Evaluating Suitability for Replacement of an Integrated Software Component

Boonserm Kulvatunyou[1], Buhwan Jeong[2], Hyunbo Cho[2], Nenad Ivezic[1], and Albert Jones[1]

[1]Manufacturing Systems Integration Division
National Institute of Standards and Technology (NIST)
MS 8263, Gaithersburg, MD 20899, USA
{serm, nivezic, jonesa}@nist.gov

[2]Department of Industrial and Management Engineering
Pohang University of Science and Technology (POSTECH)
San 31, Hyoja, Pohang, 790-784, South Korea
{bjeong, hcho}@postech.ac.kr

Abstract

An evaluation of the compatibility of a software component is based typically only on a vendor's presentation. This means that end users select a software component only based on high-level matches of functional requirements. This can result in an underestimation of the actual cost and effort required for integration of the new software component with existing components. In this paper, we provide a suitability measure that can help determine the actual integration efforts. The measure, which is built upon typical semantic similarity measures, attempts to quantify compatibility by focusing on information exchange requirements.

1. Introduction

Consider the following common scenario. A company decides to replace a software component that is integrated with other software components in the enterprise. The original component provider may be out of business, does not support that particular version of the software any longer, or may have a newer version that is deemed to be too expensive. The company decides to find another software component, which has the required functionality, from another vendor.

In this situation, the enterprise IT manager has to make a selection for a new replacement. However, the situation is complicated because this replacement must meet both functional requirements and connectivity requirements. Figure 1 illustrates this situation. The company has an Inventory Visibility (IV) system that is already integrated with its ERP system and has the necessary Web interfaces – one for the company and one for the supplier. The IV system can provide status updates to the visualization software and manage the inventory levels based on a specific inventory management policy.

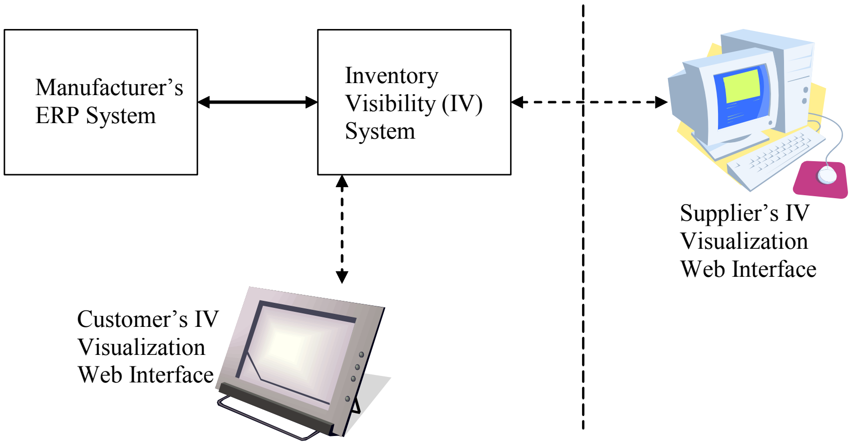
Manufacturer's ERP System
Inventory Visibility (IV) System
Supplier's IV Visualization Web Interface
Customer's IV Visualization Web Interface

Figure 1: A software component connectivity scenario

Since ERP typically do not provide these capabilities, it is common for the ERP and the IV systems to be separate software components provided by different software companies [1]. Therefore, an integration interface exists between the ERP and IV system – indicated by the bold-solid arrow connection in Figure 1. This also implies that a mapping between the corresponding information models also exists. Figure 2 shows part of such a mapping. The most suitable software replacement should have an information model similar to those in the IV system as well as in the ERP system.

The IV system has the functionality needed to support a Min/Max inventory control policy – the inventory level must be maintained between the MinQuantity and the MaxQuantity. There is inventory data in the ERP system to support only some of this functionality. The information object is called QuantityOnHand in the IV system and Inventory in the ERP system. It is important to notice that there is no obvious map between the fields MinQuantity and MaxQuantity in the QuantityOnHand model to any field in the Inventory model, since these fields are required only to implement a Min/Max inventory control policy (this functionality is not available in the ERP system).

In this research, we look at how an IT manager can make a better decision when an integrated software component needs to be replaced. The objective is to propose a suitability measure that can not only identify a potential integration problem – like the one shown above – but also indicate the magnitude of that problem. Therefore, we seek to quantify suitability from an information exchange perspective. The idea is that the more suitable the software replacement the less costly the integration of that software into the enterprise system.

