A semantic similarity analysis for data mappings between heterogeneous XML schemas

Jaewook Kim
Department of Computer Science and Electrical Engineering University of Maryland, Baltimore County, USA
Yun Peng
Department of Computer Science and Electrical Engineering University of Maryland, Baltimore County, USA

ABSTRACT

One of the most critical steps to integrating heterogeneous e-Business applications using different XML schemas is schema mapping, which is known to be costly and error-prone. Past research on schema mapping has not made full use of semantic information imbedded in the hierarchical structure of the XML schema. In this chapter, we investigate the existing schema mapping approaches and propose an innovative semantic similarity analysis approach to facilitate XML schema mapping, merging and reuse. Several key innovations are introduced to better utilize available semantic information. These innovations includes: 1) a layered structure analysis of XML schemas, 2) layer-specific semantic similarity measures, and 3) an efficient semantic similarity analysis using parallel and distributed computing technologies. Experimental results using two different schemas from a real world application demonstrate that the proposed approach is valuable for addressing difficulties in XML schema mapping.

INTRODUCTION

The electronic business (e-Business) requires interoperability between different e-Business systems for the seamless exchange of information either within or across enterprises. One of the most critical steps to achieving a successful integration and interoperability between heterogeneous e-Business systems is schema mapping, which is known to be costly and error-prone. Schema mapping is, roughly speaking, to identify how information can be shared between heterogeneous schemas and how they can be mapped, merged, or reused for integration and interoperability of their e-Business systems.

A schema typically defines the syntax of business documents (or instances), but it also contain semantic information, i.e., the meaning of elements in business documents. Sometimes, the schema refers to other semantic sources such as ontology, dictionary, or documentation for additional semantic information. The typical way for schema mapping is to identify semantically identical or similar elements between the two schemas. Many approaches have been proposed, but the challenge is still daunting because of the complexity of schemas and immaturity of technologies in semantic representation, measuring, and reasoning.

The goal of this chapter is to investigate the existing schema mapping approaches and to propose an innovative semantic similarity analysis approach to facilitate XML schema mapping, merging and reuse. Several key innovations are introduced to better utilize available semantic information. These innovations includes: 1) a layered structure analysis of XML schemas, 2) layer-specific semantic similarity measures, and 3) an efficient semantic similarity analysis using parallel and distributed computing technologies. Experimental results using two different schemas from a real world application demonstrate that the proposed approach is valuable for addressing difficulties in XML schema mapping.

BACKGROUND

The Challenges for Data Mappings between Heterogeneous XML Schemas

Over the past decades, the eXtensible Markup Language (XML) has emerged as one of the primary languages to help information systems in sharing structured data. Especially, XML schemas have been widely used in the e-Business for enterprises to exchange the business documents with their partners in a supply chain. The popularity of the XML and XML schema leads to an exponential growth of Business-to-Business (B2B) transactions. This success, however, leads to several problems: 1) individual enterprises often create their own XML schemas with information most relevant to their own needs; 2) different enterprise groups define different but similar XML schemas; and 3) the enterprises often extend or redefine the existing standard XML schema for their own needs. To successfully integrate heterogeneous e-Business systems, therefore, it is now critical to integrate their respective different XML schemas. This is what is called schema mapping.

The schema mapping is the process of identifying if and how two schemas are semantically related (Miller *et al*, 1994; Rahm & Bernstein, 2001; Shvaiko & Euzenat, 2005). It is one of the most important steps to integrate heterogeneous e-Business systems; however, it is typically largely performed manually by human engineers who are at best supported by some graphical interface tools. This manual mapping process is known to be very labor-intensive, costly, and error-prone (Gal, 2006; Rahm & Bernstein, 2001). As the e-Business systems grow to handle more complex databases and applications, their schemas become larger and more complicated. This further increases the search space to be examined as well as the number of correspondences to be identified. As a result, it is critical to automate the schema mapping task as much as possible to reduce the costs of labor-intensive data integration work and to reduce the mapping errors.

The XML schema mapping can be classified into two types depending on the types of the e-Business standard schemas: component schema and document schema. The component schema only contains reusable and extensible components (types or elements) as global type definition (e.g., OAG Common Core Component schema), while the document schema contains a global root element to define one valid XML document (e.g., Purchase Order Schema). The document schema may reuse or extend the components defined by the component schema. For schema integration, the component schema mapping mainly identifies the relations between global components (types or elements), while the document schema mapping mainly identifies relations between leaf nodes (elements or attributes). In this research, we focus on the component schema mapping.

Schema Mapping Techniques

Many schema mapping methods have been proposed (summarized in surveys by Rahm & Bernstein, 2001; Shvaiko & Euzenat, 2005). Typically, these methods first attempt to identify semantic relationships between the elements of two schemas. According to Shvaiko & Euzenat (2005), the schema mapping techniques can be distinguished as two main alternatives by the granularity of the mapping: element-level and structure-level. The element-level approaches determine the mapping elements in the target schema for each element of the source schema; whereas structure-level approaches refer to mapping combinations of elements that appear together in a structure. In the ideal case of structure-level approach, all components of the structures in the two schemas fully match. These techniques can be further classified by different types of elementary mapping techniques.

