A Layered Approach to Semantic Similarity Analysis of XML Schemas

Jaewook Kim[1,2], Yun Peng[1], Serm Kulvatunyou[2], Nenad Ivezic[2], and Albert Jones[2]
[1]*Department of Computer Science and Electrical Engineering*
University of Maryland, Baltimore County
[2]*National Institute of Standards and Technology*
{jaewook, nivezic, jonesa}@nist.gov, ypeng@umbc.edu, and kbserm@psualum.com

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 schema-mapping researches have not fully utilized semantic information. In this paper, we propose a semantic similarity analysis approach to facilitate XML schema mapping, merging and reuse. Several key innovations are introduced, including 1) a layered semantic structure for XML schemas; 2) layered similarity measures based on actual information content; and, 3) an approach for integrating similarities at all layers. Experimental results, using two different schemas from the Automotive Industry Action Group (AIAG), demonstrate the value of these innovations.

Keywords: XML Schema, e-Business Integration, Schema Mapping, Merging, Reuse, Similarity Measure, Information Content

1. Introduction

Schema mapping, merging, and reuse are critical when integrating independently developed, heterogeneous e-Business applications. Typically, these capabilities are performed manually; this is very labor-intensive, costly, and error-prone [1]. Many computerized mapping tools have been proposed [2], but they often fail to analyze thoroughly and utilize fully the semantic information in the XML schema.

In this paper, we introduce a semantic similarity analysis approach, which we believe will facilitate both mapping and reuse. We have also developed some computerized tools that implement this approach with real-world application data. Our approach includes three major innovations: a layered semantic structure for XML schemas; layered similarity measures that use information content in those schemas; and, an approach for integrating similarities measures across all layers.

Our approach uses a recommended a set of data elements in the target schema as likely mapping/merging candidates for each element in the source schema. That recommendation is based on the values of their semantic similarity measures. Those measures attempt to quantify the semantic distance between pairs of data elements [3].

To show the potential benefit of our approach, we conducted a series of experiments using schemas from two different workgroups at the Automotive Industry Action Group (AIAG) [4]. These experiments produced encouraging results and suggested several directions for further performance improvement.

The rest of the paper is organized as follows. Section 2 provides the background of this research. It includes a brief review of selected existing similarity metrics and an introduction to the real-world integration experiments. Sections 3 and 4 give detailed descriptions of the proposed approach. Section 5 reports the experimental results. Finally, Section 6 concludes with directions for future research.

2. Background

The common approach in integrating the heterogeneous e-Business applications is to provide adapters that translate data from native specifications to an interlingua. That interlingua is frequently a standard whose structure and semantics are supposed to be agreed-upon and understood by all parties involved. The difficulty is that these agreements and understandings are never complete. Furthermore, the differences often depend on the actual context of the integration. Relevant techniques for finding and resolving these differences - such as semantic markups using domain ontologies – are not mature enough for industrial use. Instead, industrial practitioners increasingly rely on an XML schema representation for the standards. In the automotive industry for example, practitioners use XML schemas called BODs (Business Object

Documents), which were developed by the OAG (Open Application Group) [5].

It is widely known that the semantics of the information contained within such standard schemas are not defined formally. Rather, they are defined implicitly by the meanings of English words or phrases that appear in the names of the components and fields, as well as in associated descriptions. Descriptions are especially problematic because there are no clearly documented, common used approaches to associate and specify them.

For these reasons, it is still difficult and costly to use such standard XML schemas as the basis for application integration. It is even hard to identify the reusable components within these standards and to understand how and when to use them. Consequently, for any particular integration effort, mapping to and from the proprietary representation of the application to the standard representation is still mostly a manual operation.

To deal with these deficiencies, users often tailor or customize existing standards by adding new, typically duplicating or overlapping components, rather than attempting to reuse existing ones [6, 7]. The result is the proliferation of de facto standards that have duplicate or overlapping semantics structured in different ways. This, of course, further increases the cost of integration and, more importantly, creates the need for mergers.

In this paper, we use three key terms for XML schema integration: mapping, merging, and reuse. They refer to three closely related but different integration tasks. *Mapping* is a task in which one identifies how information in one format is populated into another format. *Reuse* is a task where one looks for existing integration specifications to use in a new integration project. *Merging*, perhaps the most time-consuming of the three tasks, seeks to combine two or more specifications into a one. All of these tasks rely a single underlying capability, identifying semantically similar data elements in two different schemas.

