

Enhance Reuse of Standard e-Business XML Schema Documents

Buhwan Jeong¹, Boonserm (Serm) Kulvatunyou^{2*}, Nenad Ivezic², Hyunbo Cho¹, Albert Jones²

¹Pohang University of Science and Technology, San 31, Hyoja, Pohang, 790-784, South Korea

²National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD 20899, USA, *Corresponding Author

Abstract

Ideally, e-Business application interfaces would be built from highly reusable specifications of business document standards. Since many of these specifications are poorly understood, users often create new ones or customize existing ones every time a new integration problem arises. Consequently, even though there is a potential for reuse, the lack of a component discovery tool means that the cost of reuse is still prohibitively high. In this paper, we explore the potential of using similarity metrics to discover standard XML Schema documents. Our goal is to enhance reuse of XML Schema document/component standards in new integration contexts through the discovery process. We are motivated by the increasing access to the application interface specifications expressed in the form of XML Schema. These specifications are created to facilitate business documents exchange among software applications. Reuse can reduce both the proliferation of standards and the interoperability costs. To demonstrate these potential benefits, we propose and position our research based on an experimental scenario and a novel evaluation approach to qualify alternative similarity metrics on schema discovery. The edge equality in the evaluation method provides a conservative quality measure. We review a number of fundamental approaches to developing similarity metrics, and we organize these metrics into lexical, structural, and logical categories. For each of the metrics, we discuss its relevance and potential issues in its application to the XML Schema discovery task. We conclude that each of the similarity measures has its own strengths and weaknesses and each is expected to yield different results in different search situations. It is important, in the context of an application of these measures to e-Business standards that a schema discovery engine capable of assigning appropriate weights to different similarity measures be used when the search conditions change. This is a subject of our future experimental work.

An Experimental Scenario and Evaluation

Experimental Scenario

We propose a Schema Discovery Engine that applies different combinations of similarity metrics to one or more relevant, standard, document (component) schemas that may satisfy given integration requirements. Figure 1 illustrates the experimental evaluation planned for our Schema Discovery Engine running a similarity metric.

We will use test data from a real, industrial integration problem involving B2B data exchange. The component library, stored in the repository on the left-hand side of the figure, will be based on the Open Applications Group Integration Specification (OAGIS) [3]. The OAGIS is a horizontal standard for business data exchange including supply chain data. Sample data exchange requirements (originally captured in a class model or SQL statements) will be taken from the Automotive Industry Action Group (AIAG) supply chain standards and the Oracle ERP interfaces as shown on the right-hand side of the figure. To facilitate our computing environment, the data exchange requirements will be translated into a common syntax such as an XML Schema or a pseudo XML instance. Those requirements have maps, with possible extensions, to the OAGIS. The maps are considered to be correct and will be compared against discovered components in the evaluation phase.

Figure 1 shows how this might work from the requirement in the AIAG Min/Max Vendor Managed Inventory scenario called the *QuantityOnHand* (QOH) [4]. The *QOH* model indicates that the required data fields are *Item*, *SiteId*, *Quantity*, *MinQuantity*, and *MaxQuantity*. A user who has the model of this component searches the library for reusable components. The schema discovery engine uses the QOH model information and any other relevant documentation to calculate similarities against the components in the library. It can evaluation options that consists of several combinations of different types of similarity measures to determine the best potential matches. The user can choose different combination options, such as the Harmonic mean, and set the threshold that determines how many and which kinds of components are returned.

In the figure, the *InventoryBalance* and the *WIP* components whose overall similarity values are above the threshold, 0.7, are returned. Within each final result, individual similarities are computed indicating the strength of the mapping between each field within the discovered component and each field in the requirement. Within the illustration, the discovery engine might not be able to identify any fields with sufficiently high similarity measures to induce equivalences for the *MinQuantity* and the *MaxQuantity* fields; however, it could indicate that the two fields could establish some relationships with the *Quantity* field. The relationships may include equivalent, more (or less) general, and overlapping [1].

We expect that the results from such an analysis could guide users by making better and more efficient judgments about the potential reuse of existing schemas. In Figure 1,

