this test. The final precision score for the *SRS Rank scores* was 40% ($P_s = 12/30$), which is given in the equation above, 12 correct responses resulted out of 30 responses in total. The precision score for *SYN scores* resulted in 5 correct responses out of 30 responses in total. The precision score for SYN scores was 16.67% ($P_s = 5/30$). Thus SRS provided better precision scores, and therefore the null hypothesis (H_0) was rejected.

8.4.4.2 Relevance (R_L)

The same number of word pairs was tested and relevance (R_L) was measured. The SRS Rank scores were converted integer numbers and HCR responses were matched against them. The relevance score (R_L) for the *combined syntactic and semantic* match was

96.67% ($R_L = 29/30$). This means that there were 29 relevant responses out of 30 responses. The relevance score (R_L) for SYN scores resulted in 22 relevant responses out of 30 responses in total. SYN scores resulted in 73.33% ($R_L = 22/30$) only. SRS Rank scores provided better relevance scores and the null hypothesis (H_0) is rejected.

8.4.4.3 Summary

In summary the SRS Rank scores was 40% for *precision* ($P_s = 12/30$) and 96.67% for *relevance* ($R_L = 29/30$). SYN scores resulted in only 16.67% for *precision* ($P_s = 5/30$) and 73.33% for *relevance* ($R_L = 22/30$). We can therefore conclude that SRS Rank scores produce more relevant and precise scores, which is why SRS Rank scores would be more reliable (REL) for mediation services (figure 27).

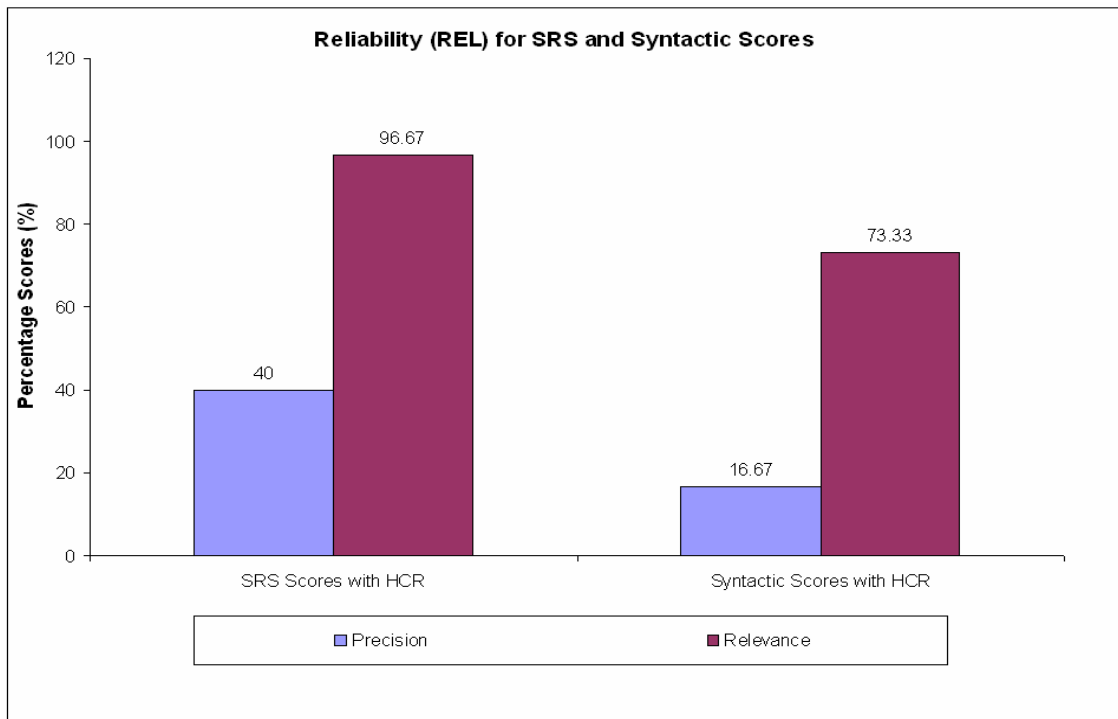


Figure 27 Comparing the Reliability SRS and Syntactic Scores

8.5 Hypothesis Testing – Survey Part I

Part I is focused on testing human judgment for similar word-pairs. The hypothesis is that SRS scores (SRS Rank) will match human cognitive judgment (HCR Rank) scores and there is a positive correlation between the two. Translated into null hypothesis (H_0), it can be expressed as (H_0): SRS Rank scores don't match HCR Rank scores. The alternate hypothesis (H_1) is expressed as: (H_1): SRS Rank scores match HCR Rank scores.

(H_0): SRS Rank scores do not match HCR responses

(H_1): SRS Rank scores matches HCR responses

8.5.1 Pearson Product Moment Correlation

Pearson product-moment correlation coefficient is employed to test the hypothesis above. Literature indicates that this is one of the best tests for dichotomous variables such as SRS Rank and HCR Rank. Table 10 shows a significant positive correlation between the two scores i.e., $r = + 0.919$ or (91.9%). The astericks indicates significant correlation. Significance value (p) for this 2-tailed test is <0.05 thus ($r = 0.919$, $p < 0.05$) supports the alternate hypothesis (H_1) and we therefore reject null hypothesis (H_0).

Table 10 Pearson Product Moment Correlation

		SRS Rank	HCR Rank
SRSRank	Pearson Correlation	1	.919(**)
	Sig. (2-tailed)		.000
	N	30	30
HCR Rank	Pearson Correlation	.919(**)	1
	Sig. (2-tailed)	.000	
	N	30	30

A t-statistic was also measured for to validate the hypothesis results. Given r coefficient = +0.919. The t-statistic resulted in 12.33 with the given degree of freedom of (n-2 = 28) and given $\alpha = 0.01$, the critical value of t was 2.76. Since the t-statistic of 12.33 > 2.76, this clearly rejected the null hypothesis (H_0) and the alternate hypothesis (H_1) was thus accepted.

8.5.2 Nonparametric Correlation

Using Kendall's tau_b and Spearman's rho, both HCR and SRS scores were tested again for nonparametric correlation. Results show a significant correlation given $\alpha = 0.01$ for this 2-tailed test.

Table 11 Nonparametric Correlations

			SRS Rank	HCR Rank
Kendall's tau_b	SRSRank	Correlation Coefficient	1.000	.703(**)
		Sig. (2-tailed)	.	.000
		N	30	30
	HCR Rank	Correlation Coefficient	.703(**)	1.000
		Sig. (2-tailed)	.000	.
		N	30	30
Spearman's rho	SRSRank	Correlation Coefficient	1.000	.843(**)
		Sig. (2-tailed)	.	.000
		N	30	30
	HCR Rank	Correlation Coefficient	.843(**)	1.000
		Sig. (2-tailed)	.000	.
		N	30	30

Given r coefficient = +0.703, the t-statistic for Kendall's tau_b resulted in 5.23 with the given degree of freedom of (n-2 = 28) and $\alpha = 0.01$, the critical value of t was 2.76. Since the t-statistic of 5.23 > 2.76, the null hypothesis (H_0) is rejected and the alternate hypothesis (H_1) is accepted. The t-statistic for Spearman's rho resulted in 8.29 with the

given degree of freedom of $(n-2 = 28)$ and $\alpha = 0.01$, the critical value of t was 2.76. Since the t -statistic of $8.29 > 2.76$, the null hypothesis (H_0) is rejected and the alternate hypothesis (H_1) is accepted. Figures 28 and 29, illustrate that SRS and HCR scores are more closely related compared to HCR and syntactic (SYN) scores. SRS is a hybrid score that combines both semantic as well as syntactic scores. Using the cognitive response (HCR) as a benchmark, we can conclude that SRS correlates closely to HCR but not SYN scores. We can therefore conclude that SRS scores produce better results than SYN scores. Table 12 shows the listing of word pairs and their normalized HCR and SRS scores and are illustrated in figures 28 and 29.

Table 12 Symbols, Word-Pairs and Scores

Symbol	Word Pairs	SYN Scores	SRS Scores	HCR Scores
a	car - automobile	0	2.50	4
b	gem - jewel	3	4.00	1
c	journey - voyage	2.5	3.27	4
d	boy - lad	3.5	3.72	4
e	coast - shore	2.5	2.86	4
f	asylum - madhouse	2	2.92	4
g	magician - wizard	1.5	3.25	4
h	midday - noon	2	3.50	5
i	furnace - stove	2	2.44	3
j	food - fruit	3	2.24	2
k	bird - cock	3	3.14	3
l	bird - crane	2.5	2.15	2
m	tool - implement	1	1.39	2
n	brother - monk	2	2.07	1
o	lad - brother	1.5	1.81	1
p	crane - implement	1	0.90	1
q	journey - car	2	2.04	1
r	monk - oracle	2	1.32	2
s	cemetery - woodland	1.5	0.91	1
t	food - rooster	2.5	1.58	2
u	coast - hill	2.5	1.84	1
v	forest - graveyard	1	0.73	1
w	shore - woodland	1.5	1.02	1
x	monk - slave	2.5	1.88	1
y	coast - forest	3.5	2.16	0
z	lad - wizard	3	1.63	0
aa	chord - smile	2.5	1.44	0
ab	glass - magician	1.5	0.92	0
ac	rooster - voyage	2.5	1.35	0
ad	noon - string	2.5	1.49	1