In the next chapter, we provide an overview of the semantic similarity measures that are bases for the suitability measure. Then, we introduce the suitability measure and provide conclusion and future work.

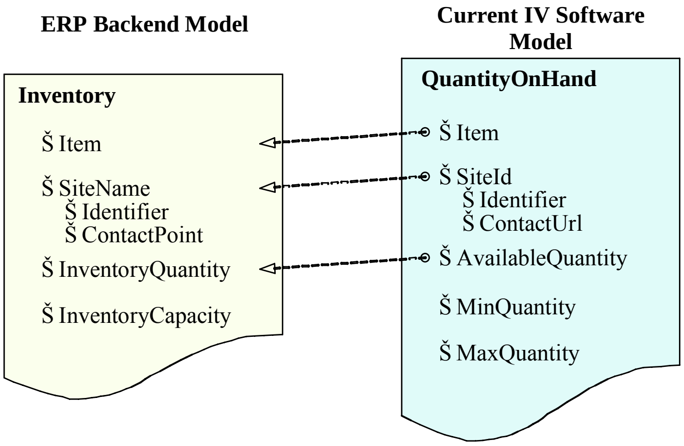

Figure 2: An exemplary mapping of data between the ERP and the IV systems

2. Overview of Semantic Similarity Measure

The basis for our suitability measure is the "semantic similarity" of the information exchange schemas implemented in the two software applications. Shvaiko suggested that schema matching could be categorized by the result it produces. The result of the syntactic-based matching is a value in the interval [0, 1], while the result of the semantic-based matching is a relationship such as subclass, equivalent, and so on. He also suggested another kind of organization based on the hierarchy of approaches and techniques used to compute the match. The hierarchical structure he suggested includes heuristic or formal techniques, implicit or explicit element information, lexicons or precompiled thesaurus, automated- or human-based reasoning, and, finally, first-order or description logic [2].

To support the suitability measure described in the next section, we investigated three types of similarity measures, which are distinguished by the three types of information used to compute the measure: lexical, structural, and logical categories [3]. The lexical approach quantifies the commonality between individual element names using purely lexical information. Commonly used lexical similarity measures include *affix*, *n-gram*, (weighted) distance-based [4] [5], word sense-based [6], and information content-based metrics [7]. The structural approach quantifies the commonality between elements by taking into account the lexical similarities of multiple, structurally related sub-components of these terms (e.g., child elements, child attributes). It is usually a more conservative measure than the lexical one, because it looks beyond the individual labels and their definitions to the "local" context. Commonly used structural measures include node, edge, and/or path matching, tree edit distance (TED) [8], (weighted) tag similarity [9], weighted tree similarity [10], and a Fourier transformation-based approach [11]. The logical approach quantifies the commonality of properties/constraints that limit element definitions beyond the lexical and structural aspects such as type, cardinality, and so forth. Some exemplary approaches include DL-based [12], SAT-based [13], and machine learning-based ones [14]. The logical approach can be viewed as a special case of the structural approach. However, we treat it separately because it is the most restrictive and it provides an accurate measure. That is, even if two elements have identical label and structures, their logical similarity value can still be imperfect.

As suggested in [3], we will combine these approaches when computing the suitability measure. Take Figure 3 as an example, where we show the map between the Factory object and the Plant object. Using the lexical similarity measure suggested in [5] and WordNet [15], the similarity values of the (Site, Location) = 0.72 and (Address, Location) = 0.82. However, using the structural approach in [16], which is based on the NamePath, the lexical information, and a harmonic mean, yields a value of 0.89 (see Table 1 for this computation) for the (Site, Location) pair.

Table 2 shows a similar computation for the (Location, Address) pair, which yields a value of 0.46. Based on these computations, the mapping should be between Location in the Plant object and Site in the Factory object.

