Similarity Identification and Measurement between Ontologies

Amjad Farooq and Abad Shah

Computer Science and Engineering Department
University of Engineering and Technology, Lahore – Pakistan
amjadfarooquet@gmail.com

Abstract: The retrieval of relevant and precise information from web has always been remained a serious problem. To address this problem, the idea of ontologies-based web, so-called semantic web, was proposed in 2001. But the problem is not completely solved due to the semantic heterogeneity suffered by ontologies. In this paper we propose a semi-automatic technique to measure the explicit semantic heterogeneity. The proposed technique identifies all candidate pairs of similar concepts without omitting any similar pair. The proposed criteria for similarity measurement are based on theme semantic web. The proposed technique can be used in different types of operations on ontologies such as merging, mapping and aligning. By analyzing its results a reasonable improvement in terms of completeness, correctness and overall quality of the results has been found. . [Journal of American Science 2010; 6(4):67-85]. (ISSN: 1545-1003).

Keywords: Semantic Web, Heterogeneity, Ontology Matching, Similarity Identification

1. Introduction

The World Wide Web (or the Web) is a global source of information, which includes information about almost every topic that a person can think. But it is difficult to retrieve relevant, specific and precise information due to semantic heterogeneity and the lack of machine understandability of contents. It has been estimated that only 37 percent to 52 percent relevant results are retrieved and other retrieved results are irrelevant (Lewandowski, 2008). The idea of semantic web was envisioned by Lee (Lee et al., 2001), which provides a promising solution to overcome the retrieval performance problem of the web. According to the theme of the semantic web, the web-contents need to be structured, formalized, stored and retrieved through ontologies.

When multiple ontologies are simultaneously used in the integrating operations such as merging, mapping and aligning then they may suffer from different types of heterogeneities such as semantic heterogeneity and non-semantic or syntactic heterogeneity (Shvaiko & Euzenat, 2008; Hauswirth & Maynard, 2007). The syntactic heterogeneity occurs due to the use of different languages. The semantic heterogeneity includes terminological, conceptual and contextual heterogeneities. The terminological heterogeneity arises when different terms are used to represent the same concept or the same term is used to represent different concepts. The conceptual heterogeneity between two concepts may occur due to their different level of granularities i.e., when a concept is sub-concept or super-concept of the other, or both are overlapped. Similarly, two concepts are explicit-semantically heterogeneous if they are terminologically and taxonomically similar but they have different roles or functionalities in their respective ontologies.

To handle the problem of ontological semantic heterogeneity, it is required to identify the similarity between ontologies. For this purpose different techniques have been proposed and reported in the literature (Shvaiko & Euzenat, 2009; Maedche & Staab, 2002; Hariri et al., 2006; Aleksovski et al., 2006; Trojahn et al., 2008; Jeong et al., 2008;Noy & Musen, 2001; Melnik et al., 2002). However, some issues are still unsolved. Explicit semantic similarity needs to be measured in order to carry the vision of semantic web (González, 2005; Uschold, 2003; Uschold, 2002). The measurement of degree of similarity (*DoS*) based on Edit-distance formula, is unreliable because it measures the *DoS* based on the criteria of finding terms-similarity rather than finding similarity between concepts represented by the terms. The criteria as reported in (Shvaiko & Euzenat, 2005; Erhard & Philip, 2001; Lambrix &Tan, 2006), used for the identifying taxonomic similarity between concepts of two ontologies declare certain pairs of similar concepts as dissimilar due to the biasness of these criteria towards those concepts whose siblings-concepts, sub-concepts or direct super-concepts are not similar. Most of the existing similarity measurement techniques only compute the *DoS* between concepts of two ontologies (Buccella et al., 2005; Giunchiglia et al., 2007), which is inadequate to determine that which concept is more generic or more specific than the other, and this issue is considered as an open research issue (Janowicz et al., 2008). Similarly, some existing techniques compute only the Semantic Relation (*SR*) between two concepts (Giunchiglia et al., 2007). Although, *SR* shows that one concept is more generic, or more specific than the other concept, yet it does not give the level of generality. Furthermore, the measurement of semantic similarity is a complex and is inefficient in execution-wise (Janowicz et al., 2008).

The above mentioned shortcomings in the existing similarity measurement techniques motivate us, to propose an integrated technique based on innovative vision of semantic web to achieve the following objectives: (i) identifying all pair of similar concepts without omitting any candidate pair of similar concepts. (ii) Identifying and measuring the explicit semantic similarity between intellectual concepts of ontologies.

The remainder of the paper is organized as follow. In Section 2, the background and related work is presented. The proposed technique is given in Section 3, and it is validated via case studies in Section 4. Results are analyzed and discussed in Section 5 and finally the paper is concluded in Section 6.

2. Background and Related Work

For aligning ontologies, several techniques have been proposed (Duchateau et al., 2007; Alasoud et al., 2008; Sherman & Price, 2001; Shvaiko & Euzenat, 2005; Erhard & Philip, 2001; Lambrix &Tan, 2006). On the basis of similarity-measuring criteria, these techniques are categorized into schema-based and instance-based techniques. In schema-based techniques, similarity between concepts is measured at structure-level while ignoring their actual data, whereas in instance-level techniques, similarity is measured by taking actual data into consideration. In structural aligning, the taxonomic characteristics of concepts are mostly considered. The two concepts are rendered taxonomically similar (Shvaiko & Euzenat, 2005; Erhard & Philip, 2001; Lambrix &Tan, 2006) if (i) their direct super-concepts are similar; (ii) their sibling-concepts are similar; (iii) their direct sub-concepts are similar; (iv) their descendant-concepts are similar; (v) their leaf-concepts are similar and vi) concepts, in the paths from the root to those concepts, are similar. Irrespective of the structural aligning technique used, it has been observed that certain pairs of similar concepts are categorized dissimilar because of bias of above mentioned criteria towards those concepts whose siblings-concepts, sub-concepts or direct super-concepts are not similar.

In (Aleksovski et al., 2006), the background knowledge of domain has been used via ontology to determine similarity between concepts of two ontologies, especially for those concepts which are not lexically and structurally similar. It has been evaluated by matching a medical ontology to another while using comprehensive medical domain ontology as background knowledge. This technique is well suited for those ontologies having very poor taxonomic and non-taxonomic relations between concepts. There are some other approaches for measuring semantic similarities between concepts of XML schemas, database schemas and some graph-like structures (Giunchiglia et al., 2007, Janowicz et al., 2008; Jeong et al., 2008; Noy & Musen, 2001; Melnik et al., 2002; Duchateau et al., 2007). In these schemas, the explicit meanings of concepts are determined either from their respective attributes or from their hierarchical positions. The meanings of concepts in terms of their interactions with other concepts are not explicitly defined in these schemas. Therefore these approaches seem to be inappropriate for measuring the similarities between concepts of ontologies schemas.

Ontology matching technique, proposed in (Alasoud et al., 2008) has three phases. It uses Levenshtein Distance (Cohen et al., 2003) and WordNet (Pedersen et al., 2004) techniques in first phase. A matrix with binary values is the output of first phase. For a_i there may be multiple corresponding b_js that are similar and only one among them is short listed on the bases of high score computed in the third phase from the neighbors of those concepts search in the second phase. This algorithm does not properly differentiate between concepts and its data properties and object properties. Data types and constraints are ignored while measuring similarity between data properties. Due to use of Levenshtein Distance formula of degree of similarity, the completeness and the correctness of the result is comparative low. Secondly, the pairs of similar concepts are not accompanied by their semantic relations. Whole-part relationships are only considered whereas the role-based and taxonomic characteristics are overlooked.

In order to detect and retrieve relevant ontologies Alexander Maedche, and Steffen Staab (Maedche & Staab, 2002) proposed a set of similarity measures for ontologies. The lexical and conceptual aspects of concepts of ontologies are considered. In lexical level measures, the terms used to name concepts are compared and their similarity is computed using well known method known as edit distance (Cohen et al., 2003) and they proposed a lexical metric for similarity computing which is equal to *MAX (0, MIN (|Li|, |Lj|) – ed (Li, Lj) / MIN (|Li|, |Lj|)),* where Li and Lj are two lexical entities whose similarity is being computed. The metric value varies in between 0 and 1. The 0 means both are dissimilar whereas the 1 indicates the similarity exactness of terms. The *ed* is a function that returns an integer which is equal to number of insertions, deletions or substitutions to transform one lexical term into other. At conceptual level, the similarity is computed from the similarities of their respective super-entities. Two entities are similar if their direct super-entities in their respective taxonomies are similar or all super-entities of first entity are similar to super-entities of second entity used in comparison.

In (Trojahn et al., 2008), composite ontology mapping technique has been proposed. Different existing matchers have been collectively used in this technique. The technique has been automated through agent-based scenarios. For lexical similarity measuring, they use the string-based measures and to examine the linguistic semantics of terms, the WordNet has been used. The structural similarity between two terms has been computed based on the similarities of their respective super and sub concepts. The overall degree of similarity has been computed from the lexical, linguistic and structure similarities of terms.