Figure 1 shows a simplified version of the classification of schema mapping techniques suggested by (Shvaiko & Euzenat, 2005) based on frequently used techniques. Note that all ontology related techniques are omitted, corpus-based metric includes all the techniques using corpus (including linguistic resources), and graph-based metric includes all the techniques related to the graph analysis techniques (including taxonomy-based metric).

Figure 1. Classification of schema mapping approaches

The simplest mapping technique is a string-based metric which computes similarity between terms or their descriptions using its lexical information. There are a variety of string-based metrics, such as the widely used cosine similarity and Jaccard coefficient measures (Rijsbergen, 1979; Sneath, 1957). The string-based metric can be enhanced using a language-based metric which is a kind of preprocessor for the input string such as tokenization, lemmatization, and elimination (Madhavan et al, 2001; Shvaiko & Euzenat, 2005).

Corpus-based metric can also improve the string-based metric by obtaining more accurate and less ambiguous semantics (e.g., synonyms or hyponyms) for words in the element labels. Not only can the common knowledge corpora such as WordNet (Miller, 1995) but also domain specific corpora can be used to enrich the meaning of the words. One of the important resources in a corpus is the lexical taxonomy among words (e.g., parents, children, ancestor, and descendant relationships). Some researches have been proposed based on a lexical taxonomy of the corpus (Qin et al., 2009; Yang & Powers, 2005).

Another important resource obtained from corpus is the contents linked to topically related words. Topically related words form the Topic Signatures (Lin & Hovy, 2000) which provide word vectors related to a particular topic. Topic Signatures are built by retrieving a group of words that related to a target word from corpus. The topic signature can be defined as a family of related terms $\{t, <(w_1,s_1)\ldots(w_i,s_i)\ldots>\}$, where t is the topic (i.e. the target concept) and each w_i is a word associated with the topic, with strength s_i.

Corpus also provides the statistical information related to the importance of words. The different importance individual entities and relationships have plays the different roles in semantic similarity measurement. The information content (IC)-based metric was proposed to utilize this statistical information (Resnik, 1995; Lin, 1998). This approach measures the similarity between two entities (e.g., two words, two objects, or two structures) A and B based on how much information is needed to describe $common(A, B)$, the commonality between them (e.g., the features or hypernyms that two words share). According to information theory (Cover & Thomas, 1991), entities appear widely in many objects have less information than those appear rarely. In other words, more specific entities carry more information than generic and common entities as an indication of similarity between two specific objects. Therefore, the more specific the *common*(*A*, *B*) is, the more similar *A* and *B* will be. By the information theory, the information content of a concept or word *C* is defined as $I(C) = -\log P(C)$. Then *common*(*A*, *B*) can be measured by the information content of the most specific common hypernyms of *A* and *B*, and the similarity between *A* and *B* is given as

$$Sim_{IC}(A, B) = \max_{C \in S(A,B)} I(C) = \max_{C \in S(A,B)} (-\log P(C)) \qquad (1)$$

(where *S*(*A*, *B*) is the set of all concepts that subsume both *A* and *B*, *I*(*C*) is the information content of *C*, and *P*(*C*) can be calculated as word frequencies in a corpus).

A variety of graph-based techniques have been proposed for structure-level mapping. Typically, the graph-based metric quantifies the commonality between components by taking into account the lexical similarities of multiple structurally-related sub-components of these terms (e.g., child, parents, and leaf components). Because most schemas can be viewed as hierarchical graphs containing terms and their parent-children relationships, many mapping algorithms have been developed based on either top-down or bottom-up traversal techniques to analyze all elements (Rahm & Bernstein, 2001). Among the existing approaches, TransScm (Milo & Zohar, 1998) and Tess (Lerner, 2000) are based on the top-down approach, while Cupid (Madhavan *et al.*, 2001) and Similarity Flooding (Melnik *et al.*, 2002) take the bottom-up approach. Another technique of graph-based metrics is a taxonomy-based technique that can be applied to 'IS-A' taxonomy such as ontology. For example, the edge counting is a well-known traditional approach based on conceptual distance in taxonomy (Rada *et al.*, 1989).

The graph-based metric typically provides a more comprehensive measure than do the string-based and corpus-based similarity metrics because it looks beyond the individual labels and considers terms' relationships to others. However, it often fails to recognize the semantics in the language and corpus.

Each of the existing similarity measures has its strengths and weaknesses. More importantly, each typically makes use of only part of the available semantic information. Therefore, a mapping that uses just one approach is unlikely to find as many good mapping candidates as does the hybrid approaches that combine several mapping approaches (Rahm & Bernstein, 2001). A few hybrid mapping approaches have been proposed. For example, Jeong (2008) used a machine learning to combine several mapping approaches. Giunchiglia (2004) proposed a common hybrid system called S-Match which allows a single component be plugged in, unplugged or customized. These hybrid approaches choose different mapping criteria that can capture different semantic information in the schema and then combine them in a particular way to increase the accuracy of the mapping result. However, they fail to examine and make full use of the semantic information imbedded in the hierarchical structure of the XML schema. The hierarchical structure can be divided into several layers according to the level of hierarchy. Typically, different layers may carry different semantic information of the data elements, which may require different, layer-specific approaches to gauge the similarities. Therefore, it is beneficial to establish layered specific metrics for semantic similarity analysis of the XML schemas (Kim *et al.*, 2007).

