# An approach for Large Scale Schema Matching by using Ontology

Su Su Hlaing
*University of Computer Studies, Yangon*
*susuhlaing22@gmail.com*

Nan Saing Moon Kham
*University of Computer Studies, Yangon*
*moonkham.ucsy@gmail.com*

## Abstract

*Schema Matching can be efficiently done by using ontology concepts. Ontology matching is one of the well known topics of Semantic Web research. With the development and the use of a huge variety of data (e.g. DB schemas, ontologies, taxonomies), in many domains (e.g. libraries, travelling, sports, medical fields, etc), Matching Techniques are attempted to overcome the challenge of reconciling these different interrelated representations. In this paper, we are interested in clustering web data sources for large scale schema matching approaches. We attempt to cover the problems of some hidden regularities and semantic conflicts over different representation of web databases. So, we use a same model representation of ontologies to tackle for all relational schemas for large scale semantic matching in our system.*

## 1. Introduction

Matching Schemas is an important implementation in many application domains, such as Semantic Web, data warehouses, e- commerce, ontology integration, query mediation, etc. Specific criteria have been proposed for evaluating and distinguishing between matching approaches. Databases in these domains are filling up with huge amounts of data information with different representations. These data are heterogeneous, frequently changing, distributed, and their number is increasing rapidly.

The presence of vast heterogeneous collections of data causes one of the greatest challenges in the data integration field [8]. Hence, Matching techniques attempt to develop automatic procedures that search the correspondences between these data in order to obtain useful information. In fact, Matching is an operation that takes data as input and returns the semantic similarity values of their elements/ attributes.

It takes as input two ontologies, each consisting of a set of discrete entities and determines as output the relationships (e.g.,equivalence, subsumption) holding between these entities. Many diverse solutions to the matching problem have been proposed so far. Although, there is a difference between schema and ontology matching problems, techniques developed for each of them has been of a mutual benefit [10].

Several tools have been developed towards solving the ontology-matching problem, either in a semiautomatic or fully automatic way. Human-involvement during the process is usually in a trade with the precision and recall percentages of the resulted mappings. Still, both automated and semi-automated tools are suffering in their performance. For instance, most of them cannot handle large real-domain ontologies, although more and more realistic test-beds are used to evaluate ontology matching tools (scalability problem).

Beyond ontology matching methods, tools, and evaluation initiatives/frameworks, recent efforts have been made on ontology-matching-tool-design frameworks [12]. Most recently, ontology matching methods were used to solve the Semantic Web services matchmaking problem [9], transforming the problem of discovering web services into a matching problem between an ontology description representing a service request to an ontology description representing an offered service. Although there are a few tools implementing such an approach still there is more to be done towards improving the process.

In this paper, we represent an approach to organize similar attributes and semantic conflicts for large scale semantic matching. So we apply a general representation for all ontologies while performing semantic mapping approach for relational databases. This paper will emphasis only semantic mapping in web integration system. The rest of the paper is presented as follows. In section 2, it describes the related work of the system. Section 3 describes the Semantic Matching of the proposed system and section 4 presents the ongoing test results of the system. Finally, section 6 is the conclusion and future work of the system.

## 2. Related Work

Heterogeneity in databases also leads to problems like schema matching and integration. The problem of schema matching is becoming an even more important issue in view of the new technologies for the Semantic Web [6].

Schema Matching is the task of identifying semantic correspondences between elements of metadata structures, such as, database schemas, ontologies. However, in today's systems, schema matching is still manual; a time consuming, tedious, and error-prone process, which becomes increasingly impractical considering the high complexity and number of schemas and data sources to be dealt with.

Matching has been approached mainly by finding pair-wise attribute correspondences, to construct an integrated schema for two sources. Several pair-wise matching approaches over schemas and ontologies have been developed.

Holistic schema matching [5], [2], approach attempted to match many schemas at the same time and focused to greedily discover both simple 1:1 and complex matching with a dual mining of positive and negative correlations. The MetaQuerier [3] system's techniques leverage the large scale "regularity" of Web query interfaces to explore their hidden "semantics".

The monotonicity principle and see how it leads to the use of top-*K* mappings [11] presented m:n matching rather than a single mapping. With clustering the elements in schema matching an optimal set of clusters [4] is obtained and each cluster contains elements representing semantically similar (not the same) information. An efficient mapping result in good performance and relevant elements are discovered via matched clusters [7] where irrelevant elements are filtered out via non matched clusters.

B. He, T. Tao, and K. C-C. Chang [4] proposed a schema-based, model-differentiation approach by clustering sources by their *query schemas* with the hierarchical agglomerative clustering algorithm. This approach hypothesized that "homogeneous sources" are characterized by the same hidden generative models for their schemas.