2.1. Similarity measures and related works

Various approaches, based on a notion of similarity, have been developed recently to implement the aforementioned tasks. The simplest approach to semantic similarity is linguistics based. It uses a metric that computes the similarity between element names or descriptions using a string matching algorithm [8]. Many such algorithms exist including the widely used Jaccard [9] and cosine similarity [10, 11] measures. Other straight forward approaches have been developed based on a linguistic taxonomy [12], such as the popular WordNet [13]. One can use such taxonomy to obtain more accurate and less ambiguous semantics for words in the element names.

A more complicated approach is based on structural similarity measures. On such approach in common use is based on the path length between two entities in a taxonomy. These approaches typically fail to take into account the different roles played by the entities and the relative and importance of their relationships to one another. A new approach, based on information content (IC), was proposed to address this problem [14, 15].

IC approaches measure the similarity between two entities x and y, based on another metric called common(x, y). This second metric is based on how much information is needed to describe the commonality between x and y. Here, x and y can be two words, two objects, or two structures. Commonality can be based on the features or hypernyms two words share. Using to information theory, then, and the degree of specificity can be measured by the information content of common(x, y) – namely, $\log(P(\text{common}(x, y))$.

The IC based similarity metric was first proposed in [15] and applied to semantic similarity between words in the WordNet. The common(x, y) is defined as the most specific hypernym C, and the similarity is given as

$$Sim(x, y) = I(C) = -\log P(C) \quad (1)$$

where $I(C)$ is the information content of C, and the probabilities were calculated as word frequencies in a corpus. Research in [16] compares the differences between the IC and structural approaches in measuring similarity between elements in a single XML schema. It shows better results can be achieved by combing the two approaches.

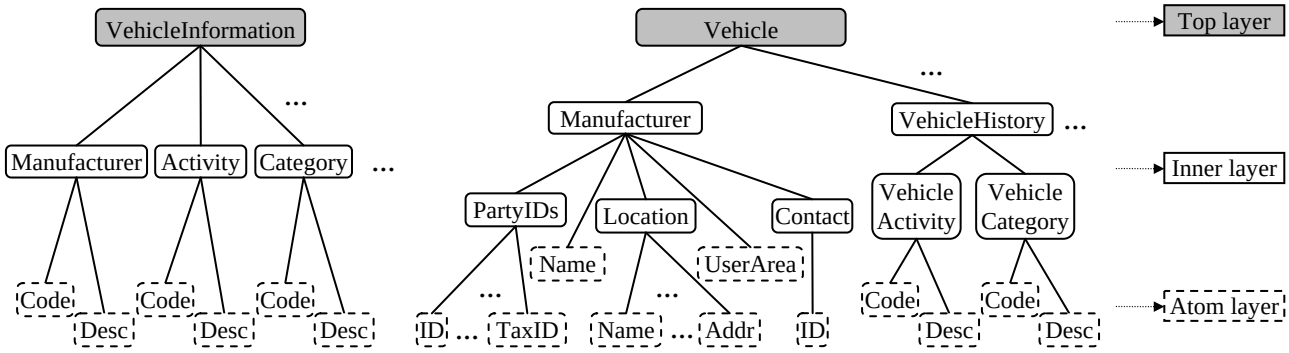
Each of the existing similarity metrics has its strengths and weaknesses; and, each typically only makes use of part of the available semantic information. In this paper, we propose an innovative approach that employs a variety of similarity metrics, including lexical-, taxonomical-, and IC- based.

2.2. Experimental data

To test and evaluate our proposed approach, we obtained schemas and manual mapping data from two workgroups at the Automotive Industry Action Group (AIAG): the Resource (RES) group and the Truck and Heavy Equipment (T&HE) group. We used the RES schema as the target and the T&HE schema as source. Both schemas are based on the OAG schema [5] and have overlapping concepts. However, they define some elements quite differently (see Figure 1). In Figure 1, the RES top-level concept "Vehicle" and T&HE top-level concept "VehicleInformation" are intended to describe the same object. Nevertheless, the two concepts have different labels (names) and quite different data structures.

At the component level, there are a 139 global elements defined in the T&HE schema. These elements must be mapped to the set of 145 global elements of RES schema. Human experts spent roughly 140 hours developing this mapping, which required an examination of 139

x 145 (~ 20,000) pairs of elements. Additional time was then required to merge these two schemas at the message level. This is an indication that manual mapping ad merging can be very time consuming.