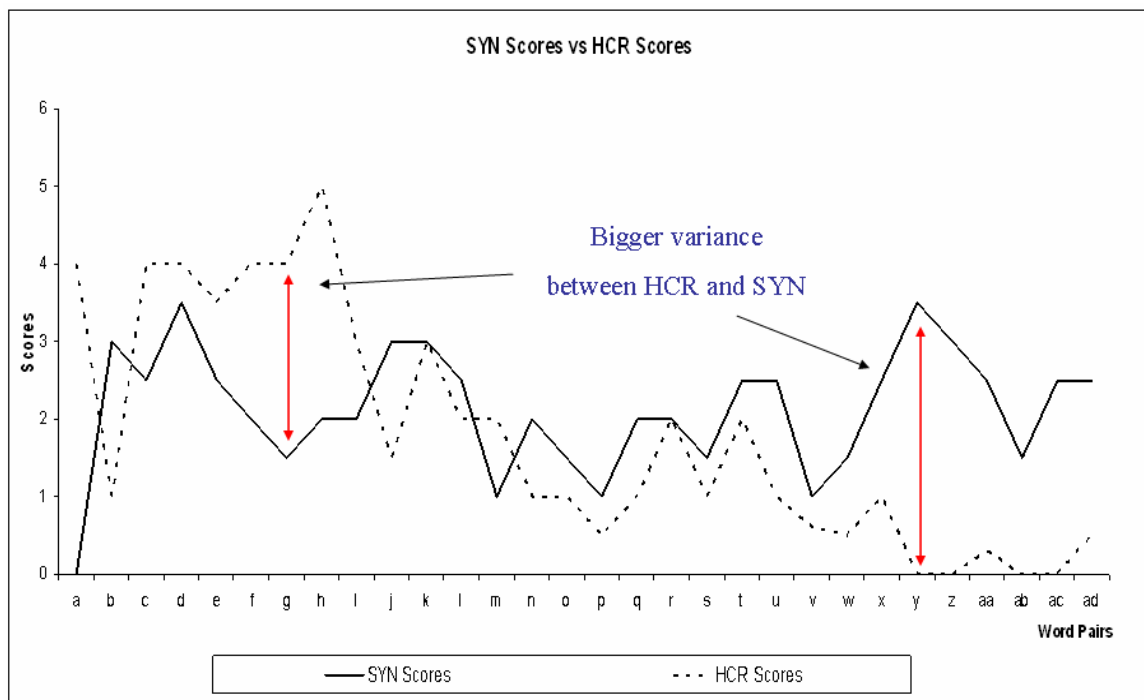


Figure 28 Comparing SYN to HCR Scores

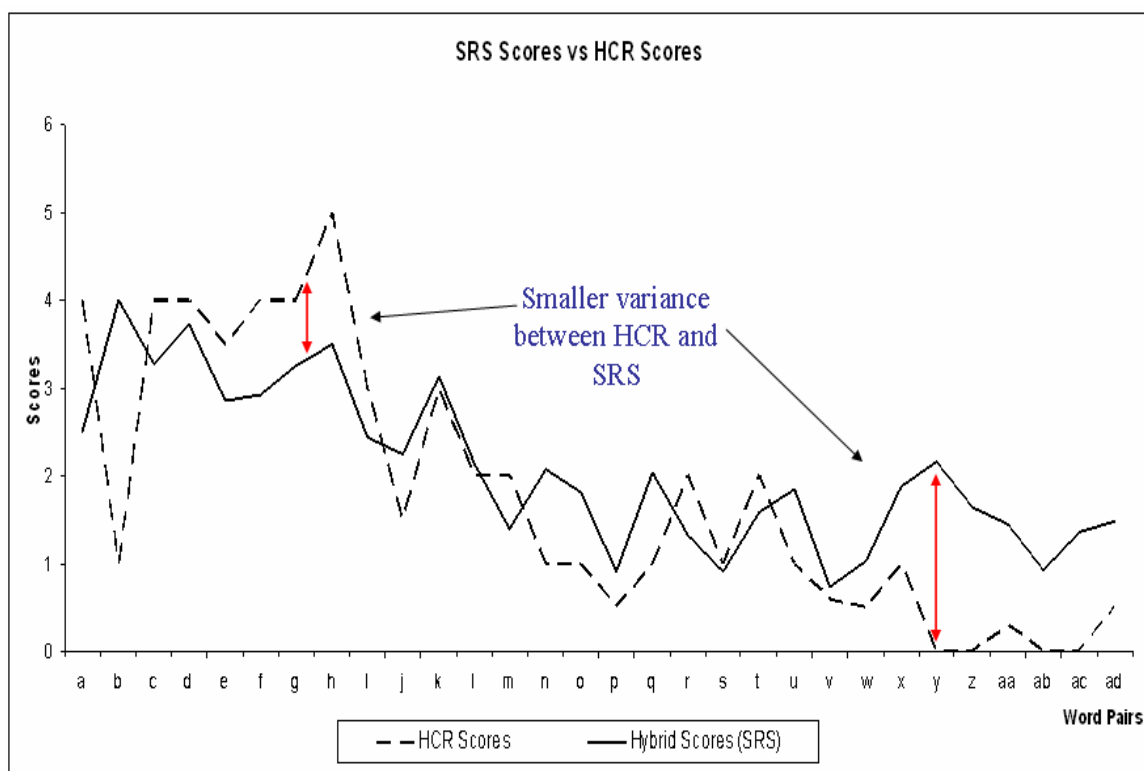


Figure 29 Comparing SRS to HCR Scores

The correlation score of SRS and HCR results in $r = 0.92$ (92%) but SYN and HCR results in a negative score, $r = -0.08$ (-8%). This validates that syntactic matching would not be reliable for ontology mediation. SRS is more reliable because it combines four important measures i.e., Lin (LN), Gloss Vector (GV), WordNet (WN) and LSA. SRS was developed by testing all thirteen linguistic and non-linguistic measures individually and in combinations, the four measures mentioned earlier, resulted in the most precise scores when match with HCR scores. Other mediation techniques only use pure syntactic measures without considering cognitive and linguistic elements for schema matching. This gives SRS a higher leverage compared to other matching algorithms.

8.6 Data Analysis - Survey Part II

The hypothesis here is that matching concepts with semantic and syntactic scores alone does not suffice; we also need context to be determined within matching. It is intuitive to believe that a person's similarity judgment is influenced by context but to prove this, survey part II (Appendix II) was constructed. The hypothesis here is that context influences human judgment regarding similarity, thus HCR has to be context dependant. As such we must give consideration to it in view of the mediation process. Context evaluation provides another layer of tacit understanding of what is being matched. Two words although similar, will convey different meanings when used in different contexts. In the experiment carried out for this thesis, entity concepts were provided and respondents had to rank them from 1 to 11. The lower the rank meant that the concept was more relevant given the context.

The context used for this part of the survey was based on a travel ontology. A dictionary of definitions was also provided to assist respondents to understand the concepts better and to avoid ambiguity. HCR responses were recorded for all the contexts. The results show that similarity judgment changed when contexts changed. This clearly supports the hypothesis that context does influence HCR.

8.6.1 Hypothesis Testing – Survey Part II

Part II is focused on testing the sensitivity of human judgment to changes in context. The hypothesis is that HCR scores are influenced by changes in context, translated into null hypothesis (H_0), it can be expressed as (H_0): HCR responses are not influenced by changes in context. The alternate hypothesis (H_1) is expressed as: (H_1): HCR responses are influenced by context changes.

(H_0): HCR responses are not influenced by context changes

(H_1): HCR responses are influenced by changes in context

8.6.1.1 Hypothesis Test - Part A, B and C

Parts A, B and C of the survey form (Appendix II) are based on finding a “place to stay”. The first part A, does not have any context specified, it reads “which of the following are most similar if you are looking for a place to stay”. All responses were recorded and averaged. In Part B the context “for work, conference, etc” was added to “place to stay”. In Part C the context “only staying 1st class” was added to “place to stay”. For all three parts, scores were averaged and compared to study the discrepancy that resulted in responses.

Table 13 Responses for Part A, B and C

	N	Minimum	Maximum	Mean	Std. Deviation
Motel	3	5.78	6.22	6.02	.23
Inn	3	4.26	5.22	4.84	.51
Hotel	3	1.15	3.56	2.10	1.28
Lodge	3	5.93	7.22	6.44	.69
Hostel	3	7.19	8.74	7.72	.89
Court	3	7.44	8.33	7.85	.45
Tcourt	3	7.15	7.93	7.51	.39
Chalet	3	4.33	6.15	5.25	.91
BoardingH	3	6.48	7.85	6.98	.76
Villa	3	2.67	5.74	4.35	1.56
Apartment	3	4.37	6.41	5.07	1.16
Valid N (listwise)	3				

Table 13, shows the mean, minimum, maximum and standard deviation for all parts. The mean, minimum, maximum and standard deviations for all parts are also listed. Apartment, villa and hotel have higher standard deviation scores, which means that the three contexts have significantly affected similarity judgment of respondents and clearly support the alternate hypothesis (H_1). Table 14, shows the averaged and normalized scores for HCR scores in Part A, B and C of the survey. The respondents clearly have modified their scores when the contexts were changed. Hotel has a score of 3.56 for Part A (“place to stay”), which is the highest similarity score. HCR scores were 4.37 for apartment and 4.69 for villa. In this case the rank order is hotel (rank 1), apartment (rank 2), and villa (rank 3). Hotel scores 1.15 for Part B (“place to stay”-place for meeting and work), which is the highest similarity score. HCR scores were 4.26 for inn and 5.74 for villa. In this case the rank order is hotel (rank 1), inn (rank 2), and villa (rank 3). Hotel score of 1.59 for Part C (“place to stay”-someone keen on staying 1st class), which has the

highest similarity score. HCR scores were 2.67 for villa and 4.44 for apartment. In this case the rank order is hotel (rank 1), villa (rank 2), and apartment (rank 3). Clearly we can say from the above results that all three parts support the alternate hypothesis (H_1).

Table 14 HCR Scores for Part A, B and C

Survey	motel	inn	hotel	lodge	hostel	court	tourist court	chalet	boarding house	villa	apartment
Part A	6.07	5.22	3.56	5.93	7.19	8.33	7.93	5.26	6.59	4.63	4.37
Part B	5.78	4.26	1.15	6.19	7.22	7.78	7.44	6.15	6.48	5.74	6.41
Part C	6.22	5.04	1.59	7.22	8.74	7.44	7.15	4.33	7.85	2.67	4.44

Figure 30 shows all the HCR ranks for Part A, B and C that we have discussed. All 11 categories of the entity concepts and their responses are shown. Respondents have modified their ranks (HCR score) when the context was changed.