Quality of Mapping Measures

To evaluate the quality of the mapping measures, several performance evaluation scoring functions have been proposed (Do *et al.*, 2003). The typical method to evaluate the mapping measures is to compares the derived mappings to the real mappings. The human mapping integrators first have to manually generate a set of real mappings which can be used as "gold standard" to be compared to the automatically derived mappings. The comparison of the real mapping to derived mapping is shown in Figure 2.

Figure 2. Real mappings vs. Derived mappings

The set of derived mappings can be categorized as true positives (i.e., B); false positives (i.e., C); false negatives (i.e., A); and true negatives (i.e., D). Note that among all derived mappings, only the true positives are considered correct mappings. Based on these categories, two basic measures of mapping quality, Precision and Recall, can be computed.

Precision expresses the proportion of correct mappings among all the derived mappings, which can be defined as:

$$\text{Precision} = \frac{\text{number of correct mappings derived}}{\text{total number of mappings derived}} = \frac{|B|}{|B|+|C|} \qquad (2)$$

Recall expresses the proportion of the found correct mappings among all the correct mappings, which can be defined as:

$$\text{Recall} = \frac{\text{number of correct mappings derived}}{\text{total number of correct mappings}} = \frac{|B|}{|A|+|B|} \qquad (3)$$

Precision = 1 indicates that all the mappings derived by the mapping measures are correct, while Recall = 1 means that all correct mappings are found by the mapping measures. A trade-off between recall and precision is provided by the F-measure:

$$\text{F-Measure} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (4)$$

Although formal F-Measure supports different relative importance to be attached to Precision and Recall, Eq. (4) is a special case when Precision and Recall are considered equally important.

Parallel and Distributed Computing Technologies for Large-scale Schema Mapping

As more complicated and large-scale schemas and more sophisticated mapping algorithms have been introduced, large-scale schema mapping has become a challenging problem in terms of the computational cost. Several research groups have actively studied the issues concerning large-scale schema mapping. (He & Chang, 2004) proposed a "holistic schema matching" approach that can match many schemas at the same time and find all matches at once. Another similar approach (Saleem & Bellahsene, 2008) creates a mediated schema tree from a large set of input XML schema trees and defines mappings from the contributing schema to the mediated schema.

Alternatively, one can address the performance of computationally intensive similarity analysis in large-scale schema mapping by parallel and distributed computing technologies. The parallel computing is a computation technology in which multiple concurrent processes work simultaneously and cooperate with each other for a single task, while the distributed computing deals with the development of applications that execute on different computers interconnected by networks. Thus, the parallel and distributed computing can refer to a computation technology in which many calculations are carried out simultaneously, operating on multiple computers interconnected by networks for a single task. The parallel and distributed computing in local networks is also called cluster computing and called grid computing in wide-area networks.

Many parallel and distributed computing technologies have been introduced (Asanovic *et al.*, 2006). There is a well-known cluster computing technology called Hadoop (Borthaku, 2007; Dean *et al.*, 2008) which is a Java software framework that supports data intensive distributed applications. Hadoop uses a new programming model called Map/Reduce for processing and generating large data sets. This platform allows programmers without any experience with parallel and distributed systems to utilize easily the resources of a large distributed system.

For grid computing technology, Globus Alliance provides an open source toolkit called Globus Toolkit (Foster & Kesselman, 1997). The Globus Toolkit makes extensive use of Web Services to define interfaces and structures of its components, which provide flexible, extensible, and widely-adopted XML-based mechanisms for describing, discovering, and invoking network services. The grid computing can also be implemented using the Message Passing Interface (MPI) (Asanovic *et al.*, 2006) standard which is an Application Programming Interface (API) specification that allows many computers to communicate with one another. MPI-like Message Passing for Java (MPJ) (Carpenter, 2000) provides a Java software toolkit for the MPI standard. These tools have different technical backgrounds and performance trade-offs. Table 1 shows different features of Hadoop, Globus Toolkit and MPJ.

Table 1. Comparison of Hadoop, Globus toolkit and MPJ

	Hadoop	Globus toolkit	MPJ
SW requirement	Java, SSHD	Java, Ant	Java
System Setup	System-specific	System-independent	System-independent
Security	SSH	WS-Security	No support
Data manage	DFS	GridFTP	No support
Type	Clustering	Grid	Grid

The Hadoop and Globus Toolkit have advantages for easy handling of a large distributed system but require extra software installation and management for clusters and Web Services, respectively. On the other hand, MPJ requires a simple environment configuration and programming architecture, but does not provide any security or data management functionalities.

LAYERED SEMANTIC SIMILARITY ANALYSIS OF XML SCHEMAS

As mentioned earlier, many schema mapping methods have been proposed, but most of them fail to sufficiently investigate and utilize semantic information imbedded in the hierarchical structure of the XML schemas. The data elements in the XML schema can be divided into several layers according to the level of the hierarchy. To utilize these layers for schema mapping, we propose a layered semantic similarity analysis which divides data elements in the XML schema into different layers and then measures semantic similarities at each layer using layer specific metrics.