A. Nathalie [1] proposed a schema matching approach based on attributes values and background ontology which implies local ontology matching. D. Aumueller, H-Hai Do, S. Massmann and E. Rahm [6] demonstrated the schema and ontology matching tool COMA++ which includes utilization of shared taxonomies, reusing previously determined match results and a so-called fragment based match approach.

A system architecture for web data integration [11] focusing on resolving the problems of semantic schema heterogeneity between web data sources and proposed an ontology-based approach as a solution for the reconciliation of semantic conflicts between web data at the schema level.

In this paper, we are interested in our work in Matching techniques that aim at identifying semantic correspondences between schemas, ontologies, query interfaces, etc. The aim of our system is to achieve a solution for resolving semantic heterogeneities between schemas matching and organizing these web data sources as a first step in the information integration process. Thus we propose an approach with a same model representation for all ontologies and to organize these similar deep web sources by using domain ontology to map and handle for the reconciliation of semantic conflicts between relational databases.

## 3. A model for Large Scale Semantic Matching

In order to compare and find similar terms between the specific relational schemas, our system in (figure 1) develops a specific domain ontology which includes a dictionary to compare similarity of attributes. Our proposed system develops domain ontology to get an optimal set of attribute pairs (tokenized words).

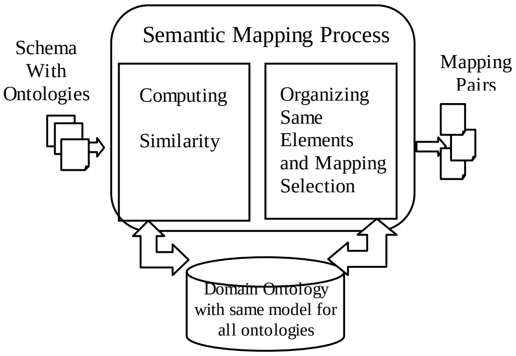
Semantic Mapping Process
Schema With Ontologies
Computing Similarity
Organizing Same Elements and Mapping Selection
Mapping Pairs
Domain Ontology with same model for all ontologies

**Figure 1: A general Procedure for large scale Matching System**

Organizing similar elements in mapping shows relevant elements are discovered via matched clusters. Irrelevant elements are filtered out via non matched clusters. Our system uses general ontology definitions to cope with any different ontologies.

*Definition1: T:=(Conc,Att,Rls,Val),* each ontology element (term) is one following entities:

- Conc: concept or instance of one concept
- Att: attribute of one concept
- Rls: relationship between concepts
- Val: value range of one relationship

*Definition2: Conc:=(name, synoset, Att, key-Att, key-Rls),* each concept is defined with its name, set of its synonyms, attributes, its key attributes, and key with other concepts. The key attributes are subset of concept attributes. The key attributes and key relationships are specific properties and specifications of one concept that characterize the concept. These key properties are specified just for concept definitions of the domain ontology during the development of the domain ontology. We will use these properties as a mapping criterion for finding similar terms in our mapping algorithm.

*Definition3: Att:=(name, synoset),* attribute is defined with a name and a set of synonyms.

*Definition 4: Rls:=(name, synoset, domain, range),* each relationship is defined with a name, set of synonyms and domain and range.

*Definition 5: Val:=(value),* this feature is used for representation range of one relationship that is a value. One value Begins with one of these characters: "=", "<", ">" or "< >" and one string that show the value of its range.

*Definition 6: O:=(G, G'),* each ontology is represented by two graphs.

*Definition 7: G:=(N, E), N=<Conc>, E=<is-a>, G* is acyclic directed rooted graph that consists of nodes, edges. Each node is a concept (or instance of a concept) and each consists of metadata which is created from proposed architecture. Each edge is *"is-a"* relation that shows sub-concept (subclass) relation between nodes. Indeed, *G* is a hierarchy concept model of ontology.

*Definition 8: G' :=( N, E'), N=<Conc, Val>, E'=<Rls>, G'* is cyclic graph that consists nodes and edges. Each node is a concept (or instance of a concept) or one value and each consists of metadata-label which is created from proposed architecture. Each edge is relationship between two nodes that show the relationship between concepts. Indeed, G' is a concept relationship model of ontology.

These definitions represent as a general model representation for all relational databases with their own ontologies. Our system uses ontology concepts for organizing and mapping to achieve the best results for semantic matching.