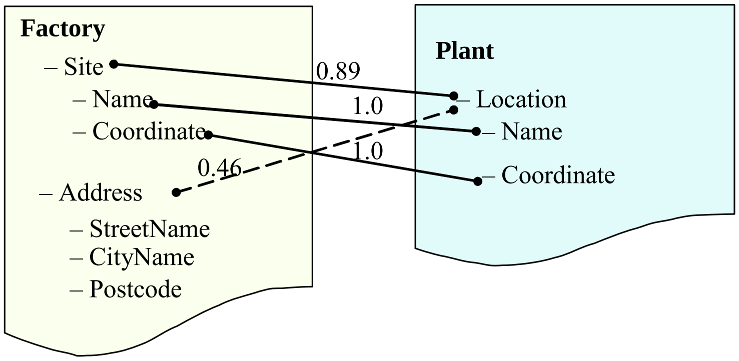

Figure 3: An example for mapping calculation

Table 1: Combined measure between the Site and Location objects

Site’s Children	Location’s Children	Lexical Similarity
Name	Name	1
Name	Coordinate	0.5
Coordinate	Name	0.5
Coordinate	Coordinate	1
Combined Similarity	3 / (1+1+1/0.72)	0.89

Table 2: Combined measure between the Address and Location objects

Address’ Children	Location’s Children	Lexical Similarity
StreetName	Name	0.5
StreetName	Coordinate	0.25
CityName	Name	0.5
CityName	Coordinate	0.25
PostCode	Name	0.33
PostCode	Coordinate	0.3
Combined Similarity	3 / (1/0.5+1/0.3+1/0.82)	0.46

3. Suitability Measure

As described in the previous section, our suitability measure is based on information compatibility between data schemas only. That is, we are concerned with measuring the degree to which the new component consumes (or produces) the same data/messages that the existing component does, and how difficult it is to integrate the new component to the backend system.

We show two alternatives to compute this measure using a relaxation of edge equality, called hard edge equality and weak edge equality. Before detailing the measure, let us revisit the scenario in Figure 2, with a candidate replacement component added as shown in Figure 4. In this example, the candidate software component uses an information model called ProductAvailability.

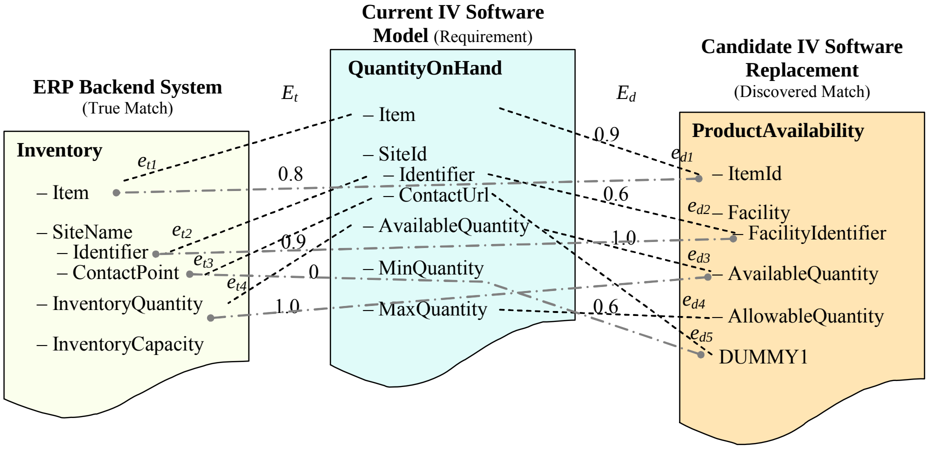

Figure 4: A suitability scenario using hard edge equality

3.1 Hard Edge Equality Metric

The suitability metric is a measure that indicates the degree of match between the data models in the candidate software replacement and the ones the current software component uses to exchange information with the backend system. That is we are treating the data model used by the current software as a requirement, the data model in the candidate software replacement as a discovered match, and the data model in the backend system as a true match. This is illustrated in Figure 4. Next we describe how these relationships are exploited.

We assume that all data models can be represented in an XML (Extensible Markup Language) tree-like representation. Consequently, each data field can be addressed using a path representation. Let a set $U = \{u_i\}$, $i = 1, 2, \ldots, n$ be the set of paths addressing each leaf node, u_i, for each field of the requirement U. Similarly, let a set $V = \{v_j\}$, $j = 1, 2, \ldots, m$ be the set for the true matched model, and a set $W = \{w_k\}$, $k = 1, 2, \ldots, p$ be the set for the discovered matched model.