In (Buccella et al., 2005), syntactic and semantic matchers are used to compute similarity, and final decision is made by the user. The syntactic matcher uses string-based techniques, known as edit distance and n-gram to measure the degree of similarity between two input terms. For semantic comparison, a thesaurus is searched for synonyms of input terms and then comparison is made using synonyms. During semantic matching, the depth of concepts from their common super-concepts in their respective taxonomies, are also considered. The overall degree of similarity is computed from the results of syntactic matcher and semantic matcher.

In HCONE (Kotis & Vouros, 2004), the ontology is defined as a set of terms used to represent concepts, their relationships and data-properties alongwith the axioms for interpretation of terms. Using WordNet and semantic index method, the highly ranked sense of each term is located and identified. For each term, all generic and specific terms are also retrieved from wordNet and then semantic relation between two terms, based on this information is identified. Finally, the merging decision based on the semantic relation, is made.

There are some others ontology merging, mapping and alignment tools (McGuinness et al., 2000; Maedche & Staab, 2002; Bouquet et al., 2003; Hariri et al., 2006; Lambrix &Tan, 2006). Each of them uses almost the same matching techniques to measure the similarity between concepts of ontologies. These toots use string-based techniques such as edit distance and n-gram to measure the degree of similarity between terms used for representing concepts. Some of them use WordNet to get linguistic information such as synonyms and hyponyms while measuring similarity and then the structural information of terms are further used to compute the overall degree of similarity.

Most of the existing works as summarized in Table 1 are about the measurement of similarity between two concepts based on their names, linguistic semantics, and the similarities of their taxonomic characteristics such as super-concepts, sub-concepts and sibling-concepts. However, no attention has been given on the explicit semantics based similarity measurement between concepts of ontologies. Secondly, the existing techniques compute only the *DoS* between concepts of two ontologies (Buccella et al., 2005; Giunchiglia et al., 2007). The value of *DoS* remains between 0 and 1 which is inadequate to determine as to which concept is more generic or more specific than the other one. It has been considered as open research issue (Janowicz et al., 2008). Similarly, some existing techniques compute only the Semantic Relation (*SR*) between two concepts (Giunchiglia et al., 2007). Although, *SR* shows that one concept is more generic, or more specific than the other concept, yet it does not give the level of generality. Therefore, each pair of similar concepts should be accompanied by their both *DoS* and *SR* in order to take a better decision while performing the aligning, merging and mapping operations of ontologies.

The measurement of degree of similarity (*DoS*) based on Edit-distance formula may produce incorrect results because the *DoS* is measured based on terms rather than concepts represented by those terms. That is, some pairs of similar concepts are declared dissimilar because of the heterogeneous terms used for the names those concepts. Similarly, some pairs of dissimilar concepts are declared similar because of the similarity of terms used for those concepts. Some approaches consider the synonyms provided by the WordNet while measurement of similarity. Their main considerations are the terms or the synonyms of terms rather than concepts represented by those terms and secondly, most of the tools consider the taxonomic characteristics of concept i.e., their relations with parents and children. The taxonomic similarity measurement criteria (Shvaiko & Euzenat, 2005; Erhard & Philip, 2001; Lambrix &Tan, 2006), as discussed before, declare certain pairs of similar concepts as dissimilar because of the biasness of these criteria towards those concepts whose siblings-concepts, sub-concepts or direct super-concepts are not similar.

3. Proposed Similarity Identification and Measurement Technique

First we give and list the basis of our proposed technique:

Table 1. A comparison of some techniques for similarity measurement between ontologies

Techniques	**Name-based similarity**	**Linguistic-based similarity**	**Taxonomic-based similarity**	**NonTax. based similarity**	**DOS**	**SR**
TAOM (Buccella et al., 2005)	Edit-distance , n-gram	Thesaurus	Parents	N	Y	N
MSBO (Maedche & Staab, 2002)	Edit-distance	N	Parents	N	Y	N
SEMC (Bouquet et al., 2003)	Edit-distance	WordNet	Parents Children	N	Y	N
HCONE (Kotis & Vouros, 2004)	String-based Techniques.	N	Parents, Children	N	N	Y
Chimaera (McGuinness et al., 2000)	Edit-distance	N	Parents, Children	N	Y	N
SSMO (Hariri et al., 2006)	String-based Techniques.	N	Parents, Children	N	N	Y
SAMBO (Lambrix &Tan, 2006)	Edit-distance	N	Parents, Children	N	N	Y
EOMT (Alasoud et al., 2008)	Edit-distance	WordNet	Parents	N	Y	N
CACOM(Trojahn et al., 2008)	Edit-distance	WordNet	Parents , Children	N	Y	N

i) Concepts are compared instead of terms used to represent concepts.

ii) Domain-specific semantics (i.e., explicit semantics of concepts) are being used in similarity measurement process, rather than their linguistic semantics.

iii) The super-concepts based contextual similarity measurement is computed and relaxing the similarities between their respective sub-concepts (or sibling concepts).

iv) The layered matching strategy is adopted to make the measurement process more efficient.

The proposed technique works in three phases as shown in Figure 1. The three phases are: i) IPS - Identifying Primary Similarity, ii) ICS - Identifying Contextual Similarity, iii) IRS - Identifying Role-based Similarity. There are some preprocessing tasks before the technique starts its actual working. These tasks are: (a) acquisition of concepts, (b) acquisition of super-concepts of primarily similar concepts, and (c) acquisition of roles of contextually similar concepts, are performed by the three phases, respectively. The structure diagram of proposed technique is shown in Figure 1. In the figure, *M* and *N* are two RDF models of the two input ontologies *A* and *B*, respectively. The *CA*, *PA* and *RA* represent concepts-acquisition, parent-acquisition and roles-acquisition processes, respectively. The *IPS*, *ICS* and *IRS* represent processes of identifying primary similarity, identifying contextual similarity and identifying role-based similarity, respectively. The label *1* represents two separate lists of concepts acquired from the models *M* and *N*, respectively. The label *2* represents a list of pairs of primarily similar concepts. The label *3* represents two separate lists of parents of primarily similar concepts, and the label *4* represents a list pairs of concepts possessing contextual similarity. Label *5* represents two separate lists of roles of contextually similar concepts, and the label *6* represents a list of pairs of concepts possessing role-based similarity. In the figure, O_1, O_2 and O_3 are the three (3) vectors containing pairs of primarily, contextually and explicit semantically similar concepts, respectively.

3.1 Definitions

(a) In an ontology we define a *concept* as a class of objects sharing common elementary, taxonomic and non-taxonomic characteristics. We define a concept as a 5-tuple i.e. <*T, P, C, S, R*>; where *T, P, C, S* and *R* are sets of terms, parents, children, siblings and roles respectively, that a concept may have. These sets are formally defined as:

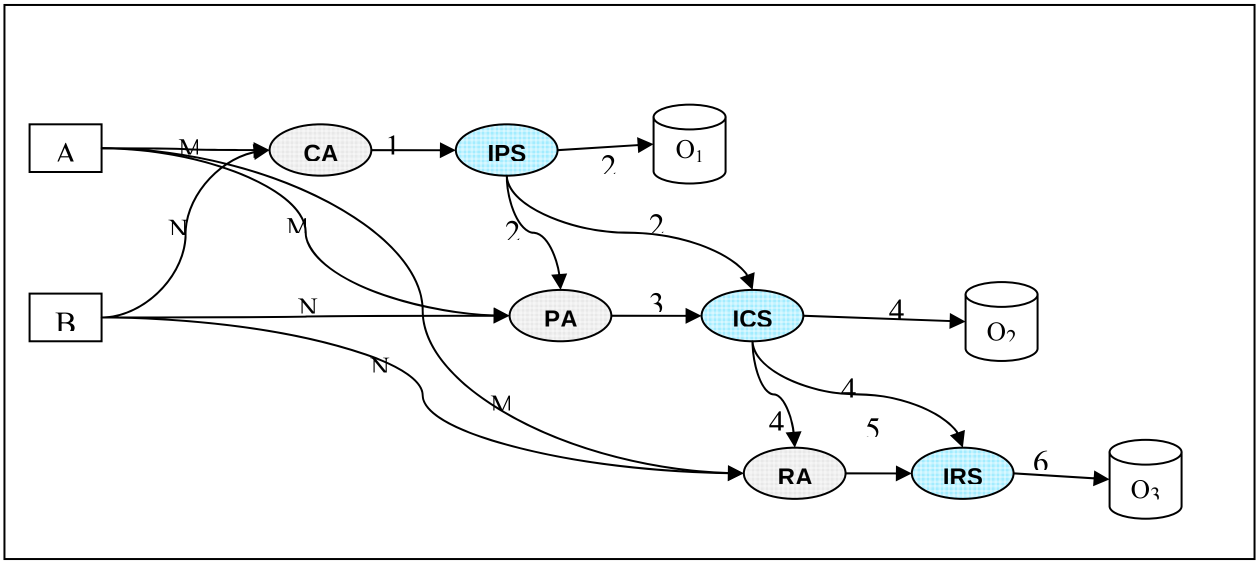


Figure 1: Structure diagram of the proposed technique

$T = \{term_i \ \ |1 \leq i \leq N_1\}$ (1)
$P = \{parent_i \ |1 \leq i \leq N_2\}$ (2)
$C = \{child_i \ \ |1 \leq i \leq N_3\}$ (3)
$S = \{sibling_i \ |1 \leq i \leq N_4\}$ (4)
$R = \{role_i \ \ |1 \leq i \leq N_5\}$ (5)