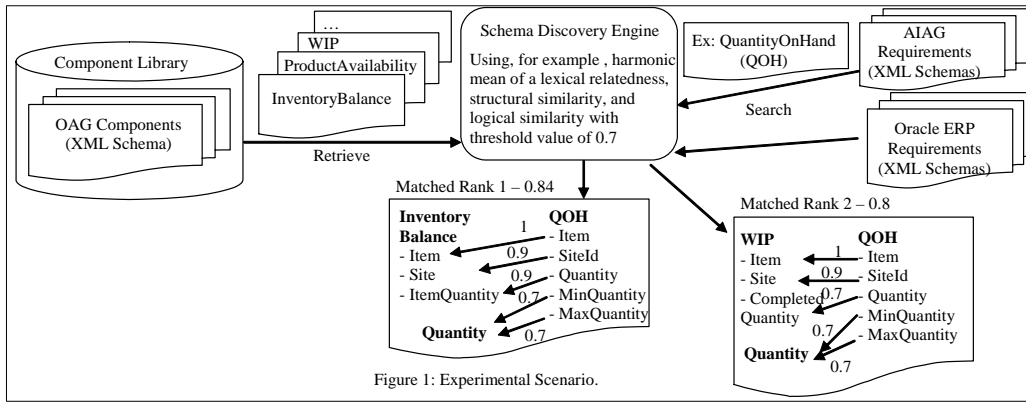


Figure 1: Experimental Scenario.

the result could be interpreted to mean that the *QOH* may be designed appropriately as an extension of the *InventoryBalance* components where the *MinQuantity* and *MaxQuantity* are the extensions of the basic *Quantity* field. Additionally, the discovery result also points to the *WIP* component as a possible basis for the *QOH* component. It is important to note that extensions to existing components should be added to the component library so that they could be discovered and reused in subsequent integration activities.

Experimental Evaluation

In the previous section, we said that the schema discovery engine selectively uses and combines similarity metrics. In this section, we illustrate an example approach to evaluate and compare the schema discovery quality in different combinations of similarity measures.

Since the components in the library and the requirements are represented using an XML tree-based structure, we argue that each data field, either element or attribute, can be addressed using the XPATH expression [2]. In Figure 1, we can address the fields as *InventoryBalance/Item* or *QOH/Item* and *InventoryBalance/Site* or *QOH/SiteId*, for example.

Let a set $U = \{u_i\}$, $i = 1, 2, \dots, n$ be a set of XPATH expressions, u_i , for each element or attribute field of the target component (the requirement) U . Similarly, let a set $V = \{v_j\}$, $j = 1, 2, \dots, m$ be that of the true mapped component(s) from the library, and a set $W = \{w_k\}$, $k = 1, 2, \dots, p$ be that of a discovered component.

Then, let a set of edge constraint $E_t = \{e_t = (u_{i'}, v_{j'})\}$, $i' = 1, 2, \dots, n'$ and $j' = 1, 2, \dots, m'$, $n' \leq n$ and $m' \leq m$ be the true map from the fields in U to V . Similar to E_t , let $E_d = \{e_d = (u_{r'}, w_{k'})\}$, $r' = 1, 2, \dots, r'$, $k' = 1, 2, \dots, p'$, $r' \leq n$, and $p' \leq p$ be the map from U to W as estimated by the discovery engine. A graphical representation of E_d is as shown in the output from the schema discovery engine in Figure 1. As shown in the figure, there may be some fields in U that have no map to any field in W (e.g., the *MinQuantity* and the *MaxQuantity* fields). Alternatively, components may be discovered for which some of the fields cannot be mapped (used) onto the required fields. It should be noted that in practice, $u_{i'}$ ($w_{k'}$) can be either a u_i (w_k) or a composition of two or more u_i (w_k). Although not

shown, a graphical representation of E_t and relationship between $v_{j'}$ and v_j are similar to that of E_d . Using the above definitions and typical information retrieval measurements, we can measure the quality of a discovered component d ,

using *Jaccard* as $Q_d = |E_t \cap E_d| / |E_t \cup E_d|$ or

using *Recall* as $Q_d = |E_t \cap E_d| / |E_t|$ or

using *Precision* as $Q_d = |E_t \cap E_d| / |E_d|$

where two edges $e_t \in E_t$ and $e_d \in E_d$ are matched, i.e., $e_t = e_d$ if and only if the paths $u_{i'} = u_{r'}$ and $v_{j'} = w_{k'}$.

If the schema discovery engine returns multiple components, an example of the overall discovery quality could be $Q_D = \max_{d \in D} (Q_d)$, where D is a set of discovered components.

There are some issues with this discovery quality measure. First, the edge equality definition makes this quality measure a conservative one, because it requires that the labels on both paths be identical. For a discovered component where some labels are different yet semantically equal, the quality would be unrealistically low.

Second, the discovery quality (Q_D) may not indicate the performance of the overall system, if our intent is to consider multiple alternatives. In that case, the measure has to be normalized against the number of suggestions returned. In addition, the discovered component providing the maximum quality, $\max(Q_d)$, may not be the component ranked the most similar by the discovery engine. Thus, we want the discovery engine to produce a similarity value that is highly correlated with the quality associated with the component. We envision the correlation value between the similarity value of a component d and Q_d across the members of the set of discovered components D as a dimension of the overall discovery quality. The advantage of such quality measure is that it is orthogonal to the number of discovered components.