Then, let a set of edge constraints $E_t \subseteq U \times V$, be the actual map from the fields in U to those in V. Similar to E_t, let a set $E_d \subseteq U \times W$, be the map from U to W. We note that the set E_t is known since the integration exists while the set E_d is computed according to some similarity measures. The dashed lines in Figure 4 demonstrate the graphical representation of E_t and E_d. As shown in the figure, there may be some fields in U that have no map to any field in W (e.g., *MinQuantity* field) and, vice versa. We note that the actual mapping may require composition or decomposition of fields using additional arithmetic or string manipulations. In such cases, the edge constraint only captures that there exists relationship between fields (leaving out the composition/decomposition information). In this paper, we further simplify the scenario by assuming that each data field in the requirement corresponds to no more than one field in the true and discovered matches and vice versa. That is, there are no edges in E_t such that $e_{t1} = (u_a, v_b)$ and $e_{t2} = (u_x, v_y)$ where $u_a = u_x$ *and* $v_b \neq v_y$ or where $u_a \neq u_x$ and $v_b = v_y$; and no edges in E_d such that $e_{d1} = (u_a, w_b)$ and $e_{d2} = (u_x, w_y)$ where $u_a = u_x$ *and* $w_b \neq w_y$ or where $u_a \neq u_x$ and $w_b = w_y$. Using these definitions and the distributional similarity metrics typically used in the information retrieval field [17], we can measure the suitability of a candidate software replacement d using *Jaccard* metric as $S_d = | E_t \cap E_d | / | E_t \cup E_d |$ for all u_i in E_t. For the hard edge equality, where the equality between two edges $e_t \equiv e_d$ ($e_t \in E_t$, and $e_d \in E_d$) is defined as follows.

Given $e_t = (u_a , v_b)$ and $e_d = (u_x, w_y)$, two edges are matched (i.e., $e_t \equiv e_d$) if and only if the paths are matched (i.e., $u_a = u_x$ and $v_b \equiv w_y$), where $v_b \equiv w_y$ if and only if $Sim(v_b, w_y) \geq \mu$, where the $Sim(v_b, w_y) \in [0, 1]$ is a similarity measure function between the two paths, and μ is an arbitrary threshold within the range [0, 1] to conclude that $e_i \equiv e_j$. In addition, the penalty for when there is no discovered match for a true match is as follows. For each edge $e_t = (u_a, v_b)$ where there is no edge $e_d = (u_a, w_y)$, a dummy edge (u_a, ω), using a dummy node ω, is added to the set E_d, and $Sim(v_b, \omega) = 0$.

In Figure 4, assume that the set E_d is created by a similarity measure approach and maps have been approximated between *U* and *W*. The true match set E_t already exists since the current IV software and the ERP backend system are integrated. The $Sim(v_b, w_y)$'s are computed for each corresponding u_i using the same similarity measure approach to find E_d. The example, in Figure 4 then provides sufficient information to compute the hard edge equality measure. For the $\mu = 1.0$, $\mu = 0.9$, $\mu = 0.8$, the hard edge suitability metric yields the following values 0.17, 0.33, and 0.75 using the *Jaccard* metric, respectively. Taking the $\mu = 0.9$ as an example, the following table shows how the suitability measure is calculated ($S_d = 2/6 = 0.33$).

Table 3: Suitability measure calculation procedure at μ = 0.9

True match edges	Discover match edges	Addition to $\mid E_t \cap E_d \mid$	Addition to $\mid E_t \cup E_d \mid$	$Sim(v_b, w_y)$	Explanation
e_{t1}	e_{d1}	0	2	0.8	$Sim(v_b, w_y) < 0.9$; hence, the $e_{t1} \neq e_{d1}$
e_{t2}	e_{d2}	1	1	0.9	$Sim(v_b, w_y) \geq 0.9$; hence, the $e_{t2} = e_{d2}$
e_{t3}	e_{d5}	0	2	0	$Sim(v_b, w_y) < 0.9$; hence, the et3 /= et4. Note that w_y, in this case, is a dummy node.
e_{t4}	e_{d3}	1	1	1.0	$Sim(v_b, w_y) \geq 0.9$; hence, the $e_{t3} = e_{d3}$

The measure is qualified as conservative and hard equality because the value of the threshold makes the equality discrete. Setting the right threshold level is a key for this measure. On the other hand, it is useful in an interactive filtering usage scenario.

3.2 Weak Edge Equality Metric

In the hard edge equality, we quantify the similarity between the discovered match (E_d) and the true match (E_t) by using the threshold value associated with the similarity between v_b and w_y. When the $sim(v_b, w_y)$ is above a certain value we conclude that two edges are perfectly match. We can relax the need to conclude the perfect match by using the $sim(v_b, w_y)$ itself to indicate the degree of match between E_d and the true match E_t. That is $\mid E_t \cap E_d \mid$ is defined as $\Sigma\, Sim(v_b, w_y)$, for all edges $(u_a, v_b) \in E_d$ and $(u_x, w_y) \in E_t$ where $u_a = u_x$. With that $\mid E_t \cap E_d \mid$ definition, the definition of the $\mid E_t \cup E_d \mid$ is derived as follows.