Layered Semantic Structure of XML Schemas

An XML schema defines a set of global components, each of which can be represented as a tree with a set of linked nodes. Each node in a tree has zero or more child nodes. There are three types of nodes: 1) the root, 2) the leaves, and 3) the intermediate nodes (those with both a parent and some children). Leaf nodes are called “atoms” since they are the smallest units and cannot be further divided. Each tree thus can be divided into three layers: 1) *the top layer* (containing the root of the tree), 2) *the atom layer* (containing leaf nodes), and 3) *the inner layer* (containing intermediate nodes). Note that some trees may have empty inner layers, whereas others may have only one node that is considered to be in both the top and atom layers.

According to the XML schema best practices (Stephenson, 2004), XML schema designers have moved away from complex multi-level type hierarchy to a simple 2~3 level hierarchy where most of the structural assemblies are done by composition. The design by composition leads the top layer labels to contain specific and representative terms (e.g., camera, vehicle) of a global component while the bottom layer labels to contain more common and shared terms (e.g., unit, size, and code). This implies that each layer typically captures the semantic information of a global component from different perspectives. Through its label and namespace, a top layer (the root node) specifies the data object that the global component is intended to describe. The bottom or atom layer (leaf nodes) includes the atomic elements that the designer felt necessary to describe the global component. The inner layer (nodes in between) provides the structural information of the global component by specifying how the atomic elements are grouped into intermediate nodes and, eventually, into the global component (the root). The linguistic information in the labels of both atomic and intermediate nodes may also help to qualify the semantics of the global component.

Figure 3 shows the labeled graph examples of two different XML schemas describing global data element "*vehicle*" (e.g., *make, model, model year, dealer information, mileage at failure, mileage at repair* and *etc*). The two schemas were defined by two different workgroups at the Automotive Industry Action Group (AIAG): (a) Truck and Heavy Equipment (T&HE) and (b) AIAG Resource schemas (AIAG-R).

Figure 3. Three layers of XML schema

The labels in their top layer nodes indicate that both are intended to represent the same object "*vehicle*". However, the designers' thoughts differ with regard to what atomic elements are needed (*e.g.,* in (a) "*VehicleInformation*" only includes generic atomic elements such as the "*code*" and the "*description*", whereas in (b) "*Vehicle*" includes more specific atomic elements such as "*TaxID*", "*Address*", "*Name*", and so on). They also differ on how these atomic elements should be organized (see their different inner layers). In fact, the VehicleInformation in the T&HE schema has 12 intermediate nodes and 198 atoms, while the numbers for the Vehicle in the AIAG schema are 81 and 972, respectively. Also note that the same set of ingredients (atoms) can produce data elements of different semantic information depending on how they are cooked (*i.e.,* structured) or packaged (*i.e.,* the identity of the top layer node). For example, several party elements (*CustomerParty*, *DealerParty*, and *SellingParty*) defined in AIAG-R schema all contain the same atoms and intermediates but are intended for semantically different data objects.

The complex relationship between nodes at different layers requires layer-specific semantic analysis tools and a mechanism to combine these layer-based similarities. For this reason, we utilize three similarity measures. The first one, called atom-level similarity, measures the similarity between two atom layers of two global components. The second one, called label similarity, measures the similarity between the labels (names). The last one, called structure-level similarity, measures the similarity between two inner layers of two elements. These three measures and the process for their combination are described in the following sections.

Atom-level Similarity Measures

Atom-level similarity between two global components is primarily determined by the atoms they share. However, not every atom is equal in determining semantic similarity. The sharing of an atom that is widely used by many components of the two global components is not as strong an indication of similarity as the sharing of a rarely used atom (Lin, 1998; Resnik, 1995). To account for the degree of importance of individual atoms, an IC-based measure for atom layer similarity is proposed. Specifically, let *A*(*x*) and *A*(*y*) denote the sets of atoms of global components *x* and *y*, respectively. Then, the atom level similarity between x and y is defined as

$$Sim_A(x,y)=\frac{2\cdot\Sigma_{c_i\in A(x)\bigcap A(y)}I(c_i)}{\Sigma_{c_i\in A(x)}I(c_i)+\Sigma_{c_j\in A(y)}I(c_j)} \qquad (5)$$

This can be seen as combining the IC-based measure with the Jaccard coefficient where the numerator measures *common*(*x*, *y*) and the denominator is a normalization factor (Lin, 1998). The probability of each atom is taken as its frequency in any corpus related to the source and target schemas. For instance, we can use a corpus formed by all labels in both the source and target schemas or by all words extracted from some domain specific documents.