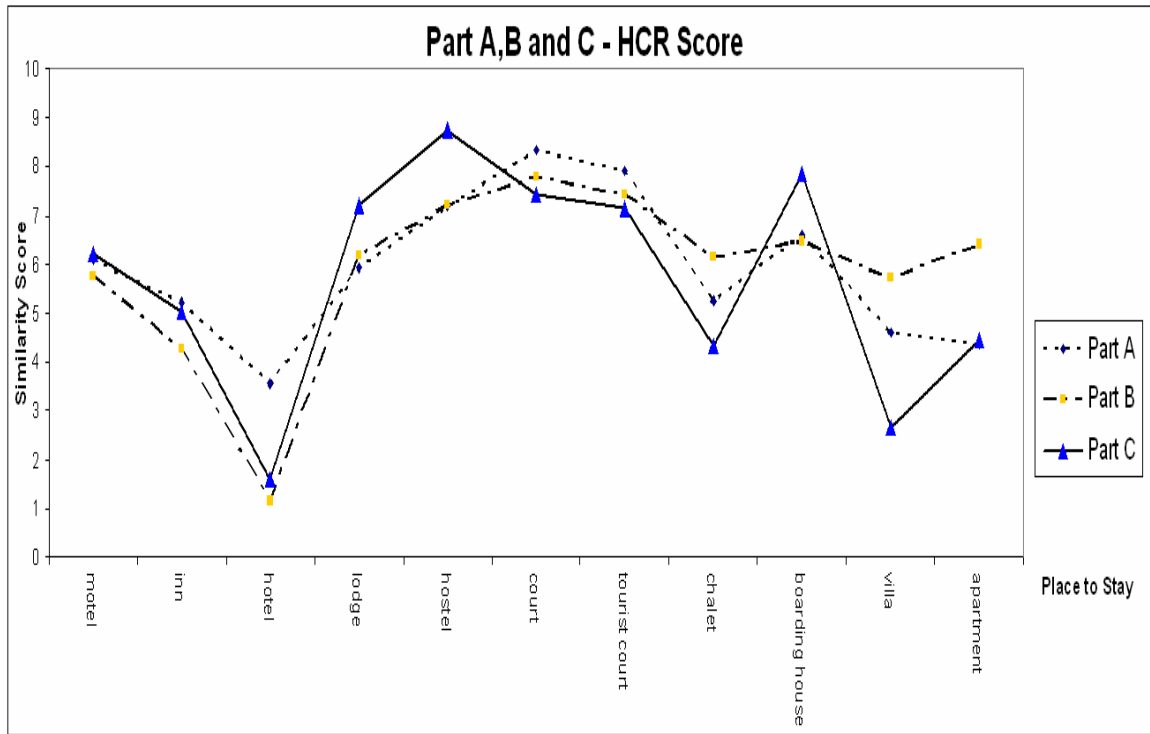


Figure 30 Part A, B and C for HCR Rank Score

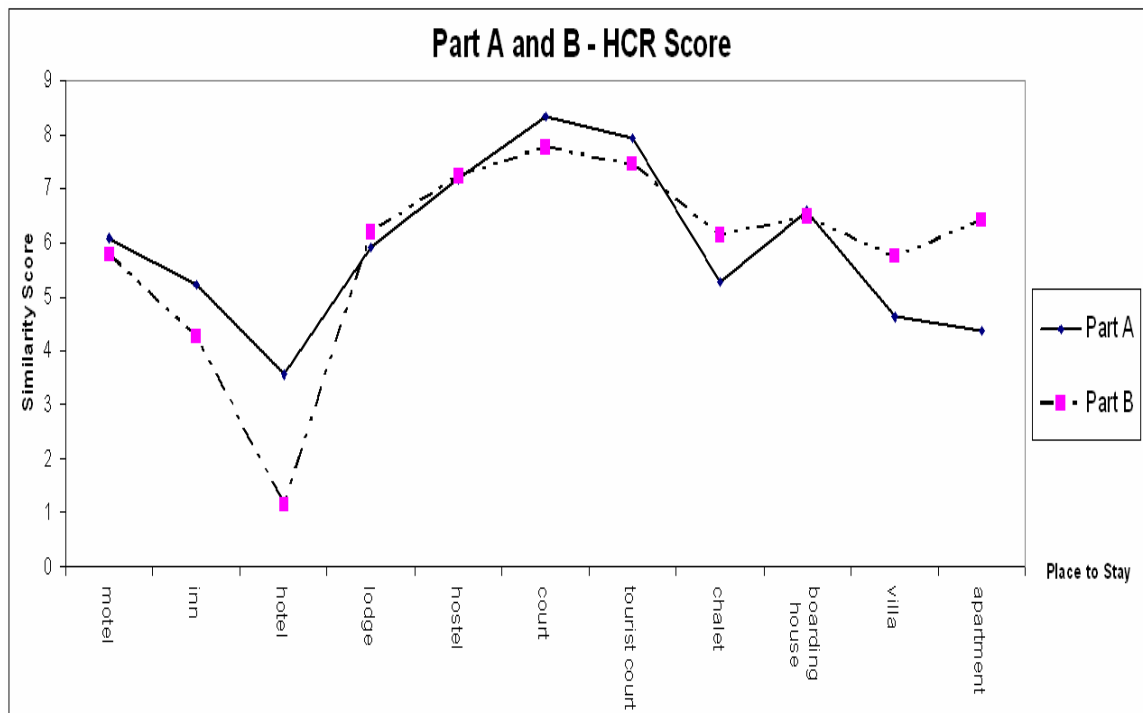


Figure 31 Part A and B HCR Rank Score

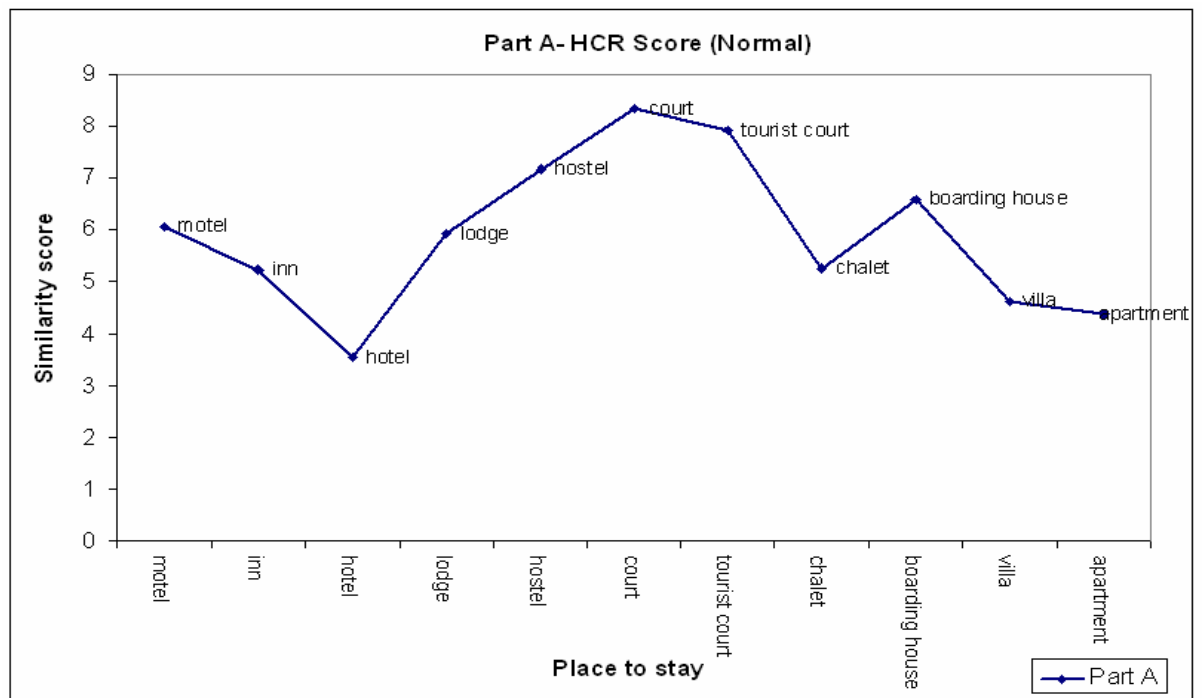


Figure 32 Part A HCR Rank Score

8.6.1.2 Hypothesis Test -Part D

HCR scores for part D shows that the most similar entity concept for someone wanting to “stay 1st class only” is a 5 star ranking. The scores were 1.67 for 5 star, 2.37 luxury and 3.07 excellent. As such the similarity ranking is 1 for 5 star, 2 for luxury and 3 excellent. Intuitively speaking 5 star should be followed by 4 star and 3 star but in this case it was interesting to see that cognitive responses did not really match conventional wisdom. Results below support the null hypothesis (H_0) as well and as such the alternate hypothesis is rejected (H_1).

Table 15 Part D –HCR Rank Score

Excellent	Upscale	3 Star	4 Star	5 Star	Mid-Scale Ltd Service	Mid-Scale Full Service	Good	Luxury	Economy	Budget
3.07	4.89	6.52	4.41	1.67	8.26	6.48	5.89	2.37	8.96	10.07

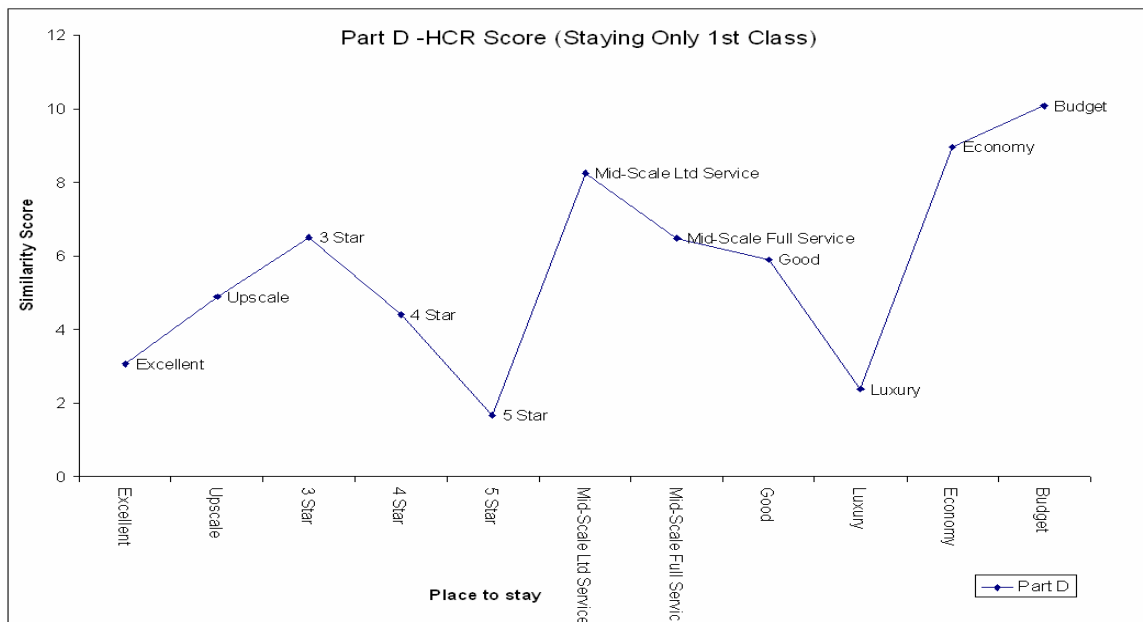


Figure 33 Part D HCR Rank Score (Staying 1st class only)

8.6.1.3 Hypothesis Test -Part E

HCR scores for part E shows that the most similar entity concept for a “place to stay” for someone wanting to “enjoy outdoors activities” is a cabin. The scores were 3.15 for cabin, 3.44 for campground and 3.63 for chalet. Results below once again support the null hypothesis (H_0). The least significant were hostel, court and tourist court.

Table 16 Part E –HCR Rank Score

motel	inn	hotel	lodge	hostel	court	tourist court	chalet	cabin	villa	camp ground
6.56	6.44	6.22	5.41	7.89	7.70	6.85	3.63	3.15	6.48	3.44

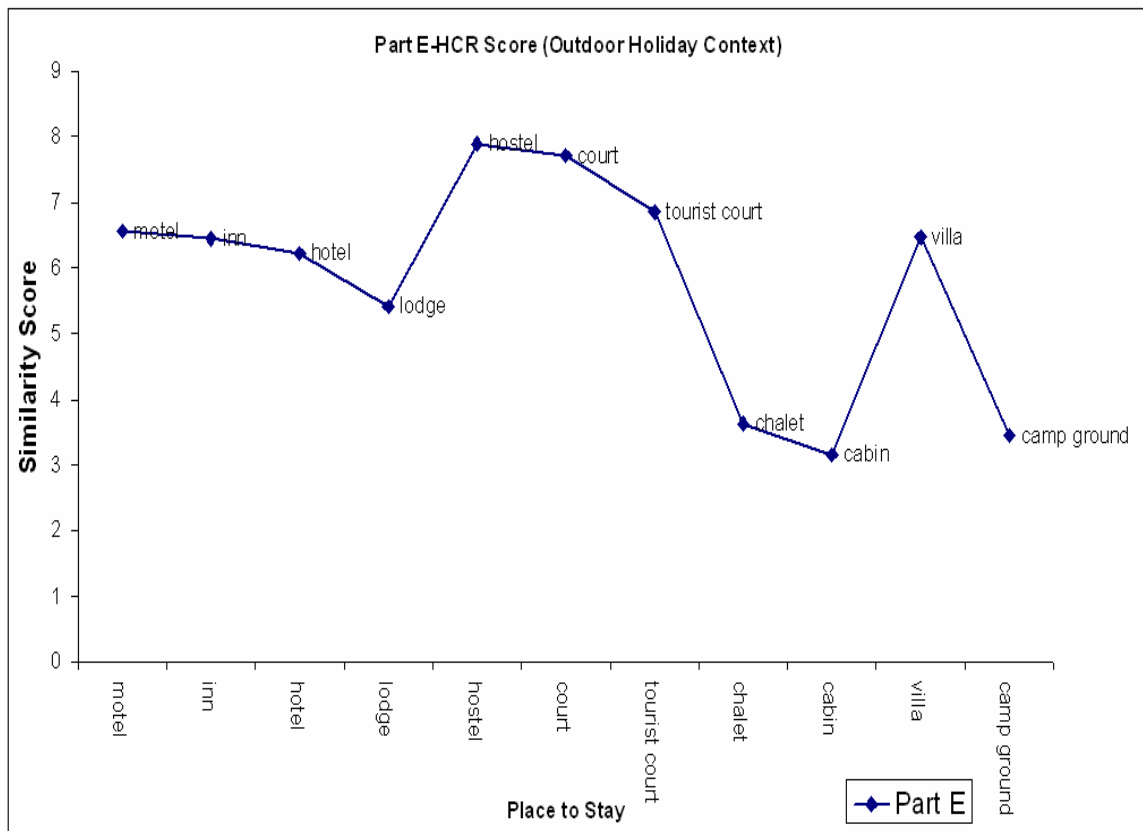


Figure 34 Part E- HCR Rank Score (Outdoor activities)

8.6.1.4 Hypothesis Test -Part D and E

HCR scores for part D (1st class only) and E (outdoor activities) when compared shows that the most similar entity concept for a “place to stay” for part D is hotel (1.67) and for part E is cabin (3.15). Once again we have evidence to support the null hypothesis (H_0).

Table 17 Comparing Part D and E

Part D	Excellent	Upscale	3 Star	4 Star	5 Star	Mid-Scale Ltd Service	Mid-Scale Full Service	Good	Luxury	Economy	Budget
	3.07	4.89	6.52	4.41	1.67	8.26	6.48	5.89	2.37	8.96	10.07
Part E	motel	inn	hotel	lodge	hostel	court	tourist court	chalet	cabin	villa	camp ground
	6.56	6.44	6.22	5.41	7.89	7.70	6.85	3.63	3.15	6.48	3.44

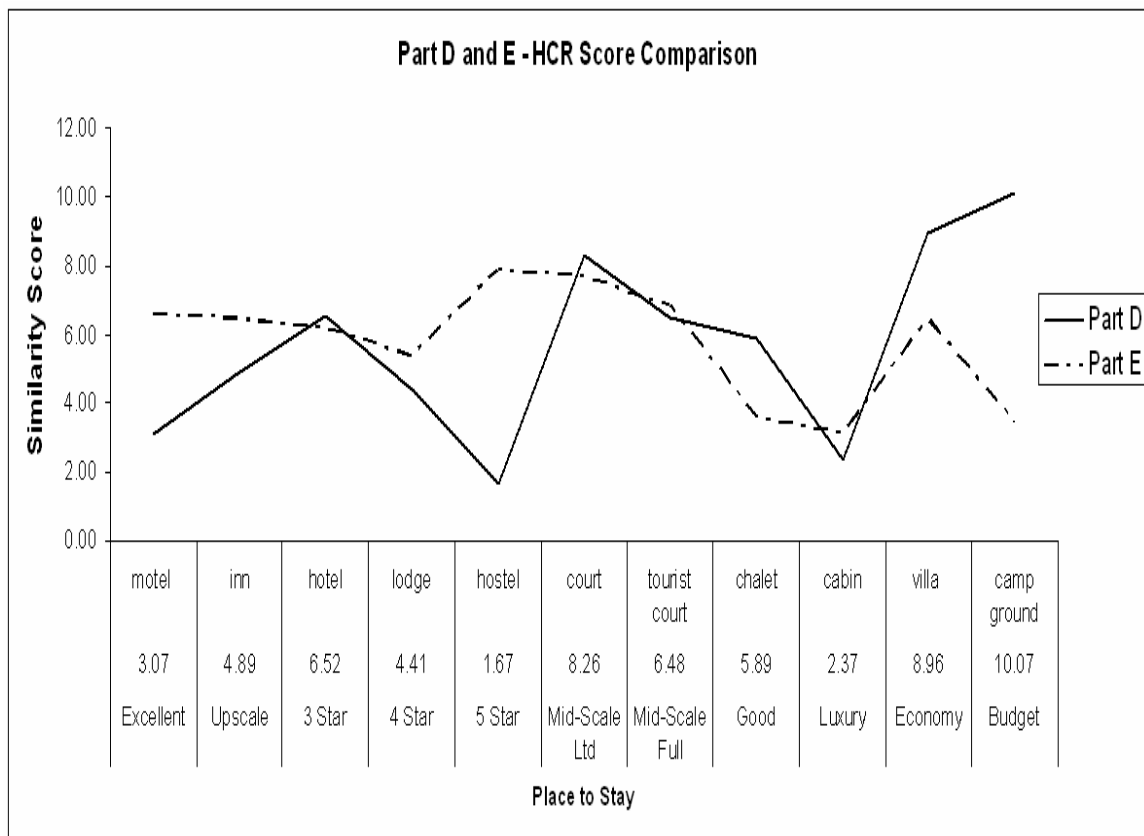


Figure 35 Part D and E – HCR Rank Score

8.7 Major Findings

The hypothesis tested clearly supports the null hypothesis (H_0) for all five parts of the survey form. As such we can reject the alternate hypothesis (H_1) and conclude that context does change the similarity judgment of a human being. As we have seen from the discussion earlier, it is really important for an ontologist to be able to use context evaluation in deciding concept matches. This is because the ontologist uses cognitive judgment in finally deciding which concepts ought to be matched and which of those should be rejected. We should be able to acquire definitions quickly and WordNet is the tool that best provides for this capability. Semi-automated systems can be equipped with an interface to WordNet for this purpose. However, the element of context must be explicitly defined. This is very important in the research of ontology mediation because, a slight modification in context can result in different similarity scores. Semi-automated mediation systems depend on human cognitive responses; as such this finding is significant for the development of future mediation systems.