$| E_t \cup E_d | = | E_t | + | E_d | - | E_t \cap E_d |$

$= | E_t | + | E_d | - \Sigma \mathit{Sim}(v_b, w_y)$, for all u_i in E_t

$= \Sigma \mathit{Sim}(u_i, v_j) + \Sigma \mathit{Sim}(u_i, w_k) - \Sigma \mathit{Sim}(v_b, w_y)$, for all u_i in E_t

For completeness, we add the dummy map to Ed similar to that of the hard edge equality as follows. For each edge $e_t = (u_a, v_b)$ where there is no edge $e_d = (u_a, w_y)$, a dummy edge (u_a, ω) is added to the set E_d, and

$\mathit{Sim}(v_b, \omega) = 0.$

Using the above definitions and the example shown in Figure 4 and additional $\mathit{Sim}(u_i, v_j)$ values shown in Figure 5, the weak edge suitability measures using *Jaccard* metric is 0.79 ($| E_t \cup E_d | = 3.4$ and $| E_t \cap E_d | = 2.7$). It should be noted that the edge e_{d4} is not included in the calculation because there is no corresponding e_t edge.

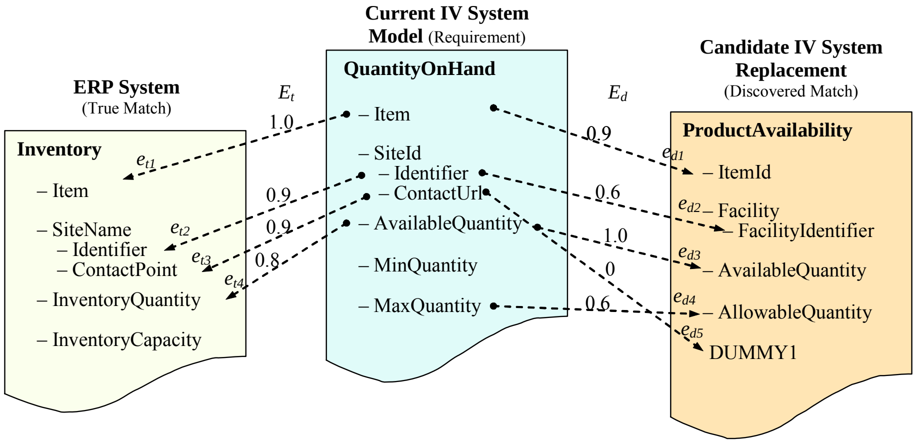

Figure 5: An exemplary software component suitability scenario using weak edge equality

4. Conclusion and Future Work

Using functionality alone can result in an underestimation of the costs and efforts required to replace a software component. This paper has introduced a quantitative measure, called a suitability measure, to assess the compatibility of the new software with the existing software component and with the existing backend system. To make this assessment, we took into account the existing relationships between the current software component and the backend system. This is intuitive because the new software component has to also integrate into this backend system.

Two approaches have been presented: hard edge equality and weak edge equality. In hard edge equality, agreement between data entities in the candidate software component and those in the backend system must be perfect. That is, there is either a complete match or no match.

We can relax this requirement using weak edge equality, which allows partial matches. The degree of match is based on the similarities between the information models. In the hard edge equality the threshold level has to be selected. It is useful in the filtering usage scenario. On the other hand, there is no need to select a threshold in the weak edge equality. It is easier to use in a quick search usage scenario.

At present, the suitability measure only takes into account those edges (maps) that have links between the information models of the backend system (true match) and the candidate software replacement (discovered match). As noted above, there are other possible measures. Combining them into an overall suitability measure will be one of our future research task. We will also perform an experiment to analyze the reliability of the suitability measure to evaluate a new software component as oppose to using merely the similarity between the existing and the new software components. We also envision the ability to compute costs/efforts required to replace the existing software component by establishing an algebraic relationship between the suitability measure and the efforts/costs to deploy the software component. Another issue yet to be addressed is also when there are 1:n or n:m maps. One potential direction is to find a similarity measure that is applicable to 1:n and n:m maps or each decomposed edge.