Eq. (5) is based on the assumption that the source and target schemas share a significant number of atoms. Two atoms can be treated as either completely similar (with a similarity score of 1) if they have the same label or completely dissimilar (with a similarity score of 0) if they do not. Eq. (5) can be generalized for use in situations where similarity scores between many atom pairs are between 0 and 1 (Peng, 2006). For that, we partition *A*(*x*) into two sets: $A_1(x)$ contains those components of *x* that have similar counterparts in *A*(*y*) (i.e., with non-zero pair-wise similarity), and $A_2(x)$ = *A*(*x*) – $A_1(x)$. For every $c_i \in A(x)$, we define its map to *A*(*y*) as

$$m(c_i)=\max_{c_j\in A(y)} Sim(c_i,c_j)>0 \qquad (6)$$

Then the similarity of *x* to *y* is given as

$$Sim_A(x,y)=\frac{\Sigma_{c_i\in A_1(x)}Sim(c_i,m(c_i))\cdot I(c_i)}{\Sigma_{c_i\in A(x)}I(c_i)+\Sigma_{c_j\in A(y)}I(c_j)} \qquad (7)$$

Label Similarity Measures

The label or name *x* of a node is a word or concatenation of words (or their abbreviations). One approach for label similarity measure is string-based metric, which computes similarity between two labels. As discussed in the previous section, the string-based metric can be enhanced using language-based metric and linguistic resources. Therefore, before similarity is compared, a pre-process called “label normalization” is conducted to obtain full words (denoted as *L*(*x*)) from the concatenations and abbreviations. For example, *L*(*VehicleInformation*) = {*vehicle*, *information*}. To better ascertain the semantics of these words and to deal with the problem of synonyms, each word is expanded using its description, which can be obtained from a variety types of sources such as WordNet, schema annotation, web search, business related documentation, and so on.

The descriptions of all the words in *L*(*x*) are then put together under two constraints to form a vector of words, *W*(*x*). First, for a fair comparison, *W*(*x*) should be independent of the lengths of descriptions, which vary greatly from word to word. To achieve this, the normalization of the *W*(*x*) is one possible approach, which make the length of descriptions for all *W*(*x*) the same. Second, words in *L*(*x*) are not equally important in defining *x*’s semantic information (e.g., “*vehicle*” is certainly more important than “*information*” in the label “*VehicleInformation*”). There are several approaches to address the weighted sets of words such as noun phrase analysis from natural language processing. One easiest way is to duplicate more important words and to truncate less important words in the *L*(*x*). The importance of the words can be obtained from its information contents. Finally, the similarity of labels *x* and *y* can be measured by *W*(*x*) and *W*(*y*) using a variety of string-level similarity measures, such as Jaccard coefficient and cosine similarity. Using cosine similarity, the similarity can be defined as

$$Sim_T(x,y) = \frac{W(x)W(y)}{|W(x)||W(y)|} = \frac{\Sigma_{k\subset W(x)\cap W(y)} f_x(k) f_y(k)}{\sqrt{\Sigma_{i\subset W(x)} f_x(i)^2}\sqrt{\Sigma_{j\subset W(y)} f_y(j)^2}} \tag{8}$$

(where $f_x(i)$ and $f_y(j)$ are the frequencies of the term i and j in $W(x)$ and $W(y)$, respectively).

Another way to better ascertain the semantics of words is to collect for each word in L(x) the Topic Signatures that appear most distinctively in documents related to the target word. In order words, we construct lists of closely related words for each word in L(x). For example, given 'car', we can find the related word set {ford, vehicle, truck, audi, safety, …}. A topic signature is a vector which can be defined as

$$TS = \{t, <(w_1, s_1), (w_2, s_2)...(w_i, s_i)...>\} \tag{9}$$

(where t is the given target word (i.e., topic) and w_i is a related word with its relatedness weight s_i).

We can build the topic signatures from the documents searched at the Internet using the query of the target word. Figure 4 shows the strategy to build such lists.

Figure 4. Procedure to build Topic Signatures

We first build queries which are used to search in the Internet (i.e., Google Search) those documents related to the given target word. The query may include domain restriction (e.g., "car site:ford.com") to search only in the given domain (e.g., "ford.com"). Query engine retrieves a collection of the relevant documents. For each collection, we extract the words and their frequencies, then retrieve a list of most relevant words (i.e., A in Figure 4) based on their frequencies. Because the most relevant words may include the common words such as stop words, we collect a list of the most common words (i.e., B in Figure 4) which are frequently appear in any documents except the relevant documents, and finally remove the words in B from. Then, the resulting list of the most relevant words is the topic signature which can be considered as a vector of words, *W(x)*. Similar to the WordNet descriptions approach, W(x) will be normalized so that all $W(x)$ have the same number of words, and the similarity measure using topic signatures can be obtained by Eq. (8).

Any type of structure-level similarity measure can be used to compute inner-layer similarity, but currently we only extend the label similarity measure for the inner-layer's similarity measures (*i.e.*, $Sim_I = Sim_T$ where x (and y) is the union of labels of all inner nodes).

Combined Similarity Score

A variety of algorithms for combining individual similarity measures (Sim_A, Sim_T, Sim_I) can be used, such as *average (a, b, c)*, *max (a, b, c)*, *additive* $(1 – (1 – a)(1 – b)(1 – c))$, and weighted sum $Sim(x, y) = w_A Sim_A + w_T Sim_T + w_I Sim_I$. The weighted sum was found the most useful because it allows the adjustment of weights to best reflect the importance of measures at individual layers. However, finding the best weights is a challenge. For the experiments results, the weights are obtained from the domain experts or learned from human semantic mapping data.