We use a similarity calculating function of tokenized words and calculate similarity of each terms or elements by using local dictionary. The process of computing similarity is to find the distance between semantically similar terms by calculating tokenized words by using ($T_S$,$T_T$ ) function.

Given the two schemas S and T, the degree of similarity between concept Ts and Tt is computed as equation (1):

$$sLSim(T_S,T_T)=\frac{\sum_{t_s\in T_S}[max_{t_t}\in T_T\ lingSim(t_s,t_t)]+\sum_{t_t\in T_T}[max_{t_s}\in T_S\ lingSim(t_t,t_s)]}{|T_S|+|T_T|} \quad (1)$$

($T_S$,$T_T$ ) =1- sLSim ($T_S$ ,$T_T$) where

($T_S$ = a source element, $T_T$ = a target element, $t_s$ = a source token, $t_t$ = a target token, |$T_S$ | = the number of source tokens, |$T_T$ | = the number of target tokens, lingSim = linguistic similarity measure of two given tokens)

In this section we assume that every relational databases schemas which has their own ontologies. Our system measure that each similar pairs which include relative similar attribute of all schemas with MF. And then our system presents a mapping algorithm to outperform quality of mapping results. The algorithm performs mapping process according to the specific domain ontology.

***MF (Mapping Function)***: *MF= [0, 1].* The semantic similarity of terms used in our system is usually a number between 0 and 1. 1 signifies extremely high similarity/relatedness, and 0 signifies little-to-none.

***First step:*** MF is executed between *Conc1(name)* (root of query path), all its synonyms names *Conc1(syn-name)* with all local ontology concepts (all *ConcL(name)* ).

```
for all ConcL(name)<>null
do
{if MF(Conc1(name), ConciL(name)) >= threshold then
add (Conc1, ConciL) to similarity-table;
else: for all Conc1(synoname) <>null do
{if MF(Conc1(synoname), ConciL (name)) >= threshold then
add (Conc1,ConciL) to similarity-table; }}
while ConciL (name)<>null do
{ while ConciL (A) < > null or ConciL (R)< >null do
if each MF ( Conc1( key-property-name & keyproperty-synonames), ConciL (Att & Rls)) >= threshold then
ConciL (similar-property) +1 ;}
Conc1L    Conc [MAX ConciL (similar-property)];
Add (Conc1, Conc1L) to C-mapping-table;
```

***Second step:*** After finding similar concept of *Conc1*, if *Conc1* has attribute in query path then we must find its similar attributes in local ontology. We should notice, we just execute MF between *Conc1-atrribute-name*, all *Conc1-att-synosetnames* with *attributes-names* and *relationships-names* of its mapping pair (*Conc1L* in mapping table). We choose maximum MF that is above threshold and store similar attribute pairs in *Conc-att-mapping table* (such as*: <Conc1-Att1, CconlL-Att1L>, <Conc1-Att2, Conc1L-Att2L>…..*).

```
while Conc1 (Att-name)<>null do
{if MF (Conc1 (Att-name), Conc1L (Att-name or Rls-name)) >=threshold then
add (Conc1 (Att-name), Conc1L (Att-name or Rls-name))     to attmapping-table;
else: while Att-synoname <>null do
{if MF (Conc1 (Att-synoname), Conc1L (Att-name or Rls-name)) >= threshold then
add (Conc1 (Att-name), Conc1L (Att-name or Rls-name))     to att-mapping-table ;}}
```

***Third step:*** we must find similar concept for next term of query path (*Conc2).* There are two situations here: *Conc2* has "*isa*" relationship with *Conc1* (*Conc2* is sub-concept of *Conc1*) or *Conc2* has "*R*" relationship with *Conc1* (*Conc1* and *Conc2* are domain and range of same *Rls*).

The algorithm executes MF between *Conc2* and all of local ontology concepts. If it finds similar concept of *Conc2* then enters similar pair <*Conc2,Conc2L*> in *Cmapping-table* and executes MF for query path attributes of *Conc2* (***second step*** of algorithm) else enters <*Conc2 , null*> in *C-mapping-table*. Algorithm repeats ***third step*** for next others nodes until last element of query path. If algorithm doesn't find similar concept of main query concept (concept in question) from local ontology so, mapping doesn't execute between user query terms and local ontology terms and this local ontology is failed.

## 4. Experimental Results

Our system use precision and recall to measure for the exact mapping. Let *A* be the set of individual concept pair mappings in the exact mapping and let *B* be the set of individual concept pairs mappings in the best mapping. Precision (*P*) and recall (*R*) are measured as:

$$P = \frac{|A \cap B|}{|A|} \qquad R = \frac{|A \cap B|}{|B|}$$

Precision and recall both reach the maximum value of *1* whenever *A=B*. Low precision is an indication of many false negatives and low recall is an indication of many false positives. We observe that experimental results in figure (2) show that our proposed system significantly achieves the exact mapping results with higher mapping thresholds.