A concept has linguistic and explicit semantics. The synonyms of a concept represent its linguistic or implicit semantics whereas the explicit semantics of a concept are defined in terms of its roles (or responsibilities), which it plays in a certain domain. In other words, the explicit semantic of a concept is domain dependent. If a concept *C* plays the roles r_1, $r_2 \ldots r_n$ in a domain *D*, then the explicit semantic of the concept *C* is formally defined as follows:

$_{ES}C^D = \{r_1, r_2 \ldots r_n\}$ (6)

(b) We refer to the 1st level similarity as the *Primary Similarity*. Two concepts are primarily similar if and only if either their names belong to *T* (see Equation (1)) of the concept C_i where $1 \leq i \leq N$; or the first-name belongs to *T* and second-name belongs to *P* (see Equation (2)) or versa where *T* and *P* both belong to the same concept C_i. The primary similarity is denoted as $\approx^1$, and can formally be written as follows:

$a \approx^1 b$ Iff $((a \wedge b) \in T(c_i)) \vee ((a \in T(c_i) \wedge b \in P(c_i)) \vee (b \in T(c_i \wedge a \in P(c_i)))$ (7)

In Equation (7), $T(c_i)$ and $P(c_i)$ are the two sets of terms and parents of the concept c_i of the target domain.

(c) The 2nd level similarity is referred it to as *contextual similarity*. Two concepts are contextually similar if and only if they possess the primary similarity and have one or more common concepts in their respective list of super-concepts. It can formally be written as follows:

$a \approx^2 b$ Iff ((condition given in Eq. (7) is true) $\wedge$ $((P_a \cap P_{b)} \neq \phi$)) (8)

In Equation (8), P_a and P_b are the two respective sets of parents of the concepts *a* and *b* as we have already defined in Equation (2).

(d) We refer to the 3rd level of similarity (or explicit-semantics based similarity) as r*ole-based similarity*, and it is especially used for identifying similarity between two intellectual concepts. Two concepts possess the role-based similarity if and only if they possess contextual similarity and they have one or more common roles in their respective list of roles.

$a \approx^3 b$ Iff ((condition given in Eq. (8) is true) $\wedge$ $(R_a \cap R_b \neq \phi)$) (9)

In Equation (9), R_a and R_b are the respective sets of roles (as we have defined in Equation (5)) of concepts *a* and *b*.

(e) Since there may be multiple roles of same concept, therefore, while identifying the similarity, we consider common roles of two concepts *a* and *b*, then the *role-based DoS* of the concept *a* with respect to the concept *b* is computed by dividing total number of common roles by total number of roles in their union. Assume *M* is a set of roles of a concept *a* and *N* is a set of roles of another concept *b*. Bother sets belong to two different ontologies *A* and *B* respectively, then *DoS* between the concepts *a* and *b* is computed by using the following empirical formula.

$$DoS = \frac{|M \cap N|}{|M \cup N|} \qquad (10)$$

Role-Based SR between pairs of similar concepts *a* and *b* may is denoted as follows.

$(a, b) = SR$ (11)

The criteria for computing *SR* are listed as follows:

i. SR = '=' ; a is equivalent to b;
$a = b$ iff $((|M \cap N| = |M|)$ & $(|M| - |N| = 0))$

ii. SR= '≥ '; a is more generic than b;
$a \geq b$ iff $(|M \cap N| = |N|)$

iii. SR='≤'; a is less generic than b;
$a \leq b$ iff $(|M \cap N| = |M|)$

iv. SR= 'x'; otherwise semantic relation is undefined; (x), take manual decision

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

(f) Granularity of a concept (*Gc*) is proportional to its level of generality. The generality of a concept may vary from the most generic to least generic or vice versa. Let *g* be the generality of the concept c and *k* is the constant of proportionality, then we define *Granularity-Based Degree of Similarity Gc* as follows:

$Gc = k * g$ (12)

In Equation 12, *g* can vary between *1* and *n*, where *n* is an integer value. If *g* is equal to 1, then the concept is considered to be the most generic concept, and if *g* is equal to *n*, then the concept is considered to be the least generic. We compute the *DoS*, between two concepts particularly the non-intellectual concepts from their granularities. Let *Ga* and *Gb* be the granularities of two primarily similar concepts *a* and *b* respectively, then their *DoS* is computed by using the empirical formula given in Equation (13).

$$DoS = \frac{|Ga - Gb|}{Max(Ga, Gb)} \qquad (13)$$

If *Ga* and *Gb* are the granularities of primarily two similar concepts *a* and *b* respectively, then *SR* between them can be computed by Equation (14).

$(a, b) = SR$ (14)

The criteria for computing SR are given below:

(i) SR = '=' ; a is equivalent to b;
$a = b$ iff $(Ga = Gb)$

(ii) SR= '≥ '; a is more generic than b;
$a \geq b$ iff $(Ga < Gb)$

(iii) SR= '≤'; a is less generic than b;
$a \leq b$ iff $(Ga > Gb)$

3.2 IPS - Identifying Primary Similarity Phase

The primary similarity (defined earlier) is not the same as terminological similarity because we mainly focus on logical meaning of concepts instead of terms used to represent the concepts. The identifying process of the primary similarity is given in algorithmic form in Figure 2. The terms used to represent concepts in both source ontologies *A* and *B*, as obtained in the vectors CS_A and CS_B (defined in Equation (15)-(16)) are the input of this phase. The vector Sim_{PS} (defined in Equation (17)) containing pairs of primarily similar concepts is the output of this phase.

$CS_A = \{a_i \mid \forall\ a_i \in A; 1 \leq i \leq M\}$ (15)

$CS_B = \{b_j \mid \forall\ b_j \in B; 1 \leq j \leq N\}$ (16)

$Sim_{PS} = \{(a, b, DoS, SR) \mid \forall\ ((a \in CS_A \wedge b \in CS_B) \wedge (a \approx^1 b))$ (17)

In Equation (17), the symbol $\approx^1$ represent the primary or the first level similarity (defined in Equation (7)) whereas *DoS* and *SR* (defined in Equation (10) – (11) and Equation (13) – (14)) based on the roles and(12) granularities of concepts, respectively.

3.3 ICS - Identifying Contextual Similarity Phase

Since the conceptual similarity between two concepts (defined earlier), is based on the similarity of their respective parent concepts, therefore, we need the parent-concepts of all those concepts which are declared primarily similar concepts in the previous phase. Hence, for all concepts in the resultant vector, Sim_{PS}, obtained from Phase-1, their respective parent concepts from the(13)ontologies *A* and *B* are separately extracted in the two vectors, i.e., C^PS_A and C^PS_B, which are formally defined as follows:

```
Input:  $CS_A$ and $CS_B$  Vectors
(DV-Domain Vocabulary, an implicit input)
Output: $Sim_{PS}$ -
            a vector containing pairs of primarily
           similar concepts
Begin
For each a in $CS_A$
For each b in $CS_B$
aId= DV.getId(a);Ga = DV. getGranularity(a)
bId= DV.getId(b);Gb = DV. getGranularity(b)
S1 = aId.size();    S2 = bId.size
If (S1 = S2) then
If aId.equal(bId) then Temp.SR = '='
Else       Temp.SR = 'x'
Else if (S1 < S2) Then  T = bId.substr(1,s1)
If T.equal(aId) Then Temp.SR = '$\supseteq$'
Else Temp.SR = 'x'
Else if (S1 > S2) Then T = aId.substr(1,s2)
If T.equal(bId) Then Temp.SR = '$\subseteq$'
Else       Temp.SR = 'x'    End if
Temp.DOS = absolute (Ga – Gb) / Maximum
(Ga, Gb); $Sim_{PS}$.add(temp)
   Next
 Next
End
```

Figure 2: A slice of pseudo code for identifying primary similarity

$$C^PS_A = \{(a_i, (p_i, p_{i+1},. . ., p_k)) \mid \forall\ a_i, p_i \in A \wedge\ p_i \text{ isParentOf } (a_i)\} \quad (18)$$

$$C^PS_B = \{(b_j, (p_j, p_{j+1}, . . ., p_k,)) \mid \forall\ b_j, p_j \in B \wedge\ p_j \quad \text{isParentOf } (b_j)\} \quad (19)$$

This phase takes C^PS_A, C^PS_B (see Equation (18) – (19)) vectors, populated in the acquisition process and Sim_{PS} (see Equation (17)) populated in the previous phase, as the input and returns a set $Sim_{CS,}$ (defined in Equation (20)), containing pairs of taxonomically similar concepts as the output.