Efficient Schema Mapping Using Grid Computing

The time complexity of XML schema mapping largely depends on three factors: the number of elements that an XML schema defines, the structural complexity of each element, and the complexity of the mapping algorithm. Recently, more complicated XML schemas and highly complex mapping algorithms have been introduced. For instance, AIAG-R and TH&E schemas used for the experiments have hundreds top-level elements and each top-level element contains thousands of intermediate and atom nodes, all of which need to be looked at during the mapping process. It can take more than an hour to conduct the semantic analysis and complete the mapping.

To address this problem, we use a grid computing technology called MPJ to distribute the heavy workload of the XML schema mapping analysis among a cluster of processors connected. The MPJ, developed to enable high-performance computing (HPC) using Java, is well-suited to handling computations where a task is divided up into subtasks, with most of the processes used to compute the subtasks, and only a few processes (often just one) for managing the tasks. The manager is called the "master" and the others the "slaves".

For mapping between a source schema with *n* data elements and a target schema with *m* data elements, we compute the semantic similarities between all possible pairs of source/target elements, generating an $n \times m$ matrix called the *similarity matrix*. According to the layered semantic similarity analysis, the semantic similarities can be computed by combining three different similarities for each of the three layer of XML schema. Because the similarity measure for each pair of source/target elements and the different semantic similarities for each layer can be computed in parallel, grid computing technology can be applied to enhance performance.

EXPERIMENTS AND RESULTS

Two sets of experiments were conducted; the first set was intended to validate the layered approach for semantic schema mapping, the second set to investigate the effect of the grid computing in speedup the computation. In these experiments, the Truck and Heavy Equipment schema (T&HE) were used as the source schema and the AIAG Resource schema (AIAG-R) as the target schema. The experiments and the results are given below.

Experiments and Results for Layered Semantic Similarity Analysis

A prototype system is implemented to compute Sim_T, Sim_I, and Sim_A as given in Eqs. (5) and (7). The system also supports several options to choose different algorithms for different layers of the schema structure and several combination rules. The 49 manual mappings from T&HE schema to AIAG-R schema produced by human integrators are used as the basis to evaluate the performance of the system. For each of the 49 T&HE global components, the system recommends five most similar AIAG-R elements.

A series of experiments has been conducted using the prototype system with varying parameters. We evaluate performance using a set rather than a single recommendation because the objective is not to fully-automate the process but rather to assist the human expert. A recommendation is considered a match if it contains the manual mapping. Various similarity measures, individual and combined, are used:

1. *SimT(WN)*: top-layer similarity using label similarity measure based on WordNet description
2. *SimI(WN)*: inner-layer similarity using label similarity measure based on WordNet description

3. *SimT(TS)*: top-layer similarity using label similarity measure based on Topic Signature
4. *SimI(TS)*: inner-layer similarity using label similarity measure based on Topic Signature
5. *SimA*: atom layer similarity using atom-level similarity measures based on Information Content
6. *Weighted Sum (WN): weighted sum of* $w_A SimA + w_I SimI(WN) + w_T SimT(WN)$, *where* $w_A : w_I : w_T$ *= the ratio of the F-Measure of each individual measure and* $w_A + w_I + w_T = 1$
7. *Weighted Sum (TS)*: weighted sum of $w_A SimA + w_I SimI(TS) + w_T SimT(TS)$, *where* $w_A : w_I : w_T$ *= the ratio of the F-Measure of each individual measure and* $w_A + w_I + w_T = 1$

Figure 5 shows the quality measures Precision, Recall, and F-Measure of different mapping results by different similarity measures.

Figure 5. Experiment results of layered semantic similarity analysis

As shown in the Figure 5, the inner-layer (i.e., *SimI(WN)* and *SimI(TS)*) and atom-layer (i.e., *SimA*) measures by themselves generate poor results. This is because, as discussed earlier, the same set of atoms and intermediates can be used to produce semantically different elements (just like the same ingredients can be made into several kinds of dishes).

The overall performance is mixed. The *Weighted Sum (WN)* had the F-Measure 0.32 (precision 0.22 and recall 0.63) and Weighted Sum (TS) was slightly better (F-Measure 0.37, precision 0.25, and recall 0.71). The combination weights are currently pre-determined according to the ratio of the F-Measure of each individual measure. This result is certainly very encouraging considering how difficult the problem is even for experienced integrators (roughly 140 human hours were spent in mapping these 49 top elements in T&HE to AIAG-R schema by humans). However, a detailed examination of the results reveals that 13 manual mappings obtained by human integrators did not appear in any of the recommendations using either individual or combined similarity measures. This calls for further investigation.

Another phenomenon to be noted is that more weight should be given to the label similarities (top and inner layers). First, only one of the 22 matches found using atom-level similarity was not found by either of the two label-similarity measures. Second, the highest number of matches found by individual measure was using the top-layer measure.