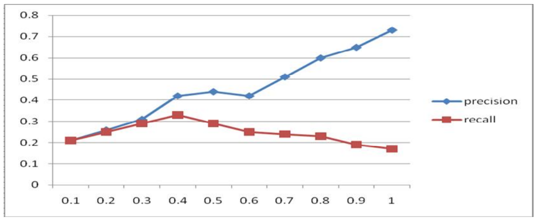

**Figure 2. Precision and Recall for mapping points using ontology.**

## 5. Conclusion

In this paper, we first recommended system architecture for schema matching with specific domain ontology. Our system intend to handle not only the specific problem of semantic heterogeneity between schemas matching but also includes a same model representation for all relational databases schemas with their own ontologies to map between the similar concepts of attributes pairs. Our proposed system achieves significant results in mapping with higher mapping thresholds. For future studies, we aim to develop our system to be better performance in schema matching.

## 6. References

[1] A. Nathalie, Institut Géographique National, Laboratoire COGIT 2 Avenue Pasteur, 94160 Saint-Mandé, France, Université de Paris Est Cité Descartes, Champs-Sur-Marne, 77454 Marne-La-Vallée cedex 2, France , "Schema Matching Based on Attribute Values and Background Ontology", Proc: 12th AGILE International Conference on Geographic Information Science 2009, pp: 1 of 9.

[2] B. He, K. C-C. Chang, and J. Han."Discovering Complex Schema Matching across Web Query Interfaces: A Correlation Mining Approach". In *Proceedings of the 2004, ACM SIGKDD Conference (SIFKDD 2004)*, 2004.

[3] B. He, K. C-C. Chang "Towards Building MetaQuerier: Extracting and Matching Web Query Interfaces" , Computer Science Department , University of Illinois at Urbana Champaign, **Proc** : 21 $^{st}$ International Conference on ICDE, 05-08 April 2005, pp: 1098-1099ISSN: 1084-4627 , ISBN: 0-7695-2285-8.

[4] B. He, T. Tao, K. C. Chang, "Organizing Structured Web Sources by Query Schemas: A Clustering Approach", Computer Science Department University of Illinois at Urbana Champaign, 2004 Publication type: Conference paper, B. He, Z. Zhang *4, December 2006 2-5.*

[5] B. He, K. C-C. Chang, "A Holistic Paradigm for Large Scale Schema Matching" ,Computer Science Department University of Illinois at Urbana Champaign, Publisher: University of Illinois at Urbana-Champaign Champaign, IL, USA , pp: 193 ,Year of Publication: 2006 , ISBN:978-0-542-98862-2.

[6] D. Aumueller, H. H. Do, S. Massmann, E. Rahm, . "Schema and Ontology Matching with COMA++ ",908Department of Computer Science, University of Leipzig Augustusplatz 10/11, Leipzig 04103, Germany , Proc: 2005 ACM SIGMOD international conference on Management of data (2005), pp. 906- 908.

[7] Hajmoosaei, A., A-Kareem, S., (2007) *Ontology-Based Approach for Resolving Semantic Schema Conflicts in the Integration of Web Data Sources.* In: Research Excellence and Knowledge Enrichment in ICT: Proceeding of the 2nd International Conference on Informatics, 27th - 28th November 2007, Petaling Jaya, Selangor, Malaysia.

[8] J. Geller, New Jersey Institute of Technology, S. A. Chun, College of Staten Island, City University of New York, Y. J. An , Fairleigh Dickinson University, "Toward the Semantic Deep Web".

[9] Klusch, M.; Fries, B.; Sycara, K. (2006). Automated Semantic Web Service Discovery with OWLS-MX. Proceedings of 5th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Hakodate, Japan, ACM Press. Best Paper Award Nominee.

[10] Shvaiko, P., Euzenat, J. (2005). A Survey of Schema based Matching Approaches Journal on Data Semantics, IV, 2005.

[11] Technion , Israel Institute of Technology , "Why is Schema Matching Tough and What Can We Do About It? *Volume 35, Number 4, December 2006 2-5.*

[12] Valarakos A., Spiliopoulos V., Kotis K., Vouros G. (2007). AUTOMS-F: A Java Framework for Synthesizing Ontology Mapping Methods. I-KNOW 2007 Special Track on Knowledge Organization and Semantic Technologies 2007 (KOST '07) September 5, 2007, Graz.