(a) T&HE's Vehi- (b) RES's Vehicle

Figure 1. Three layers of XML schema

3. Layers of XML schema structure

An XML Schema defines a set of global elements, each of which can be represented as a tree of linked nodes. Each node in a tree has zero or more child nodes and zero or one parent node. We can classify the nodes into three types: root node, intermediate nodes, and, leaf nodes. The root node has no parents. Intermediate nodes have both parents and children. Leaf nodes, commonly called atoms, have no children. This means that atoms cannot be divided any further.

Each tree can thus be divided into three layers: the top layer, containing the single root of the tree; the inner layer, containing the intermediate nodes; and, the atom layer, containing the leaf nodes. Each layer typically captures the semantics from its own perspective. A top layer node, through the linguistic information contained in its label and namespace, specifies the data object that the global element is intended to describe. Nodes in the atom layer indicate the atomic elements (XML schema attributes, simpleType, and simpleContent) needed to describe the global element. The inner layer provides structural information by specifying how atomic elements are grouped and related. The linguistic information in the labels of both atomic and inner nodes may also help to qualify the semantics of the global element.

Consider the two global elements defined in T&HE and RES schemas in Figure 1. The labels in their top layer nodes indicate both of them intend to represent the same vehicle object. However, their designers think quite differently about what atomic elements are needed (see their different atom layers) and how they 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. On the other hand, the same set of ingredients (atoms) can produce elements of different semantics depending on how they are cooked (structured) or packaged (what the top layer node is). For example, several party elements, such as CustomerParty, DealerParty, and SellingParty, may all contain the same atoms and intermediates, but they are intended for semantically different data objects.

4. Similarity measures

The complex relationships between nodes at different layers require (1) layer-specific semantic analysis tools and (2) a mechanism to combine the outputs from those tools. For this reason, we have developed two similarity measures. The first one, called *atom-layer similarity*, measures the similarity between two atom layers of the two elements. The second one, called *label similarity*, measures the similarity between the labels (names). This measure can compare two labels when applied to a pair of top layer nodes. It can also compare two sets of labels when applied to two inner layers (if they are not empty).

4.1. Atom level similarity

Not every atom has equal weight in determining the semantic similarity. Two elements that share a widely-used atom will not be as similar as two elements that share a rarely-used atom [14, 15]. To account for the degree of importance of individual atoms, we developed an IC based measure for atom-layer similarity. Specifically, let $A(x)$ and $A(y)$ denote the sets of atoms of elements x and y, respectively. Then, the atom-layer similarity between x and y is defined as

$$sim_A(A(x), A(y)) = \frac{2 \times \sum_{c_i \in A(x) \cap A(y)} I(c_i)}{\sum_{c_i \in A(x)} I(c_i) + \sum_{c_j \in A(y)} I(c_j)} \quad (2)$$

The probability of each atom is taken as its frequency using the corpus formed by all labels in both T&HE and

RES schemas. The statistics of atoms is given in the table below.

Table 1. Statistics of atoms in the two schemas

	RES Schema		T&HE Schema	
Total # of Atoms	67688		53812	
# of Unique Atoms	793		825	
	non-OAG	OAG	non-OAG	OAG
	90	703	119	706

Eq. (2) is based on the assumption that the source and target schemas share a significant number of atoms. This is the case for RES and T&HE schemas (as can be seen in Table 1, 7 out of 8 atoms in the two schemas are defined in the OAG schema). In this case, we simply treat two atoms as completely similar (with similarity score 1) if they have the same label, and completely dissimilar (score 0) if they do not. This won't work if $A(x)$ and $A(y)$ do not share common atoms but rather have atoms that are semantically similar. In that case, one can multiply each $I(c_i)$ in the numerator of Eq. (2) by a similarity score between the source atom c_i and its most similar target atom. Details of one such measure can be found in [3].

4.2. Label similarity

The label or name x of a node is a word or concatenation of words (or their abbreviations). Before similarity can be compared, a pre-process called "label normalization" is conducted to obtain full words from the concatenations and abbreviations, denoted as $L(x)$. For example, L(VehicleInformation) = {vehicle, information}. To better ascertain the semantics of these words and to deal with the problem of synonyms, we expand each word by its description from the WordNet, denoted $d(x_i \in L(x))$.

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 from the WordNet. For this we require that all $W(x)$ be normalized to the same lengths, say G words. Secondly, words in $L(x)$ are not equally important in defining x's semantics (For example, "vehicle" is certainly semantically more important than "information" in the label "VehicleInformation"). Semantic analysis using advanced techniques such as noun-phrase analysis from natural language processing is complex and time consuming. Instead, we measure the importance of each word x_i by its information content $I(x_i)$ and require that the vector $W(x)$ be formed in such a way that the number of words from description $d(x_i)$ is proportional to $I(x_i)$.