$$Sim_{CS} = \{(a, b, DoS, SR) \mid \forall\ ((a, b) \in Sim_{PS} \wedge\ (a \approx^2 b))\} \quad (20)$$

```
Algorithm: Identifying contextual similarity
Input :(i) $C^pS_A$ and $C^pS_B$ vectors
          (ii) $Sim_{PS}$ vector
Output: $Sim_{CS}$ (as defined in Eq.20); a vector containing
pairs of taxonomically similar concepts
Begin
          For each p in $Sim_{PS}$
          parent$C_a$ = $C^pS_A$.getParents(p.$C_a$)
          parent$C_b$ = $C^pS_B$.getParents(p.$C_b$)
          same = isSameParent(parent$C_{a,}$ parent$C_{b)}$
                    If same Then $Sim_{CS}$.add(p)
          Next
Function isSameParent(Vector $V_a$, Vector $V_b$): Boolean
          {match=False
           For each $p_a$ in $V_a$
             For each $p_b$ in $V_b$
                    If $p_a = p_b$ Then
                              {match= True;
                              Break ;}
      Next
           Return match
          }
End
```

Figure 3: A slice of pseudo code for identifying contextual similarity

In Equation (20), the symbol $\approx^2$ represents contextual of the 2nd level similarity as defined in Equation (8). The contextual similarity is based on taxonomic positions of a_i and b_j. To measure this similarity, it is necessary to measure the similarity between their respective parents. A segment of algorithm of the identifying process of contextual similarity is given in Figure 3.

3.4 IRS - Identifying Role-based Similarity Phase

In this phase, the role based similarity, as defined in Equation (9), is measured between two contextually similar concepts. Figure 4 shows a segment of algorithm of the identifying process of the role-based similarity. Similarly, to measure *SR* we acquire the roles of each concept. The roles of each concept of *A* and *B* ontologies are separately acquired in two vectors i.e. C^RS_A and $C^RS_{B,}$ formally defined as:

$$C^RS_A = \{(a_i, (r_i, r_{i+1}, . . ., r_n,)) \mid \forall\ a_i, r_i \in A \wedge\ r_i \text{ isRoleOf}(a_i)\} \quad (21)$$

$$C^RS_B = \{(b_j, (r_j, r_{j+1}, . . ., r_n,)) \mid \forall\ b_j, r_j \in B \wedge\ r_j \text{ isRoleOf}(b_j)\} \quad (22)$$

C^RS_A, and C^RS_B (see Equation (21) – (22)) are populated in the role-acquisition process and Sim_{CS}

(see Equation (20)) is populated in the previous phase, both are the input of the process and Sim_{RS} - a set containing pairs of similar concepts based on their roles (defined in Equation (23)), is the output of this phase.

$$\text{Sim}_{\text{RS}} = \{(a, b, \text{DoS}, \text{SR}) \mid \forall\ ((a, b) \in \text{Sim}_{\text{CS}}) \wedge (a \approx^{3} b)\} \qquad (23)$$

```
Algorithm: Measuring of role-based similarity
Input: (i) C$^R$S$_A$, C$^R$S$_B$ Vectors
           (ii) Sim$_{CS}$ Vector
Output: Sim$_{RS}$ - a vector containing pairs of role-
based similar concepts.
Begin
For each p in Sim$_{cs}$
rC$_a$= C$^R$S$_A$.getRoles(p.C$_a$);rC$_b$=
C$^R$S$_B$.getRoles(p.C$_b$)
T = countSame(rC$_{a,\ r}$C$_{b)}$
DoS = T / ((rCa.size() + rCb.size()) – T)
SR = computeSR (T, rCa.size(), rCb.size())
temp.Ca = p.Ca;   temp.Cb = p.Cb
temp.SR = SR; temp.DoS = DoS; Sim$_{RS}$.add(temp)
Next
End Sub
Function countSame(Vector V$_a$, Vector V$_b$): Return
Boolean
{same = 0
  For each r$_a$ in V$_a$
 For each r$_b$ in V$_b$
 If r$_a$ = r$_b$ Then {same = same +1;   Next
 Next;  Return same}
End Function
Function computeSR(Integer T, integer n, integer
m): Return Char {
If (t = n) && (n – m = 0) Then Return '='
Else If (t = m) && (n – m > 0) Then Return '≥'
Else If (t = n) && (n – m < 0) Then Return '≤'
                          Else Return 'X' }
End Main
```

Figure 4: A slice of pseudo code for identifying role-based similarity

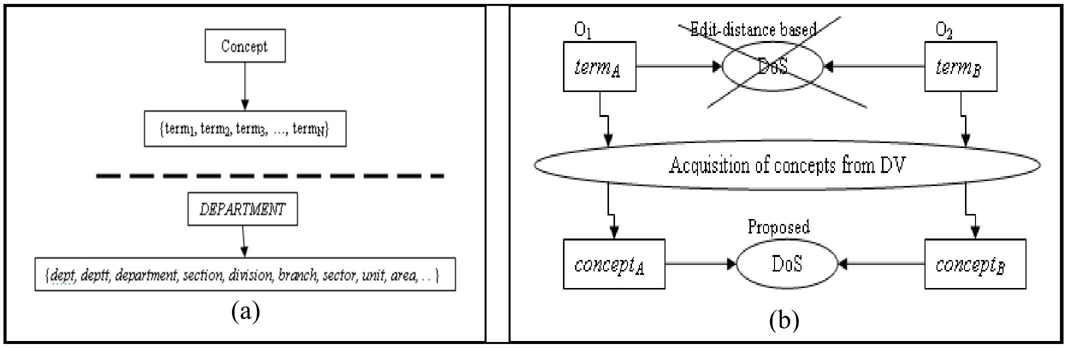
In Equation (23), the symbol $\approx^{3}$ represents the role-based or the 3rd level similarity as defined in Equation (9). In order to identify the 3rd level similarity of contextual similar concepts short listed in the previous phase, we need to acquire their roles from their respective ontologies.

In Table 2, we give a comparison between the existing techniques and proposed technique; *SM*, *DoS* and *SR* represent Similarity Measurement, Degree of Similarity and Semantic Relation, respectively. The explicit semantic similarity measurement is the key point of the proposed technique. According to theme of Semantic Web, the short comings of the current web can be overcome by formalizing explicit semantics of web-contents using ontologies. However, ontologies may themselves suffer from the explicit semantic heterogeneity problem when their lexically and contextually similar concepts have different or overlapped explicit semantics. In order to resolve such type of heterogeneity, the similarity measurement based on explicit-semantics is essential.

Table 2: Existing techniques vs. proposed technique

	Parameters	**Existing Techniques**	**Proposed Technique**
i	**Explicit-semantics based SM**	Not supported	Supported
ii	**Lexical SM**	- Terms are compared;- *DoS* is computed through string-based techniques(edit-distance, prefix, suffix and n-gram)	- Concepts are compared. - *DoS* is computed from granularities and explicit-semantics of concepts

iii	**Linguistic-semantics based SM**	Supported	Domain specific semantics of concepts
iv	**Contextual SM**	Both the optional and mandatory characteristics are considered	Only mandatory characteristic with different criterion is considered
v	**Output of overall SM**	Pairs of similar concepts with either *DoS* or *SR*	Pairs of similar concepts with both *DoS* and *SR*
vi	**Matching Strategy**	Individual Matching	Integrated and layered matching


Concept
{term1, term2, term3, ..., termN}
DEPARTMENT
{dept, deptt, department, section, division, branch, sector, unit, area, . . }
O1
Edit-distance based
DoS
O2
termA
termB
Acquisition of concepts from DV
Proposed
DoS
conceptA
conceptB

(a)

(b)

Figure 5. The *DoS* through edit-distance based formula and through proposed formula

A concept is represented by a set of terms, including its synonyms such as shown in Figure 5 (a). The existing techniques, as summarized in Table 1, use edit-distance based formula to compute *DoS* between two concepts.

In some cases, the edit-distance based *DoS* can be incorrect such as the pair (O1:dept, O2:department) shown in Figure 5(a), is declared dissimilar when edit-distance based DoS measurement formula is used Similarly, some pairs of dissimilar concepts are declared as similar pairs such as (*Software Design*, *Software Designer*) and (*System Analyst*, *System Analysis*) because the edit-distance based *DoS* between concepts of these pairs are 0.86 and, 0.85 respectively. In proposed technique, the measurement of *DoS* is performed on concepts themselves represented by the terms given in ontologies. The measurement process of *DoS* is accomplished through domain vocabulary (*DV*), as shown in Figure 5 (b).