Experiments and Results Enhanced by Grid Computing

The prototype system takes about 7 minutes to obtain the similarity analysis result between T&HE and AIAG-R schemas. To reduce the execution time, the prototype system is updated based on grid enabled MPJ tool. There were a total of 139 global (top) elements defined in the T&HE schema that needed to be mapped onto the set of 145 global components of the AIAG-R schema. Thus, the semantic distances of 139 x 145 (~20,000) pairs of elements needed to be examined.

We tested the execution time with different grid network environments which at maximum consist of 5 networked Pentium based Intel laptops to obtain the semantic similarity results using three different algorithms. Without the help of grid computing, the execution time was 420 seconds. By increasing the number of computers in the grid computing network, the execution time was reduced. Figure 6 shows that the execution time decreases exponentially as the number of processes increases. Note that, due to the networking management and communication cost, the speedup is diminishing when the number of machines in the grid increases beyond 4.

Figure 6. The number of machines vs. execution time

CONCLUSION

In this chapter, we introduced a layered semantic similarity analysis to facilitate XML schema mapping and a grid enabled service oriented architecture to enhance the efficiency of the semantic similarity analysis. We have implemented two prototype systems to evaluate the proposed approach. The first system is for the layered semantic similarity analysis which recommends for each element in a source XML schema a set of mapping candidates in a target schema based on the semantic similarity measures between the elements in these two schemas. The second system is for the efficient semantic similarity analysis using parallel and distributed computing technologies, developed based on MPJ and service oriented architecture. The proposed approach and the prototype systems have the potential to provide valuable assistance to the human integrators for the problem of XML schema mapping, merging and reuse.

A series of experiments were conducted with encouraging results. The system found a match to the human experts' mapping with match rate of 63% (31 out of the 49 manual mappings) in a real world application. This result is very encouraging considering how difficult the problem is even for experienced integrators. In addition, the second system showed significant improvements in performance (speedup of 46% when 4 machines were used in the grid).

The experiments also revealed that the problem is much more complicated than we initially thought. One observation is that the similarity scores vary greatly among the manual mappings (ranging from 0 to 1). This calls for further examination of similarity measures and the way they are combined and for exploring more elaborated mapping procedures. The following immediate steps are planned for future research.

1) Automatically determine the combination weights. Some machine learning techniques are under consideration, including regression and neural networks.
2) Increase the use of structural information. Our experiments show that labels at higher levels are more important than at lower ones. There is also evidence that the atom layer becomes more important when an element's structure is shallow. How to better incorporate the structural information into the semantic analysis will be investigated. Utilization of other features of the XML schema, such as cardinality and data type, will also be investigated.
3) Explore an iterative mapping procedure. The hypothesis is that the similarity measures for complex, difficult, or ambiguous elements will become more accurate when more mappings for other easier elements are established with each iteration.

Without proper tools, a harmonized international library of integration specifications such as that envisioned by the UN/CEFACT TBG17 is far-fetched. The number of data elements to harmonize can grow to hundreds of thousands, taking years, if possible at all, to yield usable integration results. The work discussed in this chapter shows promise to assist experts in accomplishing integration tasks more efficiently.

ACKNOWLEDGEMENTS

This work was supported in part by NIST award 60NANB6D6206.

REFERENCES

Asanovic, K., Bodik, R., Catanzaro, B. C., Gebis, J. J., Husbands, P., Keutzer, K., Patterson, D. A., Plishker, W. L., Shalf, J., Williams, S.W., & Yelick, K. A. (2006). *The landscape of parallel computing research: a view from Berkeley*, Technical Report No. UCB/EECS-2006-183, EECS Department, University of California, Berkeley.

Borthaku, D. *The Hadoop distributed file system: architecture and design*. Retrieved October 13, 2007, from http://hadoop.apache.org/common/docs/r0.17.2/hdfs_design.html.

Carpenter, B., Getov, V., Judd, G., Skjellum, A. & Fox, G. (2000). MPJ: MPI-like message passing for Java. *Concurrency: Practice and Experience*, *12*(11), 1019-1038. doi: 10.1.1.35.9869

Cover, T.M. & Thomas, J. A. (1991). Elements of information theory. *Wiley series in telecommunications*. Wiley, New York.

Dean, J. & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107-113. doi: 10.1145/1327452.1327492

Do, H.-H., Melnik, S., & Rahm, E. (2003). Comparison of Schema Matching Evaluations. *Lecture Notes in Computer Science*, 2593, 221-237. doi=10.1.1.11.4792

Foster, I. & Kesselman, C. (1997). Globus: a metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing, 11*(2), 115-128. doi: 10.1177/109434209701100205

Gal, A. (2006). Why is Schema Matching Tough and What Can We Do About It? ACM Sigmod Record, 35(4), 2-5. doi: 10.1145/1228268.1228269

Giunchiglia, F., Shvaiko, P., and Yatskevich, M. (2004). S-Match: an algorithm and an implementation of semantic matching. *Proceedings of the European Semantic Web Symposium (ESWS)*, 61–75

He, B. & Chang, K.C.C. (2004). A holistic paradigm for large scale schema matching. *SIGMOD Record, 33*(4), 20–25. doi: 10.1.1.58.7651