8.8 Summary

This chapter has addressed the testing and evaluation of semantic and syntactic mediation method, which was, used for SRS rank scores. The hypothesis was that SRS scores would mimic closely to HCR scores. The hypothesis was tested and validated. The measures of linguistic similarity between concepts were evaluated with a two-part survey. The first justifies the close relationship of SRS scores to human cognitive responses or human

judgment. The second part validates that cognitive scores do change significantly given a change in context. Ontology mediation practices today rely only on syntactic matches and do not quite address meaningful equivalence between concepts. Also such systems do not include concept evaluation for matching concept entities. Thus this thesis recommends that SRS measures be implemented as an integral component of semi-automated ontology mediation systems; SRS measures support humans in both content and context mediation.

CHAPTER 9

9. Semantic Bridging via SWRL

SRS when combined with rules such as Semantic Web Rule Language (SWRL) can produce optimum results. This section presents how semantic bridging with SRS can be further extended with rules. SRS provides an ontologist all schemas that are likely to be matched with high reliability and precision. Rules on the other hand are cardinality constraints that can be used for matching data labels, schemas and concepts. Rules can be predefined ahead of time so that frequently appearing schemas can be matched automatically. To match schemata on names for instance we can write a rule that would match `</first_name>`, `</middle_name>` , `</last_name>` with `</full_name>`. If we had schemata for example `</street_name>`, `</city>`, `</state>` and `</zipcode>` to define an address in one domain ontology and defined as just `</address>` in another, a simple rule can be executed to match them on-the-fly.

9.1 Case Study: Achieving Interoperability in E-Government Services via SWRL

Figure 36 illustrates the data heterogeneity problem for inter-organizational services between public agencies. The scenario here is that a customer wants to renew his driver's

license online. He first logs into the DMV (Department of Motor Vehicle) portal and selects the type of service. Then he provides essential information such as full_name, DOB (10-10-1965), DMV customer ID (A33-05-7156) and address (1234 Oakton Circle Rd Arlington VA 22202). These details are passed on to a license renewal inspector where details provided by customer are verified.

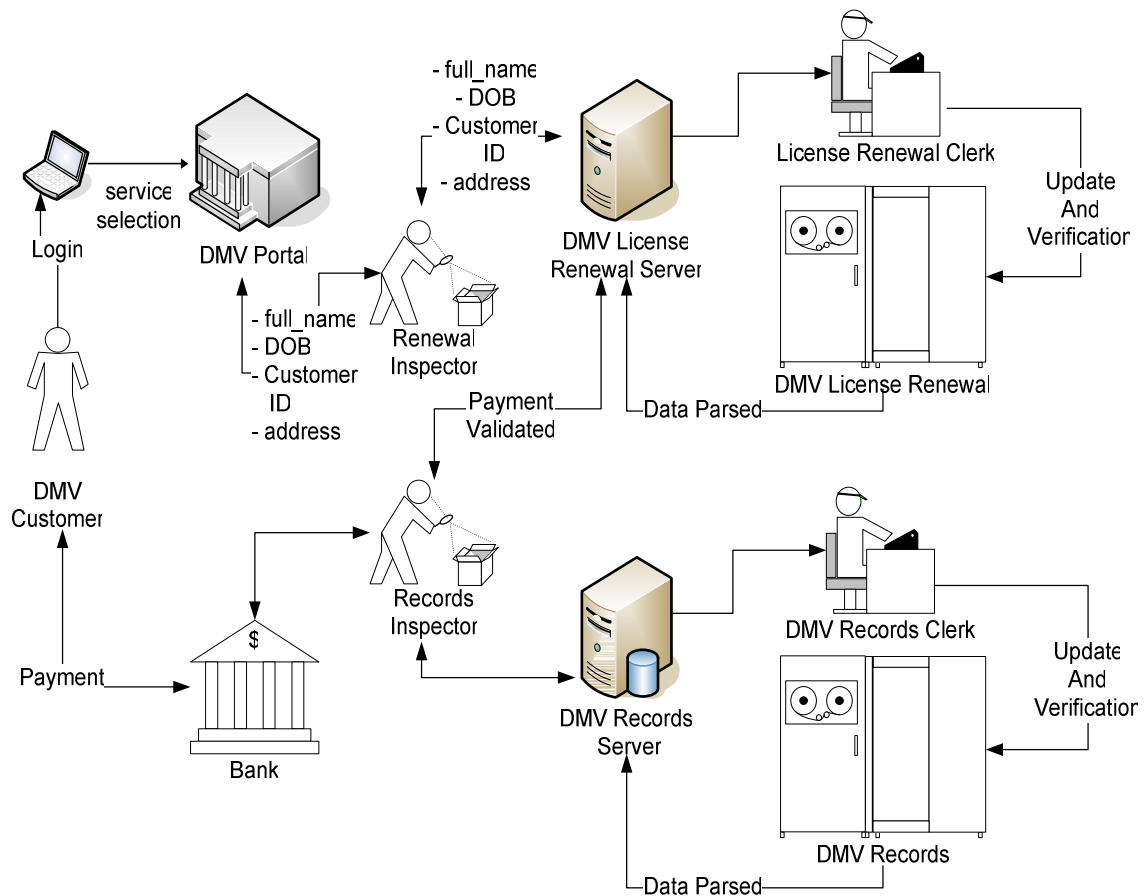


Figure 36 DMV license renewal process

At the same time a mode of payment is selected which is verified by the records inspector before license renewal is performed. Data is then passed on to the DMV License Renewal

server which validates the data and passes on details to the license renewal clerk who updates and verifies renewal data. When payment is validated with updates from the DMV License Renewal server, the records inspector receives this information with updates exchanged from DMV Records server. The bank is then notified for the charges and the customer's account is debited. An option for printing a receipt is also provided. The customer then waits for his renewed license to arrive in the mail. As mentioned earlier, DMV License Renewal and DMV Records have domain specific schema definitions that are different. Since establishing services between them will be an ongoing task rules could provide an automatic solution for creating homogeneity amongst heterogeneous schemas used by them.

9.2 SWRL rules

In view of the reasoning for the Semantic Web, which supports inter-government web services, this thesis proposes an approach where rules can be used to match concepts. Reasoning is an approach when agents in a knowledge system perform tasks by inference. Given the following statement "If X has a son Y, and X has a brother Z, given that X is a male" the agent is able to then infer that Y has an uncle, Z. The agent does not need to be explicitly told about the relationship between Y and Z. As long as uncle is defined earlier, an agent will be able to infer this quickly. SWRL is based on OWL and RuleML (Rule Markup Language). It enables OWL axioms to include Horn-logic that can be used to execute rules.

SWRL rules show the implication between an antecedent (body) and consequent (head). In other words if the antecedent holds true, then the consequent must hold true also. In our example earlier if antecedents “X has a son Y, and X has a brother Z”, X is a male” is all true then the consequent must also hold true which is “Y has an uncle, Z”. With rules in place we can easily automate matching of schemata on-the-fly, which otherwise would be very labor intensive. As such SWRL rules and SRS would be complementary efforts towards semantic bridging.

9.3 Semantic Bridging with SWRL rules

Based on the example in section 9.1, DMV Records maintains first_name, middle_name and last_name however, DMV License Renewal maintains a complex string called full_name. The address in the DMV License Renewal is treated as a complex string and in DMV Records it is divided into street_name, city, state and zipcode.

Figure 37, depicts customer ontology for DMV Records where customer name and address is expressed with greater granularity. It also shows the client ontology for DMV License Renewal where customer name and address is expressed with less granularity. Dotted lines indicate semantic bridging that is done via SWRL rules. We will demonstrate how the rules are written in the next section.

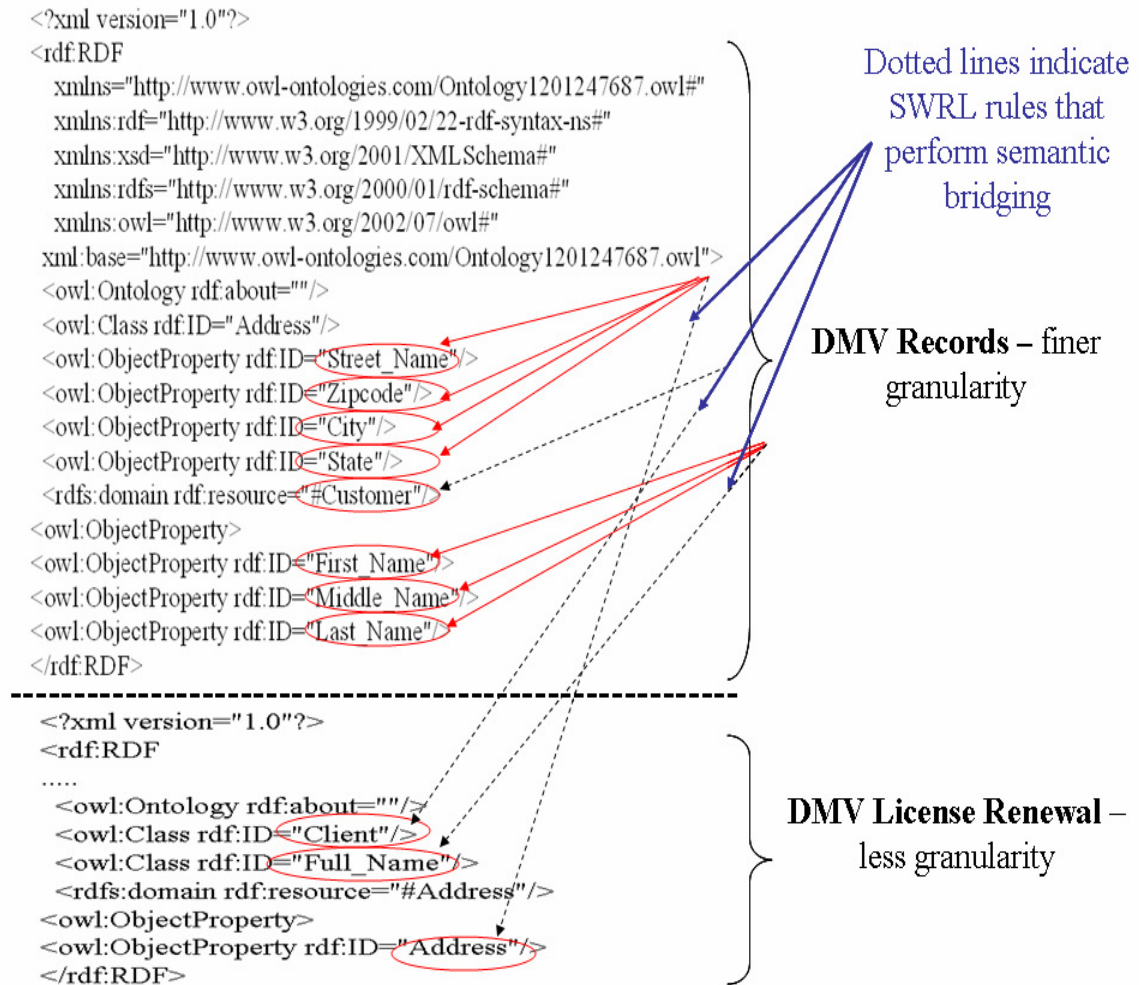


Figure 37 Semantically Bridging DMV Records and DMV Licence Renewal ontologies via SWRL rules

9.4 Writing SWRL rules

Based on the process of semantically bridging the ontologies in figure 37 earlier, in this section we show how SWRL rules are written to accomplish that task. The first rule would be to associate `</Street_Name>`, `</City>`, `</State>` and `</Zipcode>` from DMV Records to `</Address>` in DMV Licence Renewal. The following is how the rule 1 is expressed:

Rule 1: $\text{hasStreet_Name}(\text{?x1}, \text{?x2}) \wedge \text{hasZipcode}(\text{?x2}, \text{?x3}) \rightarrow \text{hasStreetZipAddress}(\text{?x1}, \text{?x3})$

This implies that the (Antecedent ($\text{hasStreet_Name}(\text{I-variable}(\text{x1}) \text{I-variable}(\text{x2}))$, $\text{hasZipcode}(\text{I-variable}(\text{x2}) \text{I-variable}(\text{x3}))$)) and thus the consequent would be ($\text{hasStreetZipAddress}(\text{I-variable}(\text{x1}) \text{I-variable}(\text{x3}))$)).