For example, suppose the vector length G = 10; I(vehicle)/I(information) = 4; and descriptions d(vehicle) = (a b c d) and d(information) = (r s t). To satisfy both constraints, we would have

W(VehicleInformation) = (a b c d a b c d r s)

where d(vehicle) is duplicated and d(information) truncated.

Finally, the similarity of labels x and y will be measured by the cosine of the two vectors $W(x)$ and $W(y)$ [10].

The procedure of label similarity is outlined below: For labels x and y:

1) Normalize x and y to obtain full words $L(x)$ and $L(y)$;
2) Calculate the semantic weight of each word $L(x)$ and $L(y)$ by
$$w_{IC}(x_i) = \frac{I(x_i)}{\Sigma_{x_k \in L(x)} I(x_k)},\ w_{IC}(y_j) = \frac{I(y_j)}{\Sigma_{y_k \in L(y)} I(y_k)} \quad (3)$$
where $I(x_i) = -\log P(x_i)$, and the probabilities of x_i and y_j are taken as their frequencies in their respective schema;
3) Obtain from the WordNet the description of each word in $L(x)$ and $L(y)$, remove most of the stop words from the descriptions [17], make each description a set of words of size $G * w_{IC}(x_i)$ by duplicating or truncating the description, and take a union (keeping all duplicates) of all these sets to form $W(x)$ and $W(y)$;
4) Measure the similarity of x and y by calculating *cosine*($W(x)$, $W(y)$) by
$$Sim_T(x,y) = \frac{W(x)\cdot W(y)}{|W(x)||W(y)|} = \frac{\sum_{i \in W(x) \cap W(y)} f_x(i) * f_y(i)}{\left(\sum_{i \in W(x)} f_x(i)^2\right)^{1/2} \left(\sum_{j \in W(y)} f_y(j)^2\right)^{1/2}} \quad (4)$$

where $f_x(i)$ is the frequency of the term 'i' in $W(x)$

Label similarity for intermediate nodes is measured in the same way and denoted as $Sim_I(x,y)$. In this case, the union of labels of all intermediate nodes of a tree is used for x and for y.

4.3. Combined similarity score

Several approaches for combining individual similarity measures (Sim_A, Sim_I, Sim_T) have been experimented with. They include average(a, b, c), max(a, b, c), additive $(1 - (1 - a)(1 - b)(1 - c))$, and weighted sum. The weighted sum seemed to work the best in the experiments:

$$Sim(x,y) = w_A Sim_A + w_T Sim_T + w_I Sim_I \quad (5)$$

where $w_A + w_T + w_I = 1$.

Among other things, this method allows us to adjust the weights to best reflect the importance of measures at individual layers.

5. Experimental results

We implemented a prototype system that not only produces Sim_A, Sim_I, and Sim_T as given in Eqs. (2) and (4), but also provides several combination rules, including Eq. (5). It normalizes all labels in both top and inner layers, and calculates the IC value of all words from these labels by obtaining their frequencies from the two schemas.

We conducted a series of experiments using the schemas shown in Figure 1. The 49 manual mappings produced by human integrators are used as the basis to evaluate the performance of the system. For each of the 49 T&HE global elements, the system recommends the 5 most similar RES elements according to a similarity measure. Note, we evaluate performance using a set rather than a single recommendation, because our objective is not to fully-automate the process but rather to assist human experts. A recommendation is considered a match if it contains the manual mapping. Results from using various similarity measures, individual and combined, were obtained and reported in the table below.

Table 2. Experiment results

Similarity measure	# matches
Sim_T	35
Sim_I	8
Sim_A	22
$Sim_T \bigcup Sim_I$	35
Weighted sum	31

Evidently, atom-level and intermediate-level measures alone give poor results because, as discussed earlier, the same set of atoms and intermediates can be used to produce several semantically different elements (just like the same ingredients can be cooked into several kinds of dishes).

The overall performance is mixed. The weighted sum leads to about 63% matches, 31 out of the 49 manual mappings. (The combination weights are currently predetermined according to the ratio of the number of matches in each individual measure.) This result is certainly very encouraging considering how difficult the problem is even for experienced integrators. However, detailed examination of the results reveals that 13 manual mappings did not appear in any of the recommendations using either individual or combined similarity measures. This calls for further investigation.