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

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

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

We have empirically observed that within a certain domain, the lexically dissimilar concepts are always contextually dissimilar. Similarly the contextually dissimilar concepts are always explicit semantically dissimilar. That is, there is no need to measure the contextual similarity between lexically dissimilar concepts. And, there is no need to measure the explicit-semantics similarity between contextually dissimilar concepts. Secondly, the direct measurement of contextual similarity without measuring the lexical similarity may produce inaccurate result. This suggests that, if the similarity measurement is performed in some integrated and layered fashion to enable the measurement process more efficient. Most of the existing techniques follow the individual matching. The individual matching strategy reduces the efficiency of overall similarity computing process because of the maximum input for all matchers. For example, there are N numbers of candidate pairs whose similarities are to be measured. In individual matching strategy each matcher gets same and the maximum input i.e. N, whereas, in integrated and layered strategy the input of second and third matchers are N_1 (where $N_1 < N$) and N_2 (where $N_2 < N_1$) number of pairs respectively. That is, the input of 2nd Matcher of proposed technique is less than the input to the second matcher of existing techniques and same is the case with third matchers of proposed and existing techniques. Furthermore, the 1st level matcher used in proposed technique, identifies similarity between input terms, based on the actual concepts represented by those terms whereas the lexical matcher, used in existing techniques, measures similarity through string-based approaches.

4. Case Studies

We evaluate the proposed technique through case studies targeting its objectives that are given earlier. The Education and the Business domains have been taken as sample domains for testing the working of the proposed technique. We take *Software Development Organization (SDO)* from Business domain and *University* from Education domain. From these two domains, different pairs of ontologies are chosen as the input ontologies to the proposed technique. We have implemented the proposed technique in Java language by using an integrated development environment - NetBeans IDE 6.1 (NetBeans, 2009). In order to load and parse ontologies, OWL API (Bechhofer et al., 2003; Horridge et al., 2007) has been used.

The ontologies of SDO, which we have selected, they mainly concentrate on human resources and their roles, i.e., the intellectual concepts and their interactions with non-intellectual concepts. A software organization has different categories of the intellectual concepts such as technical and non-technical human resources. The category of technical human resources is further divided in different teams such as Analysis-team, Design-team, Implementation-team, SQA-team, Supplemental-team and Deployment-team. There are different concepts in each team such as Analyst, Use-Case Engineer, Software Engineer, Programmer, Coder, SQA-Engineer, Technical-Writer, Librarian, and Project Manager. They work on different projects, and each project has many different modules. These intellectual concepts are commonly used in different software development organizations with same, overlapped or different roles. In order to manually trace the proposed technique, we have taken a subset of commonly used roles by the intellectual concepts of these ontologies, which are listed in Figure 6. The list of sample concepts of the first input ontology *dataSoft.owl* is shown in Table 3. For the sake of simplicity, we have chosen only those concepts which are contextually similar. The domain vocabulary includes the concepts of this ontology.

(r1)	Analyze Hardware Requirements	(r2)	Analyze Software Requirements
(r3)	Analyze Functional Requirements	(r4)	Analyze Non Functional Requirements
(r5)	Analyze Cost Benefit	(r6)	Design Database
(r7)	Design Algorithms	(r8)	Design Reports
(r9)	Design Input Screens	(r10)	Design Structure
(r11)	Design Graphics	(r12)	Design Web Pages
(r13)	Implement Database	(r14)	Implement Algorithm
(r15)	Implement Reports	(r16)	Implement GUI
(r17)	Implement Structure	(r18)	Write Requirements Specifications

(r19)	Write Design Documents	(r20)	Write Code Documents
(r21)	Test Functional Requirements	(r22)	Test Non Functional Requirements
(r23)	Test Procedures	(r24)	Tune Database
(r25)	Backup Database	(r26)	Cost Management
(r27)	Resource Management	(r28)	Define standard operating procedures
(r29)	Change Management	(r30)	Write User Manual
(r31)	Software configuration control	(r32)	Storing final released products
(r33)	Developing a test plan for the project	(r34)	Allocating database resources to projects
(r35)	Compiling source code/linking/building	(r36)	Defining user profiles
(r37)	Creating test baselines	(r38)	Ensuring Inter-group coordination
(r39)	Deploying applications in virtual machine	(r40)	Ensuring successful project closure
(r41)	Establishing SCCB and SCRB for projects	(r42)	Ensuring SQA activities
(r43)	Faxing, mailing, shipping	(r44)	Ensuring the security of project databases
(r45)	Handling and maintaining the store	(r46)	Identification of project based SCM tool(s)

Figure 6. A subset of roles in software development organization

4.1 List of Concepts of First Input Ontology

Table 3. A sample slice of intellectual concepts form A ontology

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

4.2 List of Concepts of Second Input Ontology

The ontology *ridos.owl* is chosen as the second input ontology. This ontology is also considered while populating the domain vocabulary. A subset of its concepts is shown in Table 4 .

Table 4. A sample slice of intellectual concepts form B ontology

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

Table 5. A slice of role-based similar concepts with a threshold-value

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

In the second case study, we take *csuet.owl* and *lcwu.owl* as ontology *A* ontology *B,* respectively. The semantic relation between a pair of concepts has been computed based on their respective granularities.

The sample concepts that are taken from the ontology *A* are*:* (a_1) *Project,* (a_2) *ITConsultant,* (a_3) *Director,* (a_4) *Manager,* (a_5) *UnderGradStudent,* (a_6) *Convener,* (a_7) *Course,* (a_8) *Professor,* (a_9) *Quiz,* (a_{10}) *Workshop,* (a_{11}) *NationalConference,* (a_{12}) *ResearchCentre,* (a_{13}) *PostGradStudent,* (a_{14}) *Person,* (a_{15}) *Deptt.* The sample concepts that are taken from the ontology *B* are taken: (b_1) *TermProject,* (b_2) *Consultant,* (b_3) *Director,* (b_4) *SupportManager,* (b_5) *ConvenerAdmission,* (b_6) *Student,* (b_7) *Professor,* (b_8) *PostGradCourse,* (b_9) *Workshop,* (b_{10}) *Conference,* (b_{11}) *ResearchCentre,* (b_{12}) *Department,* (b_{13}) *SoftwareEngineer,* (b_{14}) *Person,* (b_{15}) *Employee,* (b_{16}) *Faculty*. The sample pairs are (a_1, b_{12}), (a_2, b_2), (a_3, b_3), (a_4, b_4), (a_5, b_6), (a_6, b_5), (a_7, b_7), (a_8, b_7), (a_9, b_8), (a_9, b_{10}), (a_{10}, b_9) and (a_{11}, b_{10}), respectively.

4.3 Primary Similarity Identification and Measurement: As mentioned earlier, it is the first phase of proposed technique. Here, the pairs of concepts possessing primary similarity are identified.

Input: A= (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, $a_{11,}$ a_{12}); B = (b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, $b_{11,}$ b_{11}).

The output: According to the algorithm for primary similarity identification, given in Figure 2 the following pairs are identified as primarily similar pairs:

Sim^{PS} = {(a_2, b_2, 0.80, ‘≤’), (a_3, b_3,1.00, ‘=’), (a_4, $b_{4,}$ 0.75,’≥’), (a_5, b_6,0.84, ‘≤’), (a_6, b_5,0.80,’≥’), (a_8, $b_{7,}$ 1, ‘=’), (a_9, $b_{8,}$ 0.66,’≥’), (a_{10}, b_9,1, ‘=’), (a_{11}, b_{10},0.86 ‘≤’)}

4.4 Contextual Similarity Identification and Measurement: it is the second phase of proposed technique. Here, the pairs of concepts obtained in previous phase, possessing contextual similarity are identified.

Input: Sim_{PS} and super-concepts of ($a_{2,}$ a_3, a_4, a_5, a_6, a_8, a_9, a_{10}, a_{11}) and super-concepts of ($b_{2,}$ b_3, b_4, b_5, b_6, b_7, b_8, b_9 *and* b_{10}).

Output: According to the algorithm for contextual similarity identification, given in Figure 3 the following pairs are identified as contextually similar pairs:

Sim_{CS} = {(a_2, b_2, 0.80, ‘≤’), (a_3, b_3, 1, ‘=’), (a_4, $b_{4,}$ 0.75, ’≥’), (a_5, b_6, 0.84, ‘≤’), (a_8, b_7, 1, ‘=’), (a_9, b_8, 0.66, ’≥’)}

4.5 Role-based Similarity Identification and Measurement: it is the third and the final phase of proposed technique. Here, the pairs of concepts obtained in previous phase, possessing role-based similarity are identified.

Input: Sim_{CS} and roles of concepts as short-listed in the previous phase i.e. roles *of concepts* ($a_{2,}$ a_3, a_4, a_5, a_6, a_8, a_9) and ($b_{2,}$ b_3, b_4, b_6, b_7, b_8) *respectively.*

Output: According to the algorithm for role-based similarity identification, given in Figure 4 the following pairs are identified as role-based similar pairs:

Sim_{RS} = {(a_3, b_3, 1, ‘=’), (a_5, b_6, 0.84, ‘≤’), (a_8, b_7, 1, ‘=’), (a_9, b_8, 0.66, ‘≥’)}

5. Results: Analysis and Discussion

We have the following observations about the results of *IPS*, *ICS* and *IRS* phases as they have been computed in previous section. These observations are listed as follows:

i) If the 1st level similarity for a pair of concepts is *true*, then it may be *true* or *false* for the next levels of similarities.

ii) If the 1st level similarity for a pair of concepts is *false*, then its 2nd level and 3rd level of similarities are always *false*.

iii) The 3rd level similarity is *null* for a pair of concepts of the non-intellectual concepts possessing 2nd level of similarity.