The following is how rule 2 is expressed:

Rule 2: $\text{hasCity}(\text{?x1}, \text{?x2}) \wedge \text{hasState}(\text{?x2}, \text{?x3}) \rightarrow \text{hasCityStateAddress}(\text{?x1}, \text{?x3})$

This implies that the (Antecedent ($\text{hasCity}(\text{I-variable}(\text{x1}) \text{I-variable}(\text{x2}))$, $\text{hasState}(\text{I-variable}(\text{x2}) \text{I-variable}(\text{x3}))$)) $\text{hasCityStateAddress}(\text{I-variable}(\text{x1}) \text{I-variable}(\text{x3}))$ and thus the consequent would be ($\text{hasRecipient}(\text{I-variable}(\text{x1}) \text{I-variable}(\text{x3}))$)).

Rule 3 combines rule 1 and 2 to determine address and is expressed as:

Rule 3: $\text{hasStreetZipAddress} \wedge \text{hasCityStateAddress} \rightarrow \text{hasAddress}$

This implies that the antecedent $\text{hasStreetZipAddress}$ and $\text{hasCityStateAddress}$ determine the consequent hasAddress . This rule concludes the matching of the addresses of both ontologies. A simple rule (i.e. rule 4) can be executed to map *</customer>* of DMV

Records to *</client>* of DMV License Renewal which is expressed as *<owl:equivalentClass>* or *owl:sameAs*. The last rule would be to associate *</First_Name>*, *</Middle_Name>* and *</Last_Name>* from DMV Records to *</Full_Name>* in DMV Licence Renewal. The last rule, rule 5 is expressed as:

Rule 5: $\text{hasFirst_Name}(\text{?x1}, \text{?x2}) \wedge \text{hasMiddle_Name}(\text{?x2}, \text{?x3}) \wedge \text{hasLast_Name}(\text{?x3}, \text{?x4})$
 $\rightarrow \text{hasFull_Name}(\text{?x1}, \text{?x4})$

This implies (Antecedent (hasFirst_Name (I-variable(x1) I-variable(x2)), hasMiddle_Name (I-variable(x2) I-variable(x3))), hasLast_Name (I-variable(x3) I-variable(x4))) and thus the consequent would be (hasFull_Name (I-variable(x1) I-variable(x4))). These predefined rules will successfully match all the schemas and make resolve future heterogeneity issues. In the next section we describe how SWRL rules are implemented and its benefits.

9.4 Benefit of combining SRS with SWRL rules

Figure 38 extends the expression of the dotted lines shown figure 37. It highlights the semantic bridge that performs data label match for inputs from DMV Records and DMV License Renewal. The interfaces show the parameters that exist and Customer is shown in DMV Records on the left and Client is shown on the right for DMV License Renewal.

All the highlighted items in bold are schemas that are actually being mapped. All the five rules that have been triggered are summarized at the bottom of figure 38.

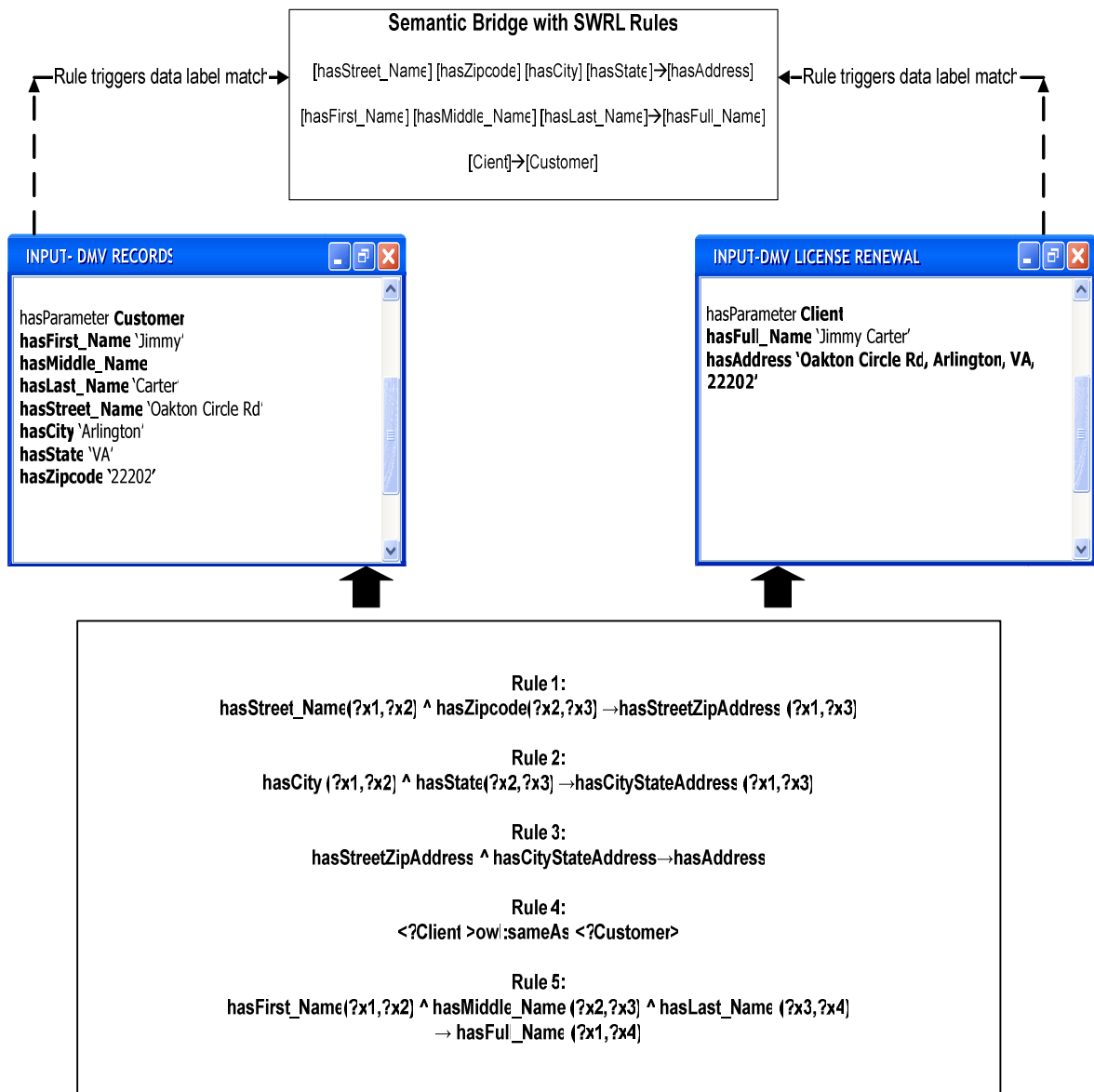


Figure 38 Implementing SWRL Rules

The main benefits of SWRL rules are:

- It overcomes the limitations of standards and ICT technology such as EDI, XML, and RosettaNet which do not allow polymorphism and inheritance. For example what has been demonstrated in rule 5.
- Has greater reliability and precision when used together with SRS.
- Allows not only binary (1:1) but also multiple mappings (1: m, m: 1, m: n).
- Once the definition are agreed among public agencies like those done in rules 1 through 5, then future data exchange is automatically triggered as it would be predefined allowing data exchanges and knowledge inheritance to happen on-the-fly.
- Improves overall scalability and efficiency in the context of Web Service compositions among public agencies.

9.5 Summary

In summary, this section presents as to how data interoperability in public services can be optimized with SWRL rules. The contention here is that all public agencies will need to share definitions irrespective of their domain specifications in order to seamlessly exchange services. Due to the fact that there cannot be a single perfect ontology for all agencies to comply with and to obtain richer definitions of data, inter-ontology mappings need to be done. SRS combined with SWRL provides excellent support to achieving this goal.

CHAPTER 10

10. Conclusion and Major Results

An ontology is defined as a formal and explicit specification of a domain of knowledge. Although ontologies are meant to resolve data heterogeneity problems, ontologies *per se* do not solve heterogeneity problems. This is because of the nature of how ontologies are being created in the industry today. A vast number of ontologies are being developed today by different organizations and individuals. Although some of these ontologies are being developed for different domains, they have a lot of overlapping data between them. Ontologists today not only want to develop their own ontologies but at the same time leverage concept definitions from multiple ontologies to solve a problem. The underlying problem here is that many of these ontologies cannot be used together as they do not share the same structure or naming conventions. Therefore it is necessary to find a way to enable interoperability between these ontologies via mapping.

Currently ontology mapping techniques are heavily based on manual mapping where a human expert has to understand the implicit semantics of the chosen ontologies to match their concepts. This causes human error and delay. As such the reliability and precision of

this approach becomes questionable. Although semi-automated systems are becoming more popular, they are mostly based on syntactic matching algorithms that do not include semantics in matching ontological concepts and instances. This thesis introduced a more reliable approach which improved the precision and relevance of concept matching. A hybrid similarity score (i.e. SRS) was developed based on a combination of five linguistic, cognitive and non-linguistic measures. Based on empirical tests, pure syntactic (i.e. SYN) matching algorithms did not perform as well as the hybrid measure. In terms of precision SRS gave a 40% score and SYN only had 16.67% score. SRS also had a higher relevance score of 96.67% compared to SYN which had only 73.33%. Also SRS had a positive correlation score of 91.9% with a t-statistic score of 12.33 which was higher than the critical t value of 2.76 at a significance level of 0.01.

Another important gap that this thesis fulfilled is the need for a framework and methodology for ontology mapping. As such a detailed process methodology was provided in chapter 6 and a Semantic Mediation Architecture (SMA) was presented in chapter 7. Identifying concept equivalence was an important goal in this architecture. The “semantic bridge” in this architecture is designed to handle syntactic and semantic matching. It can also handle issues related to conflict analysis and conflict resolution. In chapter 9 the implementation of the Semantic Web Rule Language (SWRL) was introduced for on-the-fly mappings of concepts that need to be used over and over again for fulfilling a web service based on the DMV license renewal process.

Chapter 9 illustrates how the predefined rules and concepts allow data to be concatenated allowing on-the-fly mappings to be executed efficiently.

10.1 Thesis Summary

Seventeen semi-automated mapping methods were investigated in this research and they are: Comparative Framework, MAFRA, OISs, OntoMapO, IFF Framework, FCA Merge, If Map, Prompt/SMART/Prompt-Diff, Chimeara/GLUE/Caiman, Onion, COMA (Complex Mapping), S-Match, SF, OLA (OWL Lite Algorithm), Cyc, XML mapper and NOM (Native Ontology Mapping). The results showed that most systems rely heavily on syntactic matches and do not use semantics to establish concept relations. ONION for example used substring, prefix and suffix matches without intended semantics. S-Match for instance used semantic relations over a pair of nodes in a graph. This shows that graph theory is used to determine distances. Chimeara, GLUE and Caiman used syntactic match, machine learning and probability models. Prompt/SMART/Prompt-Diff used syntactic and linguistic matches.

IFF Framework was a theoretical model, which uses knowledge scenario but does not have a clear process methodology. OISs uses a relational database approach achieved via queries and does not cater to unstructured data. FCAMerge had utilized lexical analysis and NLP techniques. IFMap was based on rules, prolog clauses and libraries. In conclusion there was not a single method that can be said to be the best for ontology mediation. Ontology mediation is not a trivial task and is laborious.

This thesis provided an end-to-end approach considering all factors for ontology mediation. In the Semantic Web real-time agent ontology systems would carry out syntactic and semantic matches on the fly to mediate ontologies.

There are several methods and techniques for ontology mediation. For example COMA, S-Match, SF and OLA use string-based, language-based, constraint-based, word-sense, linguistic resources, taxonomy, alignment and taxonomy approaches. XMLMapper and Cyc handle most semi-structured data and don't really scale for the Semantic Web. This thesis recognizes the importance of ongoing research and shows that most of the techniques tend to address mediation in a somewhat disjoint manner. A new process methodology has been proposed to handle these requirements.

Empirical tests have also been conducted to validate the concept. SPDM (Security Policy Domain Model) was introduced for proof-of-concept. This shows that this proposed approach is applicable for reconciling security policy ontologies in the Semantic Web. Also, most mapping practices largely focus on binary mappings (1:1). The proposed approach in this thesis includes complex mappings as well (i.e., 1:n, n: 1, m: n).

10.2 Future Research Direction

This thesis highlights the need for context evaluation for similarity assessment. A small five-part survey was conducted and some hypothesis tests were conducted. The empirical

analysis shows that HCR scores are influenced by context changes. Empirical tests validate that context is an important element for concept matching. Although the SRS scores proposed included syntactic and semantic measures, it does not include other context measures besides “is-a” relations yet. This is because WordNet does not support other contextual information yet. As such this area would be a possible future research direction.