We further that more weight needs to be given to the label similarities. First, only one of the 22 matches found using atom-level similarity is not found by either of the two label-similarity measures. Second, the highest number of matches found by individual measure is using the top-layer measure. Lastly, the cosine method using the combined top and intermediate labels found 35 matches (4 of them are different from those obtained using the weighted sum combination).

6. Conclusions and plan for future research

In this paper, we have proposed an innovative approach for comparing XML schemas that exploits their imbedded semantic information. This approach divides data elements into layers and measures semantic similarity based on those layers. The output from this approach will be a set of mapping candidates in a target schema for each element in the source schema. These candidates will be selected based on the semantic similarity measures between the elements in the two schemas. We have also implemented a prototype system to evaluate the proposed approach. The proposed approach and prototype system have the potential to provide valuable help for the humans attempting to integrate applications based on different schemas.

We conducted a series of experiments and have reported their results. Those results were mixed. The system found correct matches about 60% of the time. The scores associated with those matches, however, varied greatly. This calls for further examination of the similarity measures, the way they are combined, and for more elaborated mapping procedures. The following immediate steps are planned for future research.

1) Determine the combination weights automatically. Some machine learning techniques are under consideration, including regression and neural networks.
2) Increase the use of structural information. Our experiments show that higher-layer labels are more important than the lower ones. There is also evidence that the atom layer becomes more important when structure of the element is shallow. Methods to better incorporate the structural information in the semantic analysis will 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 after mappings for other easier elements are established. For example, atoms defined in the T&HE schema (not in the OAG schema) are currently considered with zero similarity with any atoms in the RES schema. This will be rectified if we map them first; and, atom-layer similarity for other elements in the subsequent iterations will be improved.

Acknowledgements

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

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.

References

[1] E. Rahm and P.A. Bernstein, "A Survey of Approaches to Automatic Schema Matching", *VLDB Journal*, volume 10, issue 4, 2001, pp. 334-350.

[2] P. Shvaiko and J. Euzenat, "A Survey of Schema-Based Matching Approaches", *Journal on Data Semantics IV*, LNCS 3730, 2005, pp. 146-171

[3] Y. Peng, "On Semantic Similarity Measures", Technical Report from Syllogism.Com to NIST, 2006.

[4] Automotive Industry Action Group (AIAG) Website, http://www.aiag.org

[5] The Open Application Group, "Open Application Group Integration Specification", version 8.0. 2002.

[6] N. Anicic, N. Ivezic, and A.T. Jones, "An Architecture for Semantic Enterprise Application Integration Standards", in Proceedings of the 1st Conference on Interoperability of Enterprise Software and Applications, Geneva Switzerland, 2005.

[7] B. Kulvatunyou, N. Ivezic, and A.T. Jones, "Content-Level Conformance Testing: An Information Mapping Case Study", in Proceedings of TestCom 2005, Montreal, Canada, May 31 - June 2, 2005, pp. 349-364.

[8] H. H. Do and E. Rahm, "COMA - A System for Flexible Combination of Schema Matching Approaches", in Proceedings of the Very Large Data Bases Conference (VLDB), 2001, pp 610–621.

[9] Jaccard similarity, http://www.dcs.shef.ac.uk/~sam/stringmetrics.html#jaccard

[10] Cosine similarity, http://www.dcs.shef.ac.uk/~sam/stringmetrics.html#cosine, and http://www.codeproject.com/useritems/Cosine_Similarity.asp

[11] B. Jeong, B. Kulvatunyou, N. Ivezic, H. Cho, and A.T. Jones, "Enhance reuse of standard e-business XML schema documents", in Proceedings of international workshop on contexts and ontology: theory, practice and application (C&O'05) in the 20th national conference on artificial intelligence (AAAI'05), 2005.

[12] D. Yang and D.M.W. Powers, "Measuring Semantic Similarity in the Taxonomy of WordNet", in the 28th Australasian Computer Science Conference (ACSC2005), Newcastle, Australia, 2005, pp. 315-322.

[13] WordNet, http://wordnet.princeton.edu/man/wnstats.7WN

[14] D. Lin, "An Information-Theoretic Definition of Similarity", in Proceedings of International Conference on Machine Learning, Madison, Wisconsin, July, 1998.

[15] P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy", in Proceedings of the 14th International Joint Conference on AI, Montreal, CA, 1995, pp. 448-453.

[16] A.Formica, "Similarity of XML-Schema elements: a structural and information content approach", *The Computer Journal*, volume 51, issue 2, 2008, pp. 240-254.

[17] English Stopwords List Website, http://www.ranks.nl/tools/stopwords.html