Similarity Metrics: A Literature Review

We organize the review of similarity approaches into three groups: lexical, structural, and logical. For each of the approaches, we discuss its relevance and potential issues in applying it to the XML Schema discovery task. In closing this section, we give our perspective on the respective roles and potentials of the investigated categories of

similarity metrics, both considered individually and in combination.

Lexical Perspective

A lexical similarity measure quantifies the commonality between individual component names using purely lexical information. Commonly used lexical similarity measures include *Tanimoto* [19], n-gram [18], (weighted-) distance-based [5, 6, 13, 14], word sense-based [1], and information content-based [7] metrics.

We found that the existing lexical similarity measures may not be directly applicable to our schema discovery problem. The reason is that an XML component name usually consists of several words and/or allowable abbreviations concatenated to enhance their expressivity. Such composite words (e.g., *QuantityOnHand*, *InventoryBalance*) provide more information than individual words because the additional words provide additional context information. Moreover, the composite words make the meaning of the included words more specific. This information is particularly important when a domain-specific lexical resource is not available. For example, we can eliminate several senses associated with the term *Contact* within the component name *DeliveryToContact*. Because *Contact* follows the verb, it must be a noun. Further, the *relationship* and the *surface* senses of the *Contact* can be eliminated because one would not deliver a product to a relationship or a surface in the business sense. Hence, the similarity measure should be constructed to focus on the meaning of the *Contact* associated with a person. Furthermore, we envision that each word in a component name should have different salience depending on its part of speech. For example, we would like the component name *DeliveryToContact* to have a higher similarity value when comparing it to the *ShipToContact* than to the *DeliveryFromContact* since the latter is, in fact, an opposite. The research to advance the lexical similarity measures for the schema discovery should exploit this type of additional information.

We also recognize that domain-specific resources are very important in analyzing lexical similarity. Consequently, our future research may include methods to model domain-specific resources in our supply-chain and logistics problem contexts. In addition, the schemas and requirements documentations are context specific resources for the content-based similarity analysis.

Structural Perspective

A structural similarity measure quantifies the commonality between components by taking into account the lexical similarities of multiple, structurally related sub-components of these terms (e.g., child components, child attributes). A structural similarity metric typically provides a more conservative measure than the lexical similarity, because it looks beyond the individual labels and their definitions to the context surrounding these labels. The tree structure is a native structure for XML documents; hence,

it is most related to our problem context. While significant research has been done to apply these methods to XML instance documents, they may be applied to schema discovery by representing the XML schema using one or more pseudo XML instances. Commonly used structural similarity measures include node, path, and/or edge matching, tree edit distance (TED) [8, 9, 12], (weighted) tag similarity [9], weighted tree similarity [10], and Fourier transformation-based approach [15].

Although existing structural similarity measures can be useful in schema discovery, there are several issues that need to be addressed. First, the existing measures are geared toward content rather than meta-data; hence, the perspective of these approaches needs to be adjusted.

Second, one of the most powerful structural measures, TED, is more applicable to ordered trees because this insures computability in polynomial time. However, the order constraint does not always apply to schemas; hence, further research is required to determine conditions under which this restriction can be relaxed. One possible approach is to re-order and represent schemas in abstract tree structures. Another is to ignore the structure in local areas and aggregate them into a single node. The less powerful measures such as path or inclusive path matching do not exploit fully context-specific information embedded in the structural relationships. The weighted measures require a practical way to obtain weights.

Logical Perspective

A logical similarity measure quantifies the commonality of properties/constraints restraining components definitions beyond the lexical and structural aspects such as type, cardinality, etc. The logical similarity is often classified as a structural category [18, 19]. However, we treat it as an independent category because it is the most restrictive and accurate measure. That is, even if two components have identical label and structures, their logical similarity value can still be imperfect.

Take a term *TelephoneNumber*, which consists of two child elements: an *AreaCode* element followed by a *Number* element. Suppose that there are two *TelephoneNumber* definitions, one defines the types (ranges) associated with child elements as *Integer* while the other defines them as *String*. Although the two have exact labels and structures, a good logical similarity measure would indicate that they are not identical and potentially incompatible. The logical similarity measures can provide more powerful estimates when matching schemas using additional model-based information. For example, if there is model-based information that indicates that the *String* type subsumes the *Integer* - indicating the *Integer* is convertible to the *String*, but not vice versa - then the measure may be used to indicate that the term is always translatable to the other but not vice versa. Some example approaches in this category include DL-based [1, 17], instance-based [16], and graph-based [11] approaches.

Although the logical similarity measures are potentially more accurate due to their formal basis, they require the

model to provide significant additional information, which is often unavailable. When model-based information is shallow, the quality of the approach may be reduced drastically. Hence, any schema discovery engine using logical similarity measures has to adjust the weights based on the amount and kind of model-based information available. In particular, a lower weight should be given to the logical similarity if the subsumption hierarchy is very shallow.