APPENDIX I: Survey Part 1

PART 1

This survey is being conducted to study how people judge semantic relatedness of word pairs. There are 30 word pairs in this survey and your response would be to rank (from 0 to 5), 0 for unrelated word pairs and 5 for highly related word pairs based on your similarity judgment. Please do not assign the same rank twice for the same category.

The whole survey should take less than 10 minutes.

The completion of this survey is absolutely voluntary and you may choose to skip any parts you wish to. Your responses will remain anonymous, please do not write your name anywhere on this form.

General Information

Age: ____ years

Gender: ____ Female ____ Male

Place of birth: _____

Place of residence: _____ (e.g. KL)

Native (first) language spoken: _____

Word Pairs	Rank	Word Pairs	Rank
car - automobile		crane - implement	
gem - jewel		journey - car	
journey - voyage		monk - oracle	
boy - lad		cemetery - woodland	
coast - shore		food - rooster	
asylum - madhouse		coast - hill	
magician - wizard		forest - graveyard	
midday - noon		shore - woodland	
furnace - stove		monk - slave	
food - fruit		coast - forest	
bird - cock		lad - wizard	
bird - crane		chord - smile	
tool - implement		glass - magician	
brother - monk		rooster - voyage	
lad - brother		noon - string	

APPENDIX II: Survey Part 2

PART 2

This survey is being conducted to study how people judge similar words or concepts and how their views change under different contexts. It has 5 parts (i.e. A, B, C, D and E). Each part describes a situation or context with a set of entity sets (i.e. words). The list of entity sets and their definitions are also provided. Your response would be to rank (from 1 to 11) the entity sets, based your similarity judgment. Please do not assign the same rank twice.

The whole survey should take less than 20 minutes.

The completion of this survey is absolutely voluntary and you may choose to skip any parts you wish to. Your responses will remain anonymous, please do not write your name anywhere on this form.

Please read the description at the top of each page and use the definitions given. Fill in your evaluation, and then turn the page.

Please do not go back.

General Information

Age: ____ years

Gender: ____ Female ____ Male

Place of birth: _____ Place of residence: _____ (e.g. KL)

Native (first) language spoken: _____

Definitions

1. **Motel:** motor hotel - a room primarily used for sleeping
2. **Inn:** a hotel for travelers
3. **Hotel:** a building where travelers pay for lodging, meals and other services
4. **Lodge:** a temporary place to stay
5. **Hostel:** inexpensive supervised lodging
6. **Court:** a hotel for motorists which provides direct access from room to car park
7. **Tourist court:** hotel for motorist tourists
8. **Chalet:** Swiss house which is used for stay
9. **Boarding house:** private house that provides meals and accommodation
10. **Villa:** (British) detached or semi-detached suburban house
11. **Apartment:** a suite of rooms usually on one floor of an apartment house
12. **Camp ground:** a site where holiday makers can pitch tents
13. **Cabin:** small house built from wood
14. **Luxury:** high quality usually with full service
15. **Budget:** very basic with low quality usually with no added service
16. **Economy:** average quality to meet basic needs and does not provide full service

17. **Good:** average quality of service to meet a traveler's basic needs
18. **Upscale:** high quality usually with full service
19. **Mid-Scale Full Service:** good quality with full service
20. **Mid-Scale Limited Service:** good quality with limited service
21. **5 star:** rating given for superior quality of service and very upscale. These **luxury** properties are members of an elite group of hotels that exhibit an exceptionally high degree of service and hospitality. These properties display an original design, elegant room decor, exceptional dining, and meticulous grounds. The flawless execution of guest services is the staff's prevailing concern.
22. **4 star:** rating given for good quality of service and mid-scale. These **superior** properties distinguish themselves with a high level of service and hospitality, as well as a wide variety of amenities and upscale facilities. A well-integrated design, stylized room decor, excellent restaurant facilities, and landscaped grounds are all present. The comfort and convenience of the guest is the staff's prevailing concern.
23. **3 star:** rating given for average quality of service and limited services. These properties offer a higher level of service with additional amenities, features, and facilities. The property grounds, decor, and quality of furnishings are a noticeable upgrade in terms of style and class. Most properties in this category feature restaurants serving breakfast, lunch, and dinner. Room service availability may vary. Full service is often provided.

Note: Full Service refers to: Valet parking, swimming pool and fitness centers.
Star definitions adapted from Travelocity

Part A

How similar is a “place to stay” to the following places (1: the most similar)?

1. [] Motel
2. [] Inn
3. [] Hotel
4. [] Lodge
5. [] Hostel
6. [] Court
7. [] Tourist court
8. [] Chalet
9. [] Boarding house
10. [] Villa
11. [] Apartment

Part B

How similar is a “place to stay” to the following places if you were keen on searching for the perfect place for meeting, work, seminar or conference (1: the most similar)?

1. [] Motel
2. [] Inn
3. [] Hotel
4. [] Lodge
5. [] Hostel
6. [] Court
7. [] Tourist court
8. [] Chalet
9. [] Boarding house
10. [] Villa
11. [] Apartment

Part C

How similar is the “a place to stay” to the following places if you are only keen on only staying first class (1: the most similar)?

1. [] Motel
2. [] Inn
3. [] Hotel
4. [] Lodge
5. [] Hostel
6. [] Court
7. [] Tourist court
8. [] Chalet
9. [] Boarding house
10. [] Villa
11. [] Apartment

Part D

How similar is the “ranking of a place to stay” to the following if you are only keen on staying only first class (1: the most similar)?

1. [] Excellent
2. [] Upscale
3. [] 3 star
4. [] 4 star
5. [] 5 star
6. [] Mid-Scale limited service
7. [] Mid-Scale full service
8. [] Good
9. [] Luxury
10. [] Economy
11. [] Budget

Part E

How similar is a “place to stay” to the following places if you are keen to enjoy outdoor holiday activities like snorkeling, diving, mountain climbing, etc (1: the most similar)?

1. [] Motel

2. [] Inn

3. [] Hotel

4. [] Lodge

5. [] Hostel

6. [] Court

7. [] Tourist court

8. [] Chalet

9. [] Cabin

10. [] Villa

11. [] Camp ground

Thanks for your cooperation.

APPENDIX III: OWL results after reasoning and mapping

A. OWL for RO (after RacerPro reasoning)

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Authorization"/>
  <owl:DatatypeProperty rdf:ID="password">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Authorization"/>
  </owl:DatatypeProperty>
  <owl:FunctionalProperty rdf:ID="user_id">
    <rdfs:domain rdf:resource="#Authorization"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:ID="token_id">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Authorization"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:ID="X.509_token">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Authorization"/>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">X.509 token</rdfs:label>
  </owl:FunctionalProperty>
</rdf:RDF>
<!-- Created with Protege (with OWL Plugin 2.2, Build 307) http://protege.stanford.edu
-->
```

Mlog Output for RO

-----START-----

Total number of generated suggestions: 0

Number of generated suggestions that were followed by the user: 0

Total number of conflicts detected: 0

Number of conflict solutions used: 0

Total number of KB operations: -3

Note:

Appendix III (A) above shows the OWL results after RacerPro had reasoned it. All the attributes and classes for RO were checked using description logic. The log shows that there were no problems with RO's ontology as such no conflicts were detected.

OWL for VO (after RacerPro reasoning)

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Authentication"/>
  <owl:DatatypeProperty rdf:ID="X.509_certificate">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >X.509 certificate</rdfs:label>
    <rdfs:domain rdf:resource="#Authentication"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="password">
    <rdfs:domain rdf:resource="#Authentication"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  </owl:DatatypeProperty>
  <owl:FunctionalProperty rdf:ID="identifier">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Authentication"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 2.2, Build 307) http://protege.stanford.edu
-->
Mlog Output for VO
```

-----START-----

Total number of generated suggestions: 0

Number of generated suggestions that were followed by the user: 0
Total number of conflicts detected: 0
Number of conflict solutions used: 0
Total number of KB operations: -3
Total number of generated suggestions: 0
Number of generated suggestions that were followed by the user: 0
Total number of conflicts detected: 0
Number of conflict solutions used: 0
Total number of KB operations: -3

Note:

Appendix III (B) above shows the OWL results after RacerPro had reasoned it. All the attributes and classes for VO were checked using description logic. The log shows that there were no problems with RO's ontology as such no were conflicts detected.

C. Mapping is complete for scenario A.

Note:

Appendix III (C) above shows the mapping results for Scenario A. We did this after the reasoning process was complete. Renamed attributes and operations are mentioned in column 3 and 4. The delete operation shows that a slot is being eliminated in the process. Map-level shows mapped operation and password in this case was not renamed, as it was isomorphic.

APPENDIX IV: Levenshtein Distance, in Three Flavors
By Michael Gilleland, Merriam Park Software

Java Implementation

```
public class Distance {

    /*******
    // Get minimum of three values
    /*******

    private int Minimum (int a, int b, int c) {
    int mi;

        mi = a;
        if (b < mi) {
            mi = b;
        }
        if (c < mi) {
            mi = c;
        }
        return mi;

    }

    /*******
    // Compute Levenshtein distance
    /*******

    public int LD (String s, String t) {
    int d[][]; // matrix
    int n; // length of s
    int m; // length of t
    int i; // iterates through s
    int j; // iterates through t
    char s_i; // ith character of s
    char t_j; // jth character of t
    int cost; // cost

        // Step 1

        n = s.length ();
        m = t.length ();
        if (n == 0) {
```

```

    return m;
}
if (m == 0) {
    return n;
}
d = new int[n+1][m+1];

// Step 2

for (i = 0; i <= n; i++) {
    d[i][0] = i;
}
for (j = 0; j <= m; j++) {
    d[0][j] = j;
}

// Step 3

for (i = 1; i <= n; i++) {

    s_i = s.charAt (i - 1);

    // Step 4

    for (j = 1; j <= m; j++) {

        t_j = t.charAt (j - 1);

        // Step 5

        if (s_i == t_j) {
            cost = 0;
        }
        else {
            cost = 1;
        }

        // Step 6

        d[i][j] = Minimum (d[i-1][j]+1, d[i][j-1]+1, d[i-1][j-1] + cost);

    }
}

```

```
    // Step 7
    return d[n][m];
}
}
```

C++ Implementation

In C++, the size of an array must be a constant, and this code fragment causes an error at compile time:

```
int sz = 5;
int arr[sz];
```

This limitation makes the following C++ code slightly more complicated than it would be if the matrix could simply be declared as a two-dimensional array, with a size determined at run-time. In C++ it's more idiomatic to use the System Template Library's vector class, as Anders Sewerin Johansen has done in an alternative C++ implementation.

Here is the **definition** of the class (distance.h):

```
class Distance
{
public:
    int LD (char const *s, char const *t);
private:
    int Minimum (int a, int b, int c);
    int *GetCellPointer (int *pOrigin, int col, int row, int nCols);
    int GetAt (int *pOrigin, int col, int row, int nCols);
    void PutAt (int *pOrigin, int col, int row, int nCols, int x);
};
```

Here is the **implementation** of the class (distance.cpp):

```
#include "distance.h"
#include <string.h>
#include <malloc.h>

//*****
// Get minimum of three values
//*****

int Distance::Minimum (int a, int b, int c)
{
    int mi;

    mi = a;
    if (b < mi) {
        mi = b;
    }
    if (c < mi) {
        mi = c;
    }
}
```

```

    return mi;

}

//*****
// Get a pointer to the specified cell of the matrix
//*****

int *Distance::GetCellPointer (int *pOrigin, int col, int row, int nCols)
{
    return pOrigin + col + (row * (nCols + 1));
}

//*****
// Get the contents of the specified cell in the matrix
//*****

int Distance::GetAt (int *pOrigin, int col, int row, int nCols)
{
    int *pCell;

    pCell = GetCellPointer (pOrigin, col, row, nCols);
    return *pCell;

}

//*****
// Fill the specified cell in the matrix with the value x
//*****

void Distance::PutAt (int *pOrigin, int col, int row, int nCols, int x)
{
    int *pCell;

    pCell = GetCellPointer (pOrigin, col, row, nCols);
    *pCell = x;

}

//*****
// Compute Levenshtein distance
//*****

int Distance::LD (char const *s, char const *t)

```

```

{
int *d; // pointer to matrix
int n; // length of s
int m; // length of t
int i; // iterates through s
int j; // iterates through t
char s_i; // ith character of s
char t_j; // jth character of t
int cost; // cost
int result; // result
int cell; // contents of target cell
int above; // contents of cell immediately above
int left; // contents of cell immediately to left
int diag; // contents of cell immediately above and to left
int sz; // number of cells in matrix

```

// Step 1

```

n = strlen (s);
m = strlen (t);
if (n == 0) {
    return m;
}
if (m == 0) {
    return n;
}
sz = (n+1) * (m+1) * sizeof (int);
d = (int *) malloc (sz);