Disclaimer

Certain commercial software products are identified in this paper. These products were used only for demonstration purposes. This use does not imply approval or endorsement by NIST, nor does it imply that these products are necessarily the best available for the purpose.

Acknowledgement

The authors wish to thank Professor Yun Peng at the University of Maryland Baltimore County for valuable discussion on this paper.

Reference

[1] Ivezic, N., Kulvatunyou, B., Frechette, S., Jones, A., Cho, H., and Jeong, B., An interoperability testing study: Automotive inventory visibility and interoperability, In *Proceedings of e-Challenges*, Hofburg Palce, Vienna, Austria, October 27-29, 2004

[2] Shvaiko, P., A classification of schema-based matching approaches, In *Proceedings of the Meaning Coordination and Negotiation workshop* in *the International Semantic Web Conference*, Hiroshima, Japan, November 2004

[3] Jeong, B., Kulvatunyou, B., Ivezic, N., Cho, H., and Jones, A., Enhance reuse of standard e-business XML schema documents, In *Proceedings of International Workshop on Contexts and Ontology*: *Theory*, *Practice and Applications* (*C&O'05*) in *the 20th National Conference on Artificial Intelligence* (*AAAI'05*), Pittsburgh, PA, July 9, 2005

[4] Jarmasz, M. and Szpakowicz. S, Roget's thesaurus and semantic similarity, In *Proceedings of Conference on Recent Advances in Natural Language Processing* (*RANLP 2003*), Borovests, Bulgaria, September 2003, pp.212-219

[5] Wu, Z. and Palmer, M., Verb semantics and lexical selection, In *Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics*, Las Cruces, New Mexico, 1994, pp.133-138

[6] Sussna, M., Word sense disambiguation for free-text indexing using a massive semantic network, In *Proceedings of the 2nd International Conference on Information and Knowledge Base Management* (*CIKM*), Washington, DC, November 1-5, 1993

[7] Resnik, P., Using information content to evaluate semantic similarity in a taxonomy, In *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (*IJCAI-95*), Montreal, Canada, August 29, 1995, pp.448-453

[8] Zhang, Z., Li, R., Cao, S., and Zhu, Y., Similarity metric for XML documents, In *Proceedings of Workshop on Knowledge and Experience Management* (*FGWM2003*), Karlsruhe, Germany, October 6-8, 2003

[9] Buttler, D., A short survey of document structure similarity algorithms, In *Proceedings of the 5th International Conference on Internet Computing* (*IC 2004*), Las Vegas, NV, June 21-24, 2004

[10] Bhavsa, V.C., Boley, H., and Yang, L., A weighted-tree similarity algorithm for multi-agent systems in e-business environments, In Proceedings of the Business Agents and the Semantic Web (BASeWEB) Workshop, Halifax, Nova Scotia, Canada, June 14, 2003

[11] Flesca, S., Manco, G., Masciari, E., Pontieri, L., and Pugliese, A., Detecting structural similarities between XML documents, In *Proceedings of the 5th International Workshop on the Web and Databases*, Madison, WI, June 6-7, 2002

[12] Ivezic, N., Anicic, N., Jones, A., and Marjonovic, Z., Towards semantic-based supply chain integration, In *Proceedings of the IFIP 5.7 Advances in Production Management System Conference*, Rockville, MD, September 18-21, 2005

[13] Giunchiglia, F., Shvaiko, P., and Yatskevich, M., S-Match: An algorithm and an implementation of semantic matching, In *Proceedings of the 1st European Semantic Web Symposium* (*ESWS*), Heraklion, Greece, May 10-12, 2004, pp.61-75

[14] Li, W.S. and Clifton, C., SEMINT: A tool for identifying attributes correspondences in heterogeneous databases using neural networks, Data & Knowledge Engineering, Vol. 33, 2002, pp.49-84

[15] WordNet – A Lexical Database for English Language. http://wordnet.princeton.edu/

[16] Do, H.H. And Rahm, E., COMA — A System for flexible combination of schema matching approaches, In Proceedings of the 29th International Conference on Very Large BaseBase (VLDB), Berlin, Germany, September 9-12, 2003, pp.610-621

[17] Do, H.H, Melnik, S., and Rahm, E., Comparison of schema matching evaluation, In *Proceedings of GI-Workshop "Web and Databases*", Erfurt, Germany, 2002