(iv) There is a role-based similarity between pair of concepts *(a_{13}, b_7),* i.e., *A: PostGradStudent* and *B: Professor*, because both work-on the research-project, and also there is contextual-similarity between these concepts. Same is the case of the pair of concepts *(a_8, b_{13})* i.e. *A:Professor* and *B:SoftwareEngineer* both are working on Project. These pairs are not primarily similar because the main motive behind finding the similarity between concepts is merging, aligning or mapping of two ontologies for the knowledge sharing, therefore, the merging, aligning or mapping of the *PostGradStudent* concept with the *Professor* concept, is not recommended. In the proposed technique, a pair of concepts having no primary similarity is simply discarded.

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

To realize the achievement of the different objectives, as listed before, we compare the results of proposed technique with the results from some existing techniques. The criteria for comparison include the (i) completeness; (ii) correctness and (iii) overall quality of results.

Completeness: The completeness of a similarity identifying technique is just like the precision measures used in information retrieval (Trojahn et al., 2008; Euzenat, 2007; Ehrig & Euzenat, 2005). It is the ratio of correct number of pairs found divided by the total number of pairs found. Let *totalPairsFound* be the total number of pairs found in which *CorrectPairsFound* number of pairs are correct, such as *totalPairsFound >= CorrectPairsFound*, then the completeness can be formally written as:

$$\text{Completeness} = \frac{Correct\ \_\ Pairs\ \_\ Found}{Total\ \_\ Pairs\ \_\ Found} \quad (24)$$

Correctness: The correctness of a similarity identifying technique is just like the recall measures used in information retrieval (Trojahn et al., 2008; Euzenat, 2007; Ehrig & Euzenat, 2005). The correctness is the ratio of correct number of pairs found, divided by the expected number of correct pairs. Let $Correct\_Pairs\_Expectea$ be the total number of correct pairs expected and $Correct\ \_\ Pairs\ \_\ Found$ number is of correct pairs found by a technique such as $Correct\_Pairs\_Expectea >= Correct\ \_\ Pairs\ \_\ Found$, then the correctness can be formally written as

$$\text{Correctness} = \frac{Correct\ \_\ Pairs\ \_\ Found}{Correct\ \_\ Pairs\ \_\ Expected} \quad (25)$$

Overall Quality of Result: The overall quality (OQ) of result is based on correctness and completeness of result. It is computed just as f-measure (Trojahn et al., 2008; Euzenat, 2007; Ehrig & Euzenat, 2005), used in information retrieval.

$$OQ = 2 * \frac{Completeness * Correctness}{Completeness + Correctness} \quad (26)$$

Through layered strategy, the output of first layer is used as input for the second layer and so on, whereas the output of first layer is set of pairs of concepts having primary similarity while all other concepts are discarded in the output. This means that the input to second layer is a short list of concepts instead of all concepts which reduce a reasonable execution-time for 2nd level of similarity identification. Similarly the concepts shorted-listed in second layer are input to third layer. Therefore, the overall execution-time of proposed technique is comparatively short.

Test Cases for Evaluating Performance

We have taken four pairs of ontologies as shown in Table 6 to evaluate the completeness, correctness and overall quality of results of proposed technique up to second level of similarity. And, then it is followed by the evaluation of the role-based similarity i.e. the 3rd level similarity, based on new criterion. Comparisons of results are then made with expected results and with the results of existing matching techniques used in different tools and systems

5.1 Evaluating Performance with 1st Test Case

Sample input pairs: 37; Pairs of similar concepts (expected): 25; Similar pairs (out of 25) with different terms: 10. With respect to test case 1, the results from proposed technique (SIMTO) and from some existing techniques are compared with respect to their completeness, correctness and overall quality. The graphical representation of comparison is also given in the Figures 7, 8 and 9 respectively. A comparative improvement in result of proposed technique, with respect to completeness is realized.

Table 6: Test-cases for evaluating performance of different techniques

TestCase	Ontologies	Input Pairs	Similar Pairs (expected)	Similar Pairs (with different terms)
1	A1, B_1	37	25	10
2	A2, B_2	40	22	5
3	A_3, B_3	28	12	2
4	A4, B_4	25	15	12

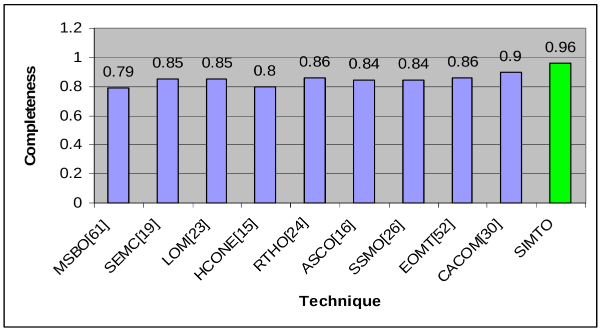


Figure 7. Completeness wise comparison of results with respect to first test case

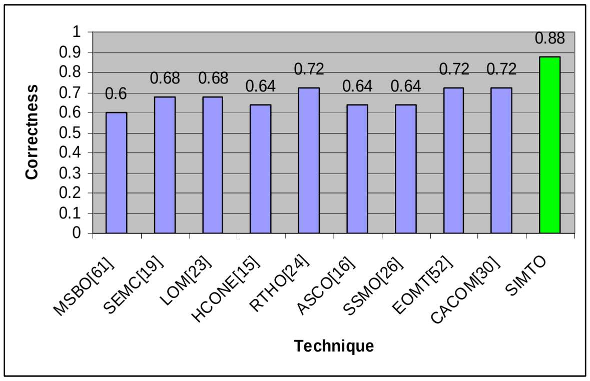


Figure 8. Correctness wise comparison of results with respect to first test case

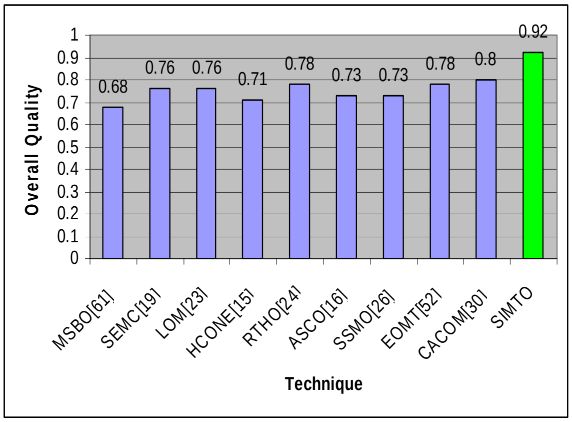


Figure 9. Overall quality wise comparison of results with respect to first test case

5.2 Evaluating Performance with 2nd Test Case

Sample input pairs: 40; Pairs of similar concepts (expected): 22; Similar pairs (out of 22) with different terms: 5. With respect to test case 2, the results from *SIMTO* and from some existing techniques are compared with respect to their completeness, correctness and overall quality. The graphical representation of comparison is also given in the Figures 10, 11 and 12 respectively. It has observed that when the number of similar pairs having different names, decrease, the completeness of results increases. Furthermore, the result of proposed technique, with respect to completeness is better than the results of existing techniques.

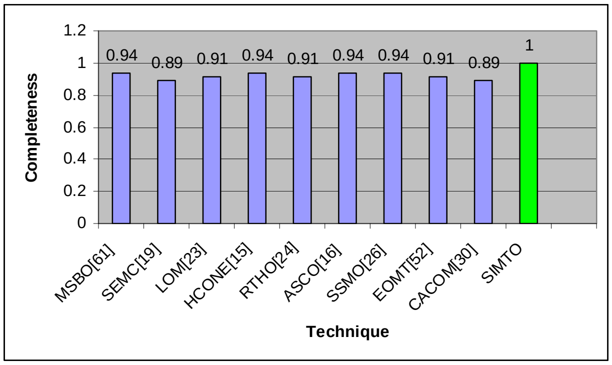


Figure 10. Completeness wise comparison of results with respect to second test case

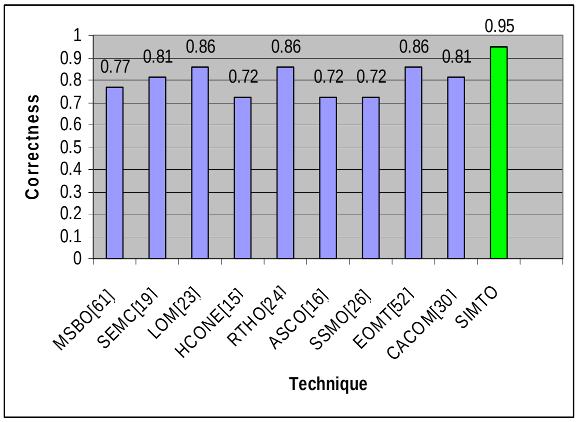


Figure 11. Correctness wise comparison of results with respect to second test case