```

// Step 2

```

for (i = 0; i <= n; i++) {
    PutAt (d, i, 0, n, i);
}

for (j = 0; j <= m; j++) {
    PutAt (d, 0, j, n, j);
}

```

// Step 3

```

for (i = 1; i <= n; i++) {

    s_i = s[i-1];

```



```

// Step 4

for (j = 1; j <= m; j++) {

    t_j = t[j-1];

    // Step 5

    if (s_i == t_j) {
        cost = 0;
    }
    else {
        cost = 1;
    }

    // Step 6

    above = GetAt (d,i-1,j, n);
    left = GetAt (d,i, j-1, n);
    diag = GetAt (d, i-1,j-1, n);
    cell = Minimum (above + 1, left + 1, diag + cost);
    PutAt (d, i, j, n, cell);
}
}

// Step 7

result = GetAt (d, n, m, n);
free (d);
return result;
}

```

Visual Basic Implementation

```
*****
*** Get minimum of three values
*****
Private Function Minimum(ByVal a As Integer, _
                        ByVal b As Integer, _
                        ByVal c As Integer) As Integer
Dim mi As Integer
mi = a
If b < mi Then
mi = b
End If
If c < mi Then
mi = c
End If

Minimum = mi

End Function

*****
*** Compute Levenshtein Distance
*****
Public Function LD(ByVal s As String, ByVal t As String) As Integer
Dim d() As Integer ' matrix
Dim m As Integer ' length of t
Dim n As Integer ' length of s
Dim i As Integer ' iterates through s
Dim j As Integer ' iterates through t
Dim s_i As String ' ith character of s
Dim t_j As String ' jth character of t
Dim cost As Integer ' cost

' Step 1

n = Len(s)
m = Len(t)
If n = 0 Then
LD = m
Exit Function
End If
If m = 0 Then
LD = n
```

```

Exit Function
End If
ReDim d(0 To n, 0 To m) As Integer

' Step 2

For i = 0 To n
    d(i, 0) = i
Next i

For j = 0 To m
    d(0, j) = j
Next j

' Step 3

For i = 1 To n

    s_i = Mid$(s, i, 1)

' Step 4

    For j = 1 To m

        t_j = Mid$(t, j, 1)

' Step 5

        If s_i = t_j Then
            cost = 0
        Else
            cost = 1
        End If

' Step 6

        d(i, j) = Minimum(d(i - 1, j) + 1, d(i, j - 1) + 1, d(i - 1, j - 1) + cost)

    Next j

Next i

' Step 7

```

LD = d(n, m)
Erase d

End Function

APPENDIX V: Levenshtein Distance Algorithm
By Michael Gilleland, Merriam Park Software

Algorithm and Steps

Step	Description
1	Set n to be the length of s. Set m to be the length of t. If n = 0, return m and exit. If m = 0, return n and exit. Construct a matrix containing 0..m rows and 0..n columns.
2	Initialize the first row to 0..n. Initialize the first column to 0..m.
3	Examine each character of s (i from 1 to n).
4	Examine each character of t (j from 1 to m).
5	If s[i] equals t[j], the cost is 0. If s[i] doesn't equal t[j], the cost is 1.
6	Set cell d[i,j] of the matrix equal to the minimum of: a. The cell immediately above plus 1: d[i-1,j] + 1. b. The cell immediately to the left plus 1: d[i,j-1] + 1. c. The cell diagonally above and to the left plus the cost: d[i-1,j-1] + cost.
7	After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell d[n,m].

APPENDIX VI: Matching Algorithm and SWRL rules

Algorithm – Step 1 and 2:

LOAD source and target ontologies loaded
GIVEN SO and TO
EXECUTE E parameters 1, 2, and 3
FOR all candidate pairs ($C_i:C_j$) and ($c_i:c_i$)
IF candidate pairs are equivalent, 1 or 2 and 3
THEN Go to step 3, 4 and 5
ELSE Go to step 1

SWRL Rule – Step 1 and 2:

Rule 1- $\text{hasSem}(?x1,?x2) \wedge \text{hasSimSlots}(?x2,?x3) \rightarrow \text{hasEquivalence}(?x1,?x3)$

Implies (Antecedent ($\text{hasSem}(\text{I-variable}(x1) \text{ I-variable}(x2))$
 $\text{hasSimSlots}(\text{I-variable}(x2) \text{ I-variable}(x3)))$
Consequent ($\text{hasEquivalence}(\text{I-variable}(x1) \text{ I-variable}(x3)))$)

Rule 1

```
<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="hasSem">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x2</ruleml:var>
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom swrlx:property="hasSimSlots">
      <ruleml:var>x2</ruleml:var>
      <ruleml:var>x3</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="hasEquivalence">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x3</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_head>
</ruleml:imp>
```

Note: imp refers to rule axioms. It is read as a logical implication between the antecedent (_body) and consequent (_head)

Rule 2- $\text{hasSem}(?x1,?x2) \wedge \text{hasSyn}(?x2,?x3) \rightarrow \text{hasEquivalence}(?x1,?x3)$

Implies (Antecedent ($\text{hasSem}(\text{I-variable}(x1) \text{ I-variable}(x2))$
 $\text{hasSyn}(\text{I-variable}(x2) \text{ I-variable}(x3)))$
Consequent ($\text{hasEquivalence}(\text{I-variable}(x1) \text{ I-variable}(x3)))$)

Rule 2

```
<ruleml:imp>
  <ruleml:_body>
```

```

    <swrlx:individualPropertyAtom   swrlx:property="hasSem">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x2</ruleml:var>
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom   swrlx:property="hasSyn">
      <ruleml:var>x2</ruleml:var>
      <ruleml:var>x3</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
</ruleml:_head>
  <swrlx:individualPropertyAtom   swrlx:property="hasEquivalence">
    <ruleml:var>x1</ruleml:var>
    <ruleml:var>x3</ruleml:var>
  </swrlx:individualPropertyAtom>
</ruleml:_head>
</ruleml:imp>

```

Algorithm – Step 3:

After passing the **equivalent test (E)**, three tests are carried out for data labels to test their similarity in terms inclusiveness (IC), disjointness (D) and consistency (CN).

GIVEN equivalence
 EXECUTE IC, D, and CN test
 FOR all candidate pairs ($C_i:C_j$) and ($c_i:c_i$)
 IF candidate pairs are IC, not D, and CN
 THEN Go to step 6
 ELSE End

SWRL Rule – Step 3, 4 and 5:

Rule 3- $\text{hasIC} (?x1) \rightarrow \text{Inclusive} (?x1)$

```

  Implies (Antecedent (hasIC (I-variable (x1)))
    Consequent (Inclusive (I-variable (x1))))

```

Rule 4- $\text{notD} (?x1) \rightarrow \text{notDisjoint} (?x1)$

```

  Implies (Antecedent (notD (I-variable (x1)))
    Consequent (notDisjoint (I-variable (x1))))

```

Rule 5- $\text{hasCN} (?x1) \rightarrow \text{Consistent} (?x1)$

```

  Implies (Antecedent (hasCN (I-variable (x1)))
    Consequent (Consistent (I-variable (x1))))

```

Rule 3

```

<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom   swrlx:property="hasIC">

```

```

        <ruleml:var>x1</ruleml:var>
      </swrlx:individualPropertyAtom>
    </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="Inclusive">
      </swrlx:individualPropertyAtom>
    </ruleml:_head>
  </ruleml:imp>

```

Rule 4

```

<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="hasIC">
      <ruleml:var>x1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="Inclusive">
      </swrlx:individualPropertyAtom>
    </ruleml:_head>
  </ruleml:imp>

```

Rule 5

```

<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="hasIC">
      <ruleml:var>x1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="Inclusive">
      </swrlx:individualPropertyAtom>
    </ruleml:_head>
  </ruleml:imp>

```

Algorithm – Step 6 to 13:

```

GIVEN IC, D, and CN
EXECUTE Syntactic Match
EXECUTE Semantic Match
FOR all candidate pairs ( $C_i:C_j$ ) and ( $c_i:c_i$ )
LET aggregated matching score = SRS
SET SRS threshold  $t = 0.5$ 
IF scores  $> t$ 
THEN Go to step 11
ELSE Go to step 12
EXECUTE Manual log
THEN Go to step 13
EXECUTE Mappings
ENDIF

```


SWRL Rule 6 to 13:

Rule 6

hasSyntacticMatch(?x1) \rightarrow Syntactic(?x1)

Implies(Antecedent(hasSyntacticMatch (I-variable(x1)))
Consequent(Syntactic(I-variable(x1))))

```
<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="hasSyntacticMatch">
      <ruleml:var>x1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="Syntactic">
      </swrlx:individualPropertyAtom>
    </ruleml:_head>
</ruleml:imp>
```

Rule 7

hasSemanticMatch(?x1) \rightarrow Semantic(?x1)

Implies(Antecedent(hasSemanticMatch (I-variable(x1)))
Consequent(Semantic(I-variable(x1))))

```
<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="hasSemanticMatch">
      <ruleml:var>x1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="Semantic">
      </swrlx:individualPropertyAtom>
    </ruleml:_head>
</ruleml:imp>
```

Rule 8

hasAggregateScore(?x1) \rightarrow PopulateMatrix(?x1)

Implies(Antecedent(hasAggregateScore (I-variable(x1)))
Consequent(PopulateMatrix(I-variable(x1))))

```
<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="hasAggregateScore">
      <ruleml:var>x1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="PopulateMatrix">
      <ruleml:var>x1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_head>
</ruleml:imp>
```

```

        </swrlx:individualPropertyAtom>
      </ruleml:_body>
    </ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="PopulateMatrix">
      </swrlx:individualPropertyAtom>
    </ruleml:_head>
  </ruleml:imp>

```

Rule 9-10

hasAggregateScore(?x1) → SetSRS(?x1)

Implies(Antecedent(hasSemanticMatch (I-variable(x1)))
Consequent(SetSRS(I-variable(x1))))

```

<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="hasAggregateScore">
      <ruleml:var>x1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="SetSRS">
      </swrlx:individualPropertyAtom>
    </ruleml:_head>
</ruleml:imp>

```

Rule 11

hasSRSAbove(?x1) → t>0.5(?x1)

Implies(Antecedent(hasSRSAbove (I-variable(x1)))
Consequent(t>0.5(I-variable(x1))))

hasSRSAbove(?x1) → DomainExpertSelection(?x1)

Implies(Antecedent(hasSRSAbove (I-variable(x1)))
Consequent(DomainExpertSelection (I-variable(x1))))

```

<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property=" hasSRSAbove ">
      <ruleml:var>x1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="t >0.5">
      </swrlx:individualPropertyAtom>
    </ruleml:_head>
</ruleml:imp>

```

```

<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property=" hasSRSAbove ">
      <ruleml:var>x1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom
swrlx:property="DomainExpertSelection">
    </swrlx:individualPropertyAtom>
  </ruleml:_head>
</ruleml:imp>

```

Rule 12

hasSRSBelow(?x1) → t < 0.5 (?x1)

```

    Implies(Antecedent(hasSRSBelow (I-variable(x1)))
      Consequent(t < 0.5 (I-variable(x1))))

```

hasSRSBelow(?x1) → WriteManualLog(?x1)

```

    Implies(Antecedent(hasSRSBelow (I-variable(x1)))
      Consequent(WriteManualLog (I-variable(x1))))

```

```

<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property=" hasSRSBelow ">
      <ruleml:var>x1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="t < 0.5">
    </swrlx:individualPropertyAtom>
  </ruleml:_head>
</ruleml:imp>

```

```

<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property=" hasSRSBelow ">
      <ruleml:var>x1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="WriteManualLog">
    </swrlx:individualPropertyAtom>
  </ruleml:_head>
</ruleml:imp>

```

Rule 13

$\text{hasDomainExpertSelection} (?x1) \rightarrow \text{ExecuteMapping} (?x1)$

$\text{Implies}(\text{Antecedent}(\text{hasDomainExpertSelection } (I\text{-variable}(x1)))$
 $\text{Consequent}(\text{ExecuteMapping } (I\text{-variable}(x1))))$

```
<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="
hasDomainExpertSelection ">
      <ruleml:var>x1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="ExecuteMapping">
      </swrlx:individualPropertyAtom>
    </ruleml:_head>
</ruleml:imp>
```

REFERENCES

REFERENCES

1. Miller, G.A. and W.G. Charles, *Contextual correlates of semantic similarity* Language and Cognitive Processes, 1991. **6**(1): p. 1-28.
2. Gruber, T.R., *Toward principles for the design of ontologies used for knowledge sharing*, in *Knowledge Representation '94*. 1993.
3. Genesereth, M.R. and N.J. Nilsson, *Logical Foundations of Artificial Intelligence*. 1987, San Francisco, CA: Morgan Kaufmann Publishers. 405.
4. Doan, A., et al., *Learning to Match Ontologies on the Semantic Web*. The VLDB, 2002. **12** (4): p. 303-319.
5. Giunchiglia, F. and I. Zaihrayeu. *Making Peer Databases Interact - A Vision for an Architecture Supporting Data Coordination*. in *Conference on Information Agents*. 2002. Madrid.
6. Kalfoglou, Y. and W.M. Schorlemmer, *Ontology Mapping: State of the art*. Knowledge Engineering Review, 2003. **18**(1): p. 1-31.
7. Hovy, E. *Combining and standardizing large scale, practical ontologies for machine translation and other uses*. in *1st International Conference on Language Resources and Evaluation (LREC)*. 1998. Granada, Spain SIGMOD Record.
8. Kashyap, V. and A.P. Sheth, *Semantic and Schematic Similarities Between Objects in Databases: A Context-Based Approach*. The VLDB Journal, 1995. **5**(4): p. 276-304.