Finally, we offer our synthesized view of the respective roles and potentials uses of the aforementioned similarity metrics on the XML Schema discovery task. Schema discovery in the enterprise-applications-integration context is a unique information retrieval problem, because the goal is not to retrieve the content but the data model associated with the content. Specific consideration must be given to terms and naming conventions, design and structure conventions, usage cases, and semantic/ontology models, all of which must be considered simultaneously when matching schemas to a requirement. Therefore, it is not likely that a single similarity category would yield optimal results.

Synthesis of various similarity metrics within a search algorithm is likely to produce more accurate results. However, achieving such a synthesis is not straightforward. On the one hand, lexical measures may be more effective when a domain-specific thesaurus or dictionary is available. On the other hand, structural measures will be more effective when the data exchange requirements and the standard specification schemas within the repository are similarly constructed or are known to follow the same design conventions. In such well-controlled situations, the two similarity metric categories may play more deterministic roles, while the measures within the logical similarity category may appropriately play an auxiliary role, particularly when the schemas and the requirement are totally disparate.

References

1. Giunchiglia, F., Shvaiko, P., and Yatskevich, M. 2004. S-Match: An Algorithm and an Implementation of Semantic Matching. In *Proc. of ESWS*:61-75.
2. WWW Consortium. 1999. *XML PATH Language 1.0*.
3. The Open Application Group. 2002. *Open Application Group Integration Specification version 8.0*.
4. Automotive Industry Action Group. 2005. *Proof of Concept Phase I Project Summary*.
5. McHale, M. 1998. A Comparison of WordNet and Roget's Taxonomy for Measuring Semantic Similarity. In *Proc. of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, Montreal, Canada:115-120.
6. Jarmasz, M., and Szpakowicz, S. 2003. Roget's Thesaurus and Semantic Similarity. In *Proc. of Conf. on Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria:212-219.

7. Resnik, P. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proc. of the 14th Intl. Joint Conf. on AI*, Montreal, Canada:448-453.
8. Zhang, Z., Li, R., Cao, S., and Zhu, Y. 2003. Similarity Metric for XML Documents. In *Proc. of Workshop on Knowledge and Experience Management*, Karlsruhe, Germany.
9. Buttler, D. 2004. A Short Survey of Document Structure Similarity Algorithms. In *Proc. of the 5th Intl. Conf. on Internet Computing*, Las Vegas, Nevada.
10. Bhavsar, V.C., Boley, H., and Yang, L. 2003. A Weighted-Tree Similarity Algorithm for Multi-Agent Systems in e-Business Environments, In *Proc. of the Business Agents and the Semantic Web (BAsEWEB) Workshop*, Halifax, Nova Scotia, Canada.
11. Noy, N.F., and Musen, M.A. 2001. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In *Proc. of the Workshop on Ontologies and Information Sharing at the 17th Intl. Joint Conf. on Artificial Intelligence*, Seattle, WA.
12. Nierman, A. and H.V. 2002. Evaluating Structural Similarity in XML Documents. In *Proc. of the 5th Intl. Workshop on the Web and Databases*, Madison, WI.
13. Sussna, M. 1993. Word Sense Disambiguation for Free-Text Indexing using a Massive Semantic Network. In *Proc. of the 2nd Intl. Conf. on Information and Knowledge Management*, Arlington, VA.
14. Richardson, R., and Smeaton, A.F. 1995. *Using WordNet in a Knowledge-based Approach to Information Retrieval*, Working Paper, School of computer applications, Dublin City University, Ireland.
15. Flesca, S., Manco, G., Masciari, E., Pntieri, L., and Pugliese, A. 2002. Detecting Structural Similarities between XML Documents. In *Proc. of the 5th Intl. Workshop on the Web and Databases*, Madison, WI.
16. Doan, A., Madhavan, J., Domingos, P., and Halevy, A. 2003. Learning to Match Ontologies on the Semantic Web, *VLDB Journal, Special Issue on the Semantic Web*.
17. Peng, Y., Zou, Y., Luan, X., Ivezic, N., Gruninger, M., and Jones, A. 2003. Semantic Resolution for e-Commerce, Innovative Concepts for Agent-Based Systems. *Springer-Verlag* :355-366.
18. Do, H. and Rahm, E. 2001. COMA: A System for Flexible Combination of Schema Matching Approaches, In *Proc. of VLDB*, Roma, Italy:610-621.
19. Duda, R., Hart, P., and Stork, D. 2001. *Pattern Classification*, 2nd Edition, *Wiley-Interscience*.
20. Castano, S., De Antonellis, V., and De Capitani di Vimercati, S. 2001. Global Viewing of Heterogeneous Data Sources, *IEEE Transactions on Knowledge and Data Engineering* 13(2):277-297.

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 these products are necessarily the best available for the purpose.