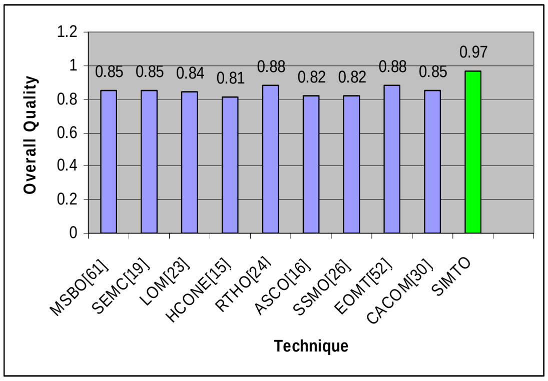


Figure 12. Overall quality wise comparison of results with respect to second test case

5.3 Evaluating Performance with 3rd Test Case

Sample input pairs: 28; Pairs of similar concepts (expected): 12; Similar pairs (out of 12) with different terms: 2. With respect to test case 3, the results from *SIMTO* and from some existing techniques are compared with respect to their completeness, correctness and overall quality. The graphical representation of comparison is given in the Figures 13, 14 and 15 respectively. An improvement in result of proposed technique, with respect to completeness, correctness and overall quality, is realized in comparison. It has also observed that when the number of similar pairs having different names, decreases, the completeness of results increases. Furthermore, the result of proposed technique, with respect to completeness, correctness and overall quality is better than the results of existing techniques.

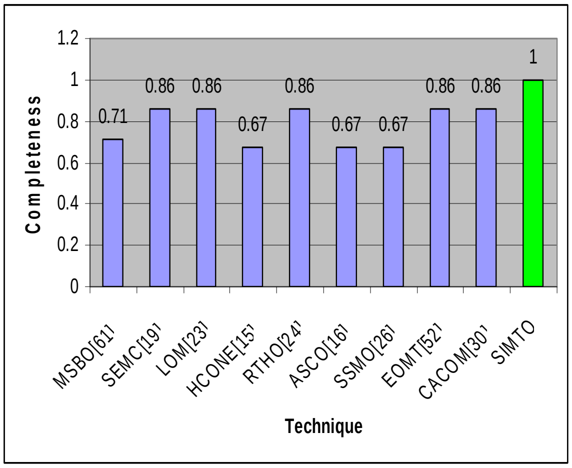


Figure 13. Completeness wise comparison of results with respect to third test case

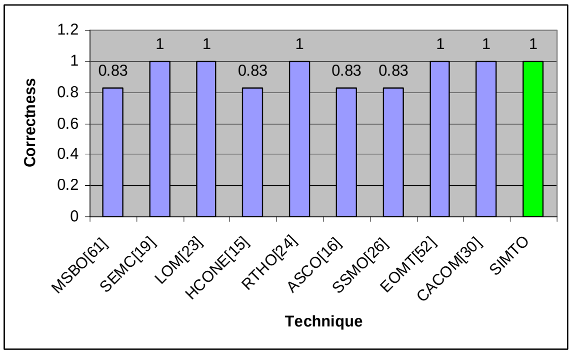


Figure 14. Correctness wise comparison of results with respect to third test case

.

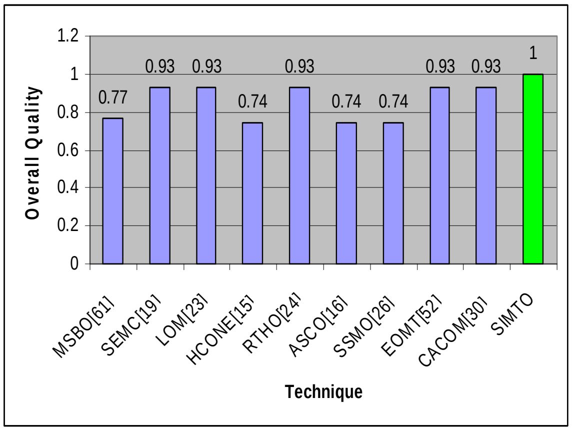


Figure 15. Overall quality wise comparison of results with respect to third test case

5.4 Evaluating Performance with 4th Test Case

Sample input pairs: 25; Pairs of similar concepts (expected): 15. Similar pairs (out of 15) with different terms: 12. With respect to test case 4, the results from *SIMTO* and from some existing techniques are compared with respect to their completeness, correctness and overall quality. The graphical representation of comparison is given in the Figures 16, 17 and 18 respectively. It has observed that when the number of similar pairs having different names increase, the completeness of results decreases. There is a considerable decrease in correctness of results from existing techniques particularly the techniques excluding the linguistic similarity of terms. Furthermore, the overall qualities of results are also badly affected. However, the result of proposed technique, with respect to completeness is better than the results of existing techniques. The comparisons between results from some current techniques and from proposed technique (up to 2nd level of similarity) are shown in Figures 7-18.

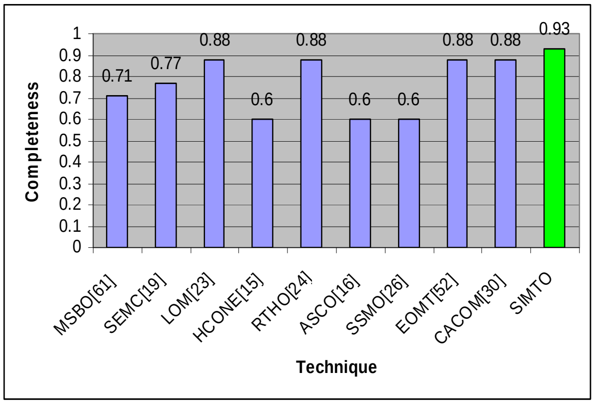


Figure 16. Completeness wise comparison of results with respect to fourth test case

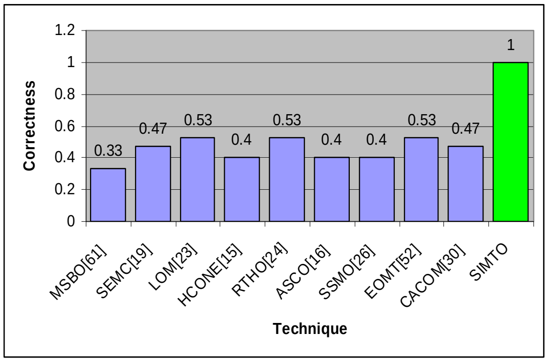


Figure 17. Correctness wise comparison of results with respect to fourth test case

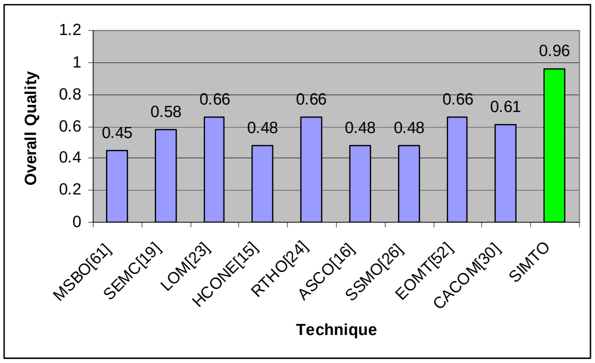


Figure 18. Overall quality wise comparison of results with respect to fourth test case

Furthermore, a comparison between results from proposed technique with the new criterion (i.e. role-based similarity) and expected results has also made in Figure 19. We examined more than ten ontologies of different software houses for evaluating the new criterion of proposed technique. The results are verified by respective domain experts and are declared satisfactory.

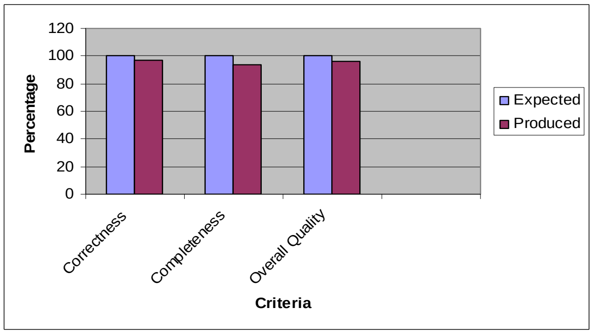


Figure 19. Role-based similarity: expected results vs. produced results

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

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

Overall results of existing techniques heavily rely on the heterogeneity of terms as shown in Table 7; higher the number of concepts represented with different terms, lower the completeness and overall quality of results will be.