9. Missikoff, M., F. Schiappelli, and F. Taglino. *A Controlled Language for Semantic Annotation and Interoperability in e-Business Applications*. in *ISWC 03*. 2003. Sanibel Island, Florida.
10. Madhavan, J., et al. *Representing and Reasoning about Mappings between Domain Models*. in *Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence*. 2002. Edmonton, Alberta, Canada: AAAI Press.
11. Maedche, A., B. Motik, and N. Stojanovic. *MAFRA - A MApping FRAmework for Distributed Ontologies*. in *13th International Conference, Knowledge Engineering and Knowledge Management (EKAW 2002)*. 2002. Siguenza, Spain.
12. Park, J. and S. Ram, *Information Systems Interoperability: What Lies Beneath?* *ACM Transactions on Information Systems (TOIS)*, 2004. **22**(4): p. 595-632.
13. Morikawa, A.R.Y. and L. Kerschberg. *MAKO: Multi-Ontology Analytical Knowledge Organization based on Topic Maps*. in *Fifth International Workshop on Knowledge Management, DEXA, 2004*. Zaragoza, Spain.
14. Fowler, J., et al., *Agent-Based Semantic Interoperability in Infosluth*. *ACM SIGMOD Record*, 1999. **28**(1): p. 60-67.
15. Muthaiyah, S. and L. Kerschberg. *Dynamic Integration and Semantic Security Policy Ontology Mapping for Semantic Web Services (SWS)*. in *First IEEE International Conference on Digital Information Management (ICDIM)*. 2006. Bangalore, India.
16. Dao, S. and B. Perry. *Applying a Data Miner to Heterogeneous Schema Integration*. in *Proceedings of International Conference on Knowledge Discovery and Data Mining*. 1995. Montreal, Quebec, Canada: AAAI Press.
17. Grüninger, M. and J. Kopena. *Semantic integration position statement*. in *Second International Semantic Web Conference*. 2003. Sanibel Island, FL, USA.

18. Ram, S., J. Park, and D. Lee, *Digital Libraries for the Next Millennium: Challenges and Research Directions*. Information System Frontiers, 1999. **1**(1): p. 75-94.
19. Ventrone, V. and S. Heiler, *Semantic Heterogeneity as a Result of Domain Evolution*. SIGMOD Record, 1991. **20**(4): p. 16-20.
20. Ram, S. and J. Park, *Semantic Conflict Resolution Ontology (SCROL): An Ontology for Detecting and Resolving Data- and Schema-Level Semantic Conflicts*. IEEE Transactions on Knowledge and Data Engineering, 2004. **16**(2): p. 189-202.
21. Halevy, A.Y., et al, *The Piazza Peer Data Management System*. IEEE Transactions on Knowledge and Data Engineering, 2004. **16**(7): p. 787-798.
22. Li, L., B. Wu, and Y. Yang. *Agent-Based Ontology Mapping Towards Ontology Interoperability*. in *Australian Joint Conference on Artificial Intelligence (AI'05)*. 2005. Sydney, Australia: LNAI 3809, Springer-Verlag.
23. Noy, N.F. and M.C.A. Klein. *A component-based framework for ontology evolution*. in *IJCAI*. 2003. Acapulco, Mexico.
24. Tomás, J., J.T. Fernández-Breis, and R. Martínez-Béjar, *A cooperative framework for integrating ontologies*. International Journal of Human-Computer Studies, 2002. **56**(6): p. 665-720
25. Calvanese, D., G. De Giacomo, and M. Lenzerini. *A Framework for Ontology Integration*. in *Proceedings of the First Semantic Web Working Symposium, SWWS-01*. 2001. Stanford, USA.
26. Kiryakov, A., K.I. Simov, and M. Dimitrov. *OntoMap: Portal for Upper-Level Ontologies*. in *Proceedings of the 2nd Conference on Formal Ontology in Information Systems* 2001. Ogunquit, Maine, USA: ACM Press

27. Kent, R.E. *The Information Flow Foundation for Conceptual Knowledge Organization*. in *Proceedings of the Sixth International ISKO Conference*. 2000. Toronto, Canada.
28. Barwise, J. and J. Seligman, *Information Flow: The Logic of Distributed Systems*. 1997: Cambridge University Press.
29. Stumme, G. and A. Maedche. *FCA-MERGE: Bottom-Up Merging of Ontologies*. in *7th International Conference on Artificial Intelligence (IJCAI'01)*. 2001. Seattle, WA, USA.
30. Ganter, B., S. Gerd, and W. Rudolf, *Formal Concept Analysis Foundations and Applications* Vol. 3626 2005: Lecture Notes in Computer Science
31. Kalfoglou, Y. and W.M. Schorlemmer, *IF-Map: An Ontology-Mapping Method Based on Information-Flow Theory*. *Journal of Data Semantics*, 2003. **1**(1): p. 98-127.
32. Noy, N.F. and M.A. Musen. *SMART: Automated Support for Ontology Merging and Alignment*. in *Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management*. 1999. Banff, Canada.
33. Noy, N.F. and M.A. Musen. *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment*. in *Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*. 2000. Austin, Texas: The MIT Press.
34. Noy, N.F. and M.A. Musen. *Evaluating ontology-mapping tools: Requirements and experience*. in *Proceedings of the Workshop on Evaluation of Ontology Tools at EKAW'02 (EON2002)*. 2002. Siguenza, Spain.
35. Noy, N.F., et al., *Creating semantic web contents with protege-2000*. *IEEE Intelligent Systems*, 2001. **16**(2): p. 60-71.

36. Doan, A., et al. *Ontology Matching: A Machine Learning Approach*. in *Proceedings of the 11th international conference on World Wide Web*. 2002. Honolulu, Hawaii, USA.
37. Lacher, M.S. and G. Groh. *Facilitating the exchange of explicit knowledge through ontology mappings*. in *14th International FLAIRS conference*. 2001. Key West Florida, USA: AAAI Press.
38. Mitra, P. and G. Wiederhold. *Resolving Terminological Heterogeneity In Ontologies*. in *Proceedings of Workshop on Ontologies and Semantic Interoperability, 15th European Conference on Artificial Intelligence (ECAI)*. 2002. Lyon, France.
39. Chalupsky, H. *OntoMorph: A Translation System for Symbolic Knowledge*. in *Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000)*. 2000. San Francisco, CA: Morgan Kaufmann.
40. Stuckenschmidt, H., H. Wache, and U. Visser. *Information Integration on the World Wide Web* in *2nd International Semantic Web Conference 2003*. Sanibel Island, Florida.
41. Stuckenschmidt, H., H. Wache, and U. Visser. *Information Integration on the World Wide Web* in *2nd International Semantic Web Conference 2003*. 2003. Sanibel Island, Florida.
42. Kurgan, L.A., W. Swiercz, and K.J. Cios. *Semantic Mapping of XML Tags Using Inductive Machine Learning*. in *ICMLA 2002*. 2002.
43. Muller, C. and I. Gurevych. *Exploring the Potential of Semantic Relatedness in Information Retrieval*. in *LWA 2006*. 2006.
44. Gilleland, M. *Levenshtein Distance, in Three Flavors*, <http://www.merriampark.com/ld.htm> [cited; Available from: <http://www.merriampark.com/ld.htm>, <http://www.merriampark.com/ld.htm#DEMO>].

45. Lesk, M. *Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone.* in *5th Annual International Conference on Systems Documentation*. 1986. Toronto, Ontario, Canada.
46. Weeds, J. and D. Weir, *Co-occurrence Retrieval: a General Framework for Lexical Distributional Similarity*. *Computational Linguistics* 2005. **31**(4): p. 439-476.
47. Budanitsky, A. and G. Hirst, *Evaluating WordNet-based measures of semantic distance*. *Computational Linguistics*, 2006. **32**(1): p. 13-47.
48. Leacock, C. and M. Chodorow, *Combining local context and WordNet similarity for word sense identification in WordNet: An Electronic Lexical Database*, C. Feldbaum, Editor. 1998. p. 265-283.
49. Resnik, P. *Using information content to evaluate semantic similarity.* in *14th International Joint Conference on Artificial Intelligence* 1995. Montreal, Canada.
50. Resnik, P. *WordNet and distributional analysis: A class-based approach to lexical discovery.* in *AAAI Workshop*. 1992: AAAI Press.
51. Resnik, P., *Semantic Similarity in a Taxonomy : An Information based Measure and its Application to Problems of Ambiguity in Natural Language*. *Journal of Artificial Intelligence Research*, 1998. **11**: p. 95-130.
52. Jiang, J.J. and D.W. Conrath. *Semantic similarity based on corpus statistics and lexical taxonomy.* in *International Conference on Research in Computational Linguistics*. 1997. Taiwan.
53. Lin, D. *An information-theoretic definition of similarity.* in *15th International Conference on Machine Learning*. 1998. Madison, WI, USA.
54. Hirst, G. and D. St-Onge, *Lexical chains as representations of context for the detection and correction of malapropisms,* in *WordNet: An Electronic Lexical Database*, C. Feldbaum, Editor. 1998, MIT Press. p. 305-332.

55. Turney, P.D. *Mining the Web for Synonyms: PMI-IR Versus LSA on TOEFL*. in *Twelfth European Conference on Machine Learning (ECML-2001)*. 2001. Freiburg, Germany.
56. Cilibrasi, R. and P. Vitanyi. *Similarity of Objects and the Meaning of Words*. in *3rd Conference on Theory and Applications of Models of Computation (TAMC)*. 2006. Beijing, China
57. Cilibrasi, R. and P. Vitanyi, *The Google Similarity Distance*. IEEE Transactions Knowledge and Data Engineering, 2007. **19**(3): p. 370-383.
58. Landauer, T.K. and S.T. Dumais, *A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge*. Psychological Review, 1997. **104**: p. 211-140.
59. Matveeva, I., et al. *Term Representation with Generalized Latent Semantic Analysis*. in *Conference on recent advances in natural language processing (RANLP)*. 2005.
60. Anderson, J.R. and P.L. Pirolli, *Spread of Activation*. Journal of Experimental Psychology: Learning, Memory, and Cognition., 1984. **10**(4): p. 791-798.
61. Horrocks, I., et al. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. 2004 [cited 2008 23 Jan]; Available from: <http://www.w3.org/Submission/SWRL/>.
62. Beckett, D., et al. *Resource Description Framework (RDF)*. 2004 [cited 2008 Jan 23]; Available from: <http://www.w3.org/RDF/>.
63. *Jess, the Rule Engine for Real Programmmers*. [cited 2008 23 Jan]; Available from: <http://www.jessrules.com/jess/index.shtml>.
64. Muthaiyah, S. and L. Kerschberg, *Virtual Organization Security Policies: An Ontology-based Mapping and Integration Approach*. Information Systems Frontiers 2007. **9**(5): p. 505-515.

65. Muthaiyah, S. and L. Kerschberg. *Virtual Organization Security Policies: An Ontology-based Mapping and Integration Approach*. in *The Second Secure Knowledge Management Workshop (SKM) 2006*. Brooklyn, New York.
66. Rubenstein, H. and J.B. Goodenough, *Contextual correlates of synonymy*. Communications of the ACM, 1965. **8**(10): p. 627-633.
67. Coakes, S.J. and L.G. Steed, eds. *SPSS Analysis without Anguish (10th Edition)*. Version 10 for Windows ed. 2001, John Wiley.

CURRICULUM VITAE

Saravanan Muthaiyah graduated from the National University of Malaysia, Bangi, Malaysia, in 1997 with an honors degree in Finance. He received a Masters of Science degree in Information Technology from Putra University, Serdang, Malaysia in 2000. He has worked for reputable organizations among others such as IBM, Malaysia and has vast corporate experience in accounting, audit and information systems. He then joined University of Malaya for a few years and later joined Multimedia University in 2001 to become a full-time lecturer. He was quickly promoted to Senior Lecturer in 2004 based on his outstanding service and performance track record. In 2004, he won the prestigious Fulbright award which granted him a scholarship to pursue a doctoral program in Information Technology at the Department of Computer Science in George Mason University, Fairfax, VA, USA. Mr. Muthaiyah has published several articles, journals, books and book chapters as well. He has also received the George Mason Doctoral Fellowship award in 2006, 2007 and 2008.