Jeong, B., Lee, D., Cho, H. & Lee, J. (2008). A novel method for measuring semantic similarity for XML schema matching, *Expert Systems and Applications*, 34(3), 1651-1658. doi: 10.1016/j.eswa.2007.01.025

Kim, J., Peng, Y., Kulvatunyou, B., Ivezik, N. & Jones, A. (2008). A layered approach to semantic similarity analysis of XML schemas. *Proceedings of the IEEE International Conference on Information Reuse and Integration*, 274-279. doi: 10.1109/IRI.2008.4583042

Lerner, B.S. (2000). A model for compound type changes encountered in schema evolution. *ACM Transactions on Database Systems, 25*(1), 83-127. doi: 10.1.1.105.1542

Lin, C.Y. & Hovy, E.H. (2000). The Automated Acquisition of Topic Signatures for Text Summarization. *Proceedings of the COLING Conference*. Strasbourg, France. doi: 10.3115/990820.990892

Lin, D. (1998). An Information-theoretic definition of similarity. *Proceedings of the 15th International Conference on Machine Learning*, 296-304. doi: 10.1.1.55.1832

Madhavan, J., Bernstein, P.A. & Rahm, E. (2001). Generic schema matching with Cupid. *Proceeding of the 27th International Conference on Very Large Data Bases,* 49-58. doi: 10.1.1.17.4650

Melnik, S., Garcia-Molina, H., & Rahm, E. (2002). Similarity flooding - a versatile graph matching algorithm. *Proceeding of 18th International Conference of Data Engineering,* 117-128. doi: 10.1.1.61.4266

Miller, G. A. (1995). WORDNET: a lexical database for English. *Communications of ACM, 38*(11), 39-41. doi: 10.1145/219717.219748

Miller, R. J., Ioannidis, Y. E., & Ramakrishnan, R. (1994). Schema equivalence in heterogeneous systems: bridging theory and practice. *Information Systems, 19*(1), 3-31. doi: 10.1007/3-540-57818-8_42

Milo, T. & Zohar, S. (1998). Using schema matching to simplify heterogeneous data translation. *Proceeding of the 24th International Conference on Very Large Data Bases,* 122-133. doi: 10.1.1.30.2620

Peng, Y. (2006). On semantic similarity measures. *Technical Report from Syllogism.Com to NIST*. Retrieved October 28, 2007, from http://www.aiag.org

Qin, P., Lu, Z., Yan, Y., & Wu, F. (2009). A New Measure of Word Semantic Similarity Based on WordNet Hierarchy and DAG Theory, *Proceedings of International Conference on Web Information Systems and Mining*, 181-185, doi: 10.1109/WISM.2009.44

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transaction on Systems, Man, and Cybernetics*, *19*(1), 17-30. doi: 10.1109/21.24528

Rahm, E. & Bernstein, P.A. (2001). A survey of approaches to automatic schema matching. *The International Journal on Very Large Data Bases*, *10*(4), 334-350. doi: 10.1007/s007780100057

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 448-453. doi: 10.1.1.55.5277

Saleem, K., Bellahsene, Z. & Hunt. E. (2008). PORSCHE: performance oriented schema mediation, *Information Systems, 33*(2), 637-657. doi: 10.1016/j.is.2008.01.010

Shvaiko, P. & Euzenat, J. (2005). A survey of schema-based matching approaches. *Journal on Data Semantics IV, LNCS 3730*, 146-171. doi: 10.1007/11603412_5

Stephenson, D. (2004). *XML Schema Best Practices*, HP Dev Resource, Retrieved from http://xml.coverpages.org/HP-StephensonSchemaBestPractices.pdf

UN/CEFACT TBG17. *Harmonisation workgroup*. Retrieved April 22, 2007, from http://www.uncefactforum.org/TBG/TBG17/tbg17.htm.

Van Rijsbergen, C. J. (1979). *Information retrieval (2nd ed.)*. London: Butterworths.

Yang, D. & Powers, D.M.W. (2005). Measuring semantic similarity in the taxonomy of WordNet. *Proceedings of the 28th Australasian Computer Science Conference*, 315-322.

KEY TERMS & DEFINITIONS

XML schema: a description to define the structure, content and semantic information of XML documents

Business document: the name for messages exchanged between trading partners in the e-Business environment.

e-Business integration: a type of integration process that enables integrating trading partners with other trading partners to exchange business document across networks such as the Internet and incorporates the trading partners' host applications and business processes.

Semantic information: a type of informational content that contains the meaning of data

Schema mapping: a process to identify how information can be shared between heterogeneous schemas and how they can be mapped, merged, or reused for integration and interoperability of their e-Business systems.

Semantic similarity analysis: a process to analyze some degree of symmetry in either analogy and resemblance between two or more concepts, terms, or documents based on the likeness of their meaning / semantic information

Information content: a type of information that can be quantified by information theory in a sense that, as probability increases, informativeness decreases, so the more abstract a concept is, the lower its information content will be.

Parallel Computing: a computation technology in which multiple concurrent processes work simultaneously and cooperate with each other for a single task.

Distributed Computing: a computer science technology that a development of applications execute on different computers interconnected by networks.