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

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

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

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

6. Conclusion and Future Directions

In this paper a semi-automatic, integrated and layered technique has been presented for identification and measurement of similarity between two ontologies. The proposed technique is based on the innovative theme of the semantic web. The proposed technique is not only helpful in different

.Table 7: Comparison of results related to Heterogeneous Pairs of Similar Concepts (HPoSCs)

TestCase	HPoSCs	Identified (Existing Techs.)	Identified (Proposed Tech.)	Correctness (Existing Techs.)	Correctness (Proposed Techs.)
1	10	3	7	**0.30**	**0.70**
2	5	1	4	**0.20**	**0.80**
3	2	2	2	**1.00**	**1.00**
4	12	2	11	**0.17**	**0.92**

ontology integration operations such as merging, mapping, alignment and querying but also in engineering new ontologies

Identification and measurement of similarity between the ontologies is a mandatory pre-requirement of various reuse operations of ontologies such as merging, mapping and alignment. It is also a mandatory requirement for engineering new ontologies by assembling exiting ontologies or components of ontologies. Although the proposed similarity identification technique uses, as core, the innovative ideas of semantic web however essential modifications related to the issues and trends specific to the similarity between concepts of ontologies has been made. The proposed technique upgrades similarity measurement criteria, from terms to concepts, from linguistic semantic to explicit semantic and from all taxonomic characteristics of concepts to mandatory and optional characteristics. In addition we introduced the concepts of similarity levels: primary similarity or 1st level similarity; contextual similarity or 2nd level similarity and the role-based similarity or 3rd level similarity.

We conclude the research result as follows:

- Similarity measurement techniques used for database schemas and XML schemas are not well suited for identifying and measuring of similarity between ontologies schemas.
- The role of domain-specific vocabulary is vital in measurement of similarity between ontologies.
- Primary similarity measurement is the prerequisite for the contextual similarity measurement whereas the contextual similarity measurement is the prerequisite for the role-based similarity identification.
- For a pair of concepts, the degree of similarity and semantic relation are complements to each others.
- It is difficult to get hundred percent correct and complete results due to the lack of standardization in the use of terminologies for concepts and their roles.

As discussed above, the similarity identification is a core and prerequisite task for ontologies integration operations. In addition, this task is also required for ontologies engineering through reuse of ontologies. We plan to work on design and development of methodologies for reuse and integration of ontologies.

Acknowledgement

This research work has been supported by the "Higher Education Commission of Pakistan", and the University of Engineering and Technology, Lahore.

References

1. Alasoud, A., Haarslev,V., & Shiri, N. (2008). An Effective Ontology Matching Technique. *Proceedings of the 17th International Symposium on Methodologies for Intelligent Systems (ISMIS'08), LNAI 4994, 585–590.*
2. Aleksovski, Z., Kate, W., & Harmelen, F. (2006). Exploiting the structure of background knowledge used in ontology matching. *Proceedings of International Workshop on Ontology Matching collocated with the 5th International Semantic Web Conference, pp. 13-24 , USA.*
3. Aleksovski, Z., Klein, M., Kate, W., & Harmelen, F. (2006). Matching Unstructured Vocabularies Using Background Ontology. *Proceedings of 15th International Conference on Knowledge Engineering and Knowledge Management Managing Knowledge in a World of Networks, 182-197, Czech Republic.*
4. Bechhofer, S., Lord, P., & Volz, R. (2003). Cooking the Semantic Web with the OWL API. *2nd International Semantic Web Conference, ISWC, Sanibel Island, Florida.*
5. Bouquet, P., Serafini, L., & Zanobini, S., (2003). Semantic Coordination: A New Approach and an Application. *LNCS 2870, pp.130-145.*
6. Buccella, A., Cechich, A., & Brisaboa, N. (2005). A Three-Level Approach to Ontology Merging. *MICAI , LNAI 3789, 80 – 89.*
7. Cohen, W., Ravikumar, P., Fienberg, S.(2003). A Comparison of String Distance Metrics for Name-Matching Tasks. *IJCAI-03: 3-78.*
8. Duchateau, F., Bellahsene, Z., & Roche, M. (2007). Context-based Measure for Discovering Approximate Semantic Matching between Schema Elements. *Proceedings of The International Conferences on Research*

Challenges in Information Science (RCIS), Morocco.

9. Ehrig, M., & Euzenat, J.(2005). Relaxed precision and recall for ontology matching. Pr*oceedings of K-Cap 2005 workshop on Integrating ontology, Banff (CA)* 25-32.
10. Erhard, R., & Philip, B.A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal, 10(4), 334-350, Springer Berlin*.
11. Euzenat, J. (2007). Semantic Precision and Recall for Ontology Alignment Evaluation. *In proceedings of International Joint Conference on Artificial Intelligence, 348-353*.
12. Giunchiglia, F., Yatskevich, M., & Pavel Shvaiko, P. (2007). Semantic Matching: Algorithms and Implementation. *Journal on Data Semantics.* 9 (2007),1-38, Germany.
13. Giunchiglia, F., Yatskevich, M., Shvaiko, P. (2007). Semantic Matching: Algorithms and Implementation. *LNCS Journal on Data Semantics 9 (2007), 1-38*, Germany.
14. González R. G. (2005). A Semantic Web approach to Digital Rights Management. *Ph.D. Thesis, Universitat Pompeu Fabra, Barcelona*.
15. Hariri, B., Abolhassani, H., Khodaei, A.(2006). A new Structural Similarity Measure for Ontology Alignment. *In proceedings of the 2006 International Conference on Semantic Web & Web Services, pp.36-42 , USA*.
16. Hauswirth, M., & Maynard, D. (2007). Knowledge web 2.2: Heterogeneity in the semantic web. *Technical report, NoE Knowledge Web project*.
17. Horridge, M., Bechhofer, S., & Noppens, O. (2007). Igniting the OWL 1.1 Touch Paper: The OWL API. *OWLED 2007, 3rd OWL Experienced and Directions Workshop, Innsbruck, Austria*.
18. Janowicz, K., Raubal, M., Schwering, A., & Kuhn, W. (2008). Semantic Similarity Measurement and Geospatial Applications. *Transactions in GIS, 12(6), 651-659*.
19. Jeong, B., Lee, D., Cho, H., & Lee, J. (2008). A novel method for measuring semantic similarity for XML schema matching. *Expert Systems with Applications, 34(3), 1651–1658, Elsevier*.
20. Kotis, K., & Vouros, G.A. (2004). The HCONE Approach to Ontology *Merging. Proceedings. of the First European Semantic Web Symposium, LNCS 3053, Springer, 137-151*.
21. Lambrix, P., & Tan, H. (2006). SAMBO - A System for Aligning and Merging Biomedical Ontologies. *Journal of Web Semantics, 4(3), 196-206.*
22. Lee, T. B., Hendler, J. and Lassila, O. (2001). The SemanticWeb. Scientific America, 284(5),34-43.
23. Lewandowski, D. (2008). The Retrieval Effectiveness of Web Search Engines: Considering Results Descriptions. *Journal of Documentation, 66(6), 915-937*.
24. Maedche, A., & Staab, S. (2002). Measuring similarity between ontologies. *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (EKAW), 251–263, Spain*.
25. McGuinness, D., Fikes, R., Rice J. & Wilder, S. (2000). An Environment for Merging and Testing Large Ontologies. *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning. 483-493*.
26. Melnik, S., Garcia-Molina, H., & Rahm, E. (2002). A Versatile Graph Matching Algorithm and Its Application to Schema Matching. *Proceedings of International Conference on Data Engineering (ICDE), 117-128, USA*.
27. NetBeans IDE 6.1 (2009). http://www.netbeans.org/community/releases/61/ , Retrieved on October 20, 2009.
28. Noy, N., & Musen, M. (2001). Anchor-prompt: using non-local context for semantic matching. *Proceedings of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI), 63–70, USA*.
29. Pedersen, T., Patwardhan, S., & Patwardhan, S. (2004). WordNet::Similarity – Measuring the Relatedness of Concepts. *Proceedings. of 19th National Conference on AI, San Jose, CA*.
30. Sherman, C., & Price, G. (2001). The Invisible Web: Uncovering Information Sources Search Engines Can't See. *CyberAge Books*.
31. Shvaiko, P., & Euzenat, J. (2005). A Survey of Schema-based Matching Approaches. *Journal on Data Semantics, 146-171*, *Germany*.
32. Shvaiko, P., & Euzenat, J. (2008). Ten Challenges for Ontology Matching. *In Proceedings of the 7th International Conference on Ontologies, Databases, and Applications of Semantics*.
33. Shvaiko, P., & Euzenat, J. (2009). Ontology Matching web-site;

http://www.ontologymatching.org/. Retrieved on October 4, 2009.

34. Trojahn, C., Moraes, M., Quaresma, P., & Vieira, R.(2008). A Cooperative Approach for Composite Ontology Mapping. *Journal on Data Semantics X, LNCS 4900, 237–263*.

35. Uschold, M. (2002) A semantic continuum on the semantic web. *The Knowledge Engineering Review, 17(1), 87-91*.

36. Uschold, M. (2003) Where Are the Semantics in the Semantic Web?. *AI Magazine 24(3): 25-36*.

37. Visser, P., Jones, D., Bench-Capon, T., Shave, M. An Analysis of Ontology Mismatches; Heterogeneity versus Interoperability. *AAAI Spring Symposium on Ontological Engineering*, 1997.

38. W3C; The World Wide Web Consortium; http://www.w3.org

